Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Projeto e Implementação de um *Framework* para a Geração de Conjunto de Dados Sintéticos para Aprendizado Multirrótulo

Jimena Torres Tomás
Newton Spolaôr
Everton Alvares Cherman
Maria Carolina Monard

N° XXX

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos, SP, Brasil
Janeiro/2013

Projeto e Implementação de um *Framework* para a Geração de Conjunto de Dados Sintéticos para Aprendizado Multirrótulo* †

Jimena Torres Tomás[1]
Newton Spolaôr[1,2]
Everton Alvares Cherman[1]
Maria Carolina Monard[1]

[1]Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
Laboratório de Inteligência Computacional
CEP 13560-970 - São Carlos, SP, Brasil

[2]Universidade Estadual do Oeste do Paraná
Centro de Engenharias e Ciências Exatas
Laboratório de Bioinformática
CEP 85867-900 - Foz do Iguaçu, PR, Brasil

Janeiro, 2013

*Trabalho desenvolvido com o apoio da FAPESP.
†Disponível on-line em: http://www.icmc.usp.br/~biblio/relatorios_tecnicos.php.

Resumo

O conhecimento a priori das características de um conjuntos de dados contribui para um melhor entendimento dos algoritmos de aprendizado e das avaliações comparativas entre esses algoritmos. Esse conhecimento possibilita realizar experimentos em um ambiente relativamente controlado, o qual pode ser obtido utilizando conjuntos de dados sintéticos (artificiais). Entretanto, são poucas as estratégias propostas na literatura para a geração de conjuntos de dados sintéticos para aprendizado multirrótulo e há uma carência de ferramentas computacionais que implementam essas estratégias. Neste trabalho é descrito o projeto e a implementação de um *framework* para a geração de conjunto de dados sintéticos multirrótulo, que conta com duas ferramentas, a primeira utiliza hiperesferas para determinar os multirrótulos do conjunto de dados e a segunda utiliza hipercubos. O uso das ferramentas é ilustrado por meio de alguns experimentos com conjuntos de dados gerados pelo *framework*, o qual está disponível para a comunidade[1].

Palavras-Chave: Aprendizado Multirrótulo, Conjunto de Dados Sintéticos, Geração de Dados Sintéticos Multirrótulo

[1] http://sites.labic.icmc.usp.br/mldatagen/

Este documento foi preparado com o formatador de textos LaTeX. O sistema de citações de referências bibliográficas utiliza o padrão *Apalike* do sistema bibTeX.

Sumário

Lista de Figuras vii

Lista de Tabelas vii

Lista de Algoritmos vii

1 Introdução 1
- 1.1 Contextualização e Motivação . . . 1
- 1.2 Objetivos . . . 2
- 1.3 Organização do Trabalho . . . 2

2 Aprendizado Multirrótulo 2
- 2.1 Definições . . . 3
- 2.2 Medidas Multirrótulo . . . 5
  - 2.2.1 Densidade e Cardinalidade de Rótulo . . . 5
  - 2.2.2 Medidas de Avaliação . . . 5
- 2.3 Estratégias para Geração de Conjuntos de Dados Sintéticos Multirrótulo . . . 9

3 Projeto e Implementação de um Framework para Geração de Conjuntos de Dados Sintéticos 10
- 3.1 Ferramenta Hiperesferas . . . 11
  - 3.1.1 Geração da Hiperesfera Principal . . . 12
  - 3.1.2 Geração das Hiperesferas contidas na Hiperesfera Principal 12
  - 3.1.3 Geração Direcionada dos Exemplos . . . 15
  - 3.1.4 Geração dos Multirrótulos . . . 18
- 3.2 Ferramenta Hipercubos . . . 19
  - 3.2.1 Geração do Hipercubo Principal . . . 20
  - 3.2.2 Geração dos Hipercubos contidos no Hipercubo Principal . 20
  - 3.2.3 Geração Direcionada dos Exemplos . . . 21
  - 3.2.4 Geração dos Multirrótulos . . . 24
- 3.3 O *Framework* . . . 24

4 Exemplo de Uso 26
- 4.1 Descrição dos Conjuntos de Dados . . . 26
- 4.2 Algoritmo Multirrótulo Utilizado . . . 27
- 4.3 Resultados e Discussão . . . 28

5 Conclusão 32

Lista de Figuras

1 Exemplo de *CD* sintético multirrótulo proposto em (Zhang et al., 2009) . . . 10
2 Hiperesfera principal HS, para $M = 2$ . . . 13
3 Domínio para geração de C_i, dado r_i, para $M = 2$ . . . 14
4 Domínio para geração de $\mathbf{x}_k$, dado r_i, para $M = 2$ . . . 16
5 Hipercubo principal HC, para $M = 2$ . . . 20
6 Domínio para geração de C_i, dado e_i, para $M = 2$ . . . 21
7 Domínio para geração de $\mathbf{x}_k$, dado e_i, para $M = 2$ . . . 23
8 Página incial do *framework* . . . 25
9 Página de configuração do *framework* . . . 26
10 Página de saída do *framework* . . . 27
11 Gráficos radares dos CD_A, CD_B, CD_C e CD_D. . . . 31
12 Gráficos radares dos CD_E e CD_F. . . . 31

Lista de Tabelas

1 Exemplos rotulados no formato atributo-valor. . . . 4
2 Matrizes de confusão para conjunto D com $L = \{a, b, c, d\}$ . . . 8
3 Valores das medidas *micro-avareged* e *macro-avareged* para as quatro medidas binárias apresentadas . . . 8
4 Distribuição dos atributos nos *CD* sintéticos . . . 27
5 Frequência dos monorótulos . . . 29
6 Valores de cardinalidade e densidade de rótulo . . . 29
7 Porcentagem de multirrótulos gerados com cardinalidade de rótulo 1 até 10 . . . 30
8 Resultados da avaliação experimental . . . 30

Lista de Algoritmos

1 Algoritmo para geração das hiperesferas menores . . . 15
2 Algoritmo para geração dos exemplos nas hiperesferas . . . 18
3 Algoritmo para geração dos hipercubos menores . . . 22
4 Algoritmo para geração dos exemplos nos hipercubos . . . 23

1 Introdução

1.1 Contextualização e Motivação

Muitos dos métodos de Aprendizado de Máquina (*AM*) supervisionado consideram que cada exemplo está associado a um único rótulo (monorrótulo) (Mitchell, 1997; Alpaydin, 2010). Entretanto, existem vários domínios de aplicação nos quais cada exemplo está associado a mais de um rótulo. Métodos que são capazes de aprender nesses domínios são conhecidos como métodos de aprendizado multirrótulo. Recentemente, os estudos da classificação multirrótulo tem incrementado, devido à grande gama de aplicações multirrótulo existentes, tais como anotação de imagens e vídeo, bioinformática, busca e mineração na *web* e análise de emoções (Tsoumakas et al., 2010).

Com o crescente uso dessas aplicações, o progresso tecnológico tem possibilitado a geração de uma quantidade, cada vez maior, de bases de dados multirrótulo de diversos domínios. Essas bases de dados podem ser submetidas a processos de análise inteligente, como a Mineração de Dados (*MD*), com o intuito de identificar padrões para construir modelos computacionais que auxiliem a tomada de decisão nesses domínios.

As principais etapas do processo de *MD* são: pré-processamento, extração de padrões e pós-processamento. O pré-processamento consiste na preparação e transformação dos dados com o objetivo de encontrar uma representação adequada para a próxima etapa. Com o auxílio de métodos como os de *AM*, na etapa de extração de padrões são gerados modelos (hipóteses) baseados em padrões que contemplam o conhecimento embutido nos dados. O pós-processamento envolve a avaliação e a validação dos modelos gerados, o que possibilita a consolidação do conhecimento adquirido e o suporte à tomada de decisão.

A área de *AM* é naturalmente empírica (Dietterich, 1990). Assim, algoritmos dessa área são muitas vezes avaliados e comparados por meio de estudos experimentais (Langley, 2000). Parte desses estudos envolve a investigação do desempenho dos algoritmos em conjuntos de dados projetados para *benchmarking*, de origem real ou artificial (sintético). Os Conjuntos de Dados (*CD*) sintéticos, particularmente, são gerados a partir de características estabelecidas *a priori*. Uma vantagem importante do uso de *CD* sintéticos é a possibilidade de realizar experimentos em um ambiente relativamente controlado, o que permite entender e analisar em maiores detalhes o desempenho esperado de algoritmos de aprendizado em *CD* com características diferentes.

Atualmente é possível identificar alguns programas de código aberto para apoio à geração de *CD* sintéticos para aprendizado monorrótulo, como o *Data*

Generation Program/2[2], *Dataset Generator*[3] (DatGen) e *Data Set Generator*[4] (Datagen). Contudo, nota-se uma carência de ferramentas com intuito similar para a geração de *CD* multirrótulo. Até o momento, na literatura identificam-se alguns trabalhos que descrevem estratégias para a construção de *CD* artificiais para aprendizado multirrótulo (Noh et al., 2004; Chou and Hsu, 2005; Zhang et al., 2009), mas não é de nosso conhecimento a existência de programas de código aberto para apoio à geração de *CD* sintéticos multirrótulo disponíveis para a comunidade.

1.2 Objetivos

Neste trabalho é descrito o projeto e a implementação de duas estratégias para a geração de *CD* sintéticos multirrótulo. A primeira estratégia, proposta por Zhang et al. (2009), utiliza hiperesferas para a determinação dos multirrótulos. A ferramenta implementada neste trabalho utilizando essa estratégia estende as facilidades da proposta original. Utilizando uma estratégia semelhante, neste trabalho também propomos uma ferramenta que utiliza hipercubos para a determinação dos multirrótulos. Ambas ferramentas estão integradas em um *framework* disponibilizado para a comunidade[5], com o objetivo de reduzir a carência de ferramentas computacionais para a geração de *CD* sintéticos multirrótulo.

1.3 Organização do Trabalho

O restante deste trabalho está organizado da seguinte maneira: na Seção 2 são apresentados conceitos relacionados ao aprendizado multirrótulo e algumas estratégias para a geração de *CD* sintéticos multirrótulo propostas na literatura. A implementação das ferramentas Hiperesferas e Hipercubos é detalhada na Seção 3. Visando ilustrar a utilização do *framework*, foi realizada uma avaliação experimental, descrita na Seção 4. Na Seção 5 são apresentadas as considerações finais deste trabalho.

2 Aprendizado Multirrótulo

Aprendizado de máquina é uma área de Inteligência Artificial que tem como objetivo o desenvolvimento de técnicas computacionais para a aquisição automática de conhecimento utilizando exemplos ou experiências observadas (Mitchell, 1997; Alpaydin, 2010).

[2] http://archive.ics.uci.edu/ml/machine-learning-databases/dgp-2
[3] http://www.datasetgenerator.com
[4] http://www.burningart.com/meico/inventions/datagen/index.html
[5] http://sites.labic.icmc.usp.br/mldatagen/

A maioria dos métodos de aprendizado supervisionado considera que cada um dos exemplos do conjunto de exemplos está associado a um valor da classe, em outras palavras, esta associado a um único rótulo. Nesse caso, o aprendizado é chamado aprendizado monorrótulo. Entretanto, existem vários domínios de aplicação nos quais os exemplos estão associados a um conjunto de rótulos. Esse tipo de exemplos é denominado multirrótulo (Tsoumakas et al., 2010). Por exemplo, em classificação de textos, um documento pode pertencer simultaneamente a mais de um tópico, como informática e agricultura. No diagnóstico médico, um paciente pode apresentar câncer ao mesmo tempo em que também apresenta diabetes. Em bioinformática, a classificação multirrótulo é aplicada na predição das funções de genes, devido ao fato que um gene pode influenciar em mais de uma função simultaneamente. Outra aplicação é a classificação de músicas em emoções. Nesse caso, uma música pode ser caracterizada por mais de uma emoção como calma, feliz, energética, entre outras.

Devido à crescente gama de aplicações compostas por exemplos multirrótulo, a quantidade de pesquisas relacionadas ao aprendizado multirrótulo tem incrementado nos últimos anos (Spolaôr et al., 2012). A seguir são introduzidos alguns conceitos, medidas de avaliação e abordagens de aprendizado multirrótulo.

2.1 Definições

Para a tarefa de aprendizado supervisionado monorrótulo, a entrada do algoritmo consiste de um conjunto de N exemplos E_i, com $i = 1, ..., N$, escolhidos de um domínio com uma distribuição $\mathcal{D}$ fixa, desconhecida e arbitrária, da forma $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ para alguma função desconhecida $y = f(\mathbf{x})$. Os $\mathbf{x}_i$ são tipicamente vetores da forma $(x_{i1}, x_{i2}, \ldots, x_{iM})$ com valores discretos ou contínuos, tal que x_{ij} refere-se ao valor do atributo j, denominado X_j, para o exemplo E_i, como apresentado na Tabela 1(a). Os valores (rótulos) y_i referem-se ao valor que a classe Y pode assumir.

Dado um conjunto D de exemplos a um algoritmo de aprendizado, um *classificador* H, que aproxima f, *i.e.*, $H(\mathbf{x}) \approx f(\mathbf{x})$, é gerado. O classificador consiste da *hipótese* feita sobre a verdadeira (mas desconhecida) função f. Dados novos exemplos $\mathbf{x}$, o classificador, ou hipótese H, prediz o valor correspondente de y.

Assim, na classificação monorrótulo, cada exemplo E_i está associado a um único valor da classe, que é o rótulo y_i contido no conjunto de rótulos L, *i.e.*, $y_i \in L$, com $|L| > 1$. Quando a classe pode assumir somente dois valores, o problema é chamado de *classificação binária*. Quando a classe pode assumir mais de dois valores, o problema é chamado de *classificação multiclasse*. Ou

seja, o aprendizado binário pode ser considerado como um caso particular do aprendizado multiclasse. Por outro lado, na *classificação multirrótulo*, um exemplo pode estar associado a mais de um rótulo, *i.e.*, um exemplo E_i está associado a um conjunto de rótulos Y_i. O exemplo E_i é representado da forma $(\mathbf{x}_i, Y_i)$, onde $Y_i \subseteq L$ e $L = \{y_1, y_2, y_3, ...y_q\}$ é o conjunto dos q rótulos simples que participam dos multirrótulos Y_i, $i = 1..N$, como representado na Tabela 1(b)[6].

Tabela 1: Exemplos rotulados no formato atributo-valor.

(a) Multiclasse

	$\mathbf{X}_1$	$\mathbf{X}_2$	$\dots$	$\mathbf{X}_M$	Y
E_1	x_{11}	x_{12}	$\dots$	x_{1M}	y_1
E_2	x_{21}	x_{22}	$\dots$	x_{2M}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{N2}	$\dots$	x_{NM}	y_N

(b) Multirrótulo

	$\mathbf{X}_1$	$\mathbf{X}_2$	$\dots$	$\mathbf{X}_M$	Y
E_1	x_{11}	x_{12}	$\dots$	x_{1M}	Y_1
E_2	x_{21}	x_{22}	$\dots$	x_{2M}	Y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{N2}	$\dots$	x_{NM}	Y_N

A principal diferença entre aprendizado multirrótulo e aprendizado monorrótulo é que o conjunto de rótulos no aprendizado multirrótulo estão frequentemente correlacionados, enquanto que os possíveis valores da classe (rótulos) no aprendizado monorrótulo são mutuamente exclusivos.

Dado um conjunto de exemplos é importante distinguir, qualquer que seja o tipo de aprendizado, os seguintes três conjuntos utilizados no processo de aprendizado:

Conjunto de treinamento Esse conjunto é a principal entrada dos algoritmos de aprendizado. É a partir dele que são construídos os modelos (hipóteses) e, portanto, ele deve ser representativo da distribuição da população dos dados do domínio. Ou seja, os exemplos desse conjunto são "vistos" pelo algoritmo de aprendizado durante a construção do modelo.

Conjunto de teste Esse conjunto é utilizado para avaliar o modelo construído. Esse conjunto não deve ser apresentado ao algoritmo de aprendizado durante a construção do modelo. Idealmente, esse conjunto não deveria ter exemplos em comum com o conjunto de treinamento.

Conjunto de validação Em alguns casos, pode ser necessário utilizar exemplos para realizar ajustes no modelo construído pelo algoritmo de aprendizado. Esses exemplos não são utilizados diretamente na construção do modelo, mas são utilizados para o seu ajuste. Dessa maneira, esses exemplos são indiretamente "vistos" durante o processo de aprendizado,

[6]Existe também a classificação hierárquica multirrótulo, que combina classificação hierárquica, na qual uma ou mais classes podem ser divididas em subclasses ou agrupadas em superclasses, com classificação multirrótulo.

o que obriga que os exemplos de validação sejam distintos dos exemplos de teste.

2.2 Medidas Multirrótulo

As medidas tradicionais para classificadores monorrótulo não consideram classificações parcialmente corretas, o que é uma característica da classificação multirrótulo. Assim, diversas medidas para a classificação multirrótulo, descritas a seguir, foram propostas. Na definição dessas medidas, D representa um conjunto de dados multirrótulo composto por $|D| = N$ exemplos $(\mathbf{x}_i, Y_i)$, com $i = 1 \ldots N$ e $Y_i \subseteq L$ o conjunto de rótulos, *i.e.* o multirrótulo do i-ésimo exemplo. H é um classificador multirrótulo com $Z_i = H(\mathbf{x}_i)$ o conjunto de rótulos preditos por H para um dado $\mathbf{x}_i$.

2.2.1 Densidade e Cardinalidade de Rótulo

O número de exemplos multirrótulo e a quantidade de rótulos de cada exemplo influenciam o desempenho dos métodos de classificação (Tsoumakas and Katakis, 2007). As duas medidas a seguir consideram essas características.

A *Cardinalidade de Rótulo* (CR) representa a média do número de rótulos distintos atribuídos a cada exemplo. Essa medida é independente da quantidade de rótulos simples que participam dos resultados — Equação 1.

A *Densidade de Rótulo* (DR) calcula também a média do número de rótulos distintos atribuídos a cada exemplo. Porém, esse cálculo é ponderado pela quantidade de rótulos simples que participam dos multirrótulos — Equação 2.

$$CR(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} |Y_i| \qquad (1) \qquad\qquad DR(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i|}{|L|} \qquad (2)$$

Ambas as métricas estão relacionadas por $CR(D) = |L| DR(D)$.

2.2.2 Medidas de Avaliação

As medidas de avaliação de classificação monorrótulo consideram apenas dois estados possíveis para a classificação de um exemplo: correta ou incorreta. Por outro lado, na classificação multirrótulo existe o meio termo de quão correta é a classificação de cada exemplo. Nesse tipo de aprendizado, os exemplos podem ter rotulação parcialmente correta, o que exige medidas de avaliação específicas para esse problema.

Diversas medidas foram propostas para a avaliação da classificação multirrótulo e podem ser organizadas em três grupos distintos. Algumas das medidas propostas consideram as diferenças entre o conjunto de rótulos esperado, Y_i, e o predito, Z_i, nos exemplos do conjunto de teste. Esse tipo de

medidas é chamada *Baseadas em exemplos*. Outras medidas dissecam o processo de avaliação dos algoritmos em medidas separadas para cada rótulo, e então ponderam esse valor sobre todos os rótulos. Esse tipo de medida é chamada *Baseadas em rótulos*. Por fim, existem medidas que utilizam os *ranking* dos rótulos para avaliar o resultado de uma classificação; esse tipo de medida é chamada *Baseadas em Ranking*.

A seguir, são detalhadas as medidas de avaliação utilizadas neste trabalho.

Medidas de Avaliação Baseadas em Exemplos

Uma das medidas utilizadas para a avaliação de classificadores multirrótulo é a *Hamming Loss* (Tsoumakas and Katakis, 2007), a qual é definida pela Equação 3.

$$Hamming\,Loss(H, D) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \Delta Z_i|}{|L|} \tag{3}$$

Nessa medida, Δ representa a diferença simétrica entre dois conjuntos, a qual é equivalente à operação booleana *XOR*. Quanto menor for o valor de *Hamming Loss*, melhor é a classificação. O ideal é que esse valor seja zero.

A medida *Subset Accuracy* é definida pela Equação 4.

$$Subset\,Accuracy(H, D) = \frac{1}{N}\sum_{i=1}^{N} I(Z_i = Y_i) \tag{4}$$

onde I é uma função que mapeia verdadeiro em 1 e falso em 0. O *Subset Accuracy* é uma medida que não leva em conta resultados *parcialmente* corretos, uma vez que a avaliação $Z_i = Y_i$ só possui duas possíveis respostas.

Outras medidas foram utilizadas em (Godbole and Sarawagi, 2004), as quais correspondem a adaptações das medidas tradicionalmente utilizadas em problemas monorrótulo para problemas multirrótulo. Essas medidas de avaliação multirrótulo são apresentadas nas Equações 5, 6, 7 e 8.

$$\textit{Precision}(H, D) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{5}$$

$$\textit{Recall}(H, D) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{6}$$

$$\textit{Accuracy}(H, D) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{7}$$

$$\textit{F-Measure}(H, D) = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \tag{8}$$

De maneira intuitiva, a medida *Accuracy* representa os acertos obtidos da predição dos rótulos verdadeiros e também considera o fato de não predizer um rótulo que deveria ser predito. A medida *Precision* considera apenas se os rótulos preditos realmente estão entre os rótulos que deveriam ser preditos. Caso todos estejam, não importando a quantidade, o valor de *Precision* é máximo. O *Recall*, no entanto, considera a quantidade de rótulos preditos que estão contidos no conjunto de rótulos verdadeiros. Em outras palavras, se todos os rótulos verdadeiros estão contidos no conjunto de rótulos preditos, não importando a existência de outros rótulos no conjunto predito, o valor de *Recall* é máximo. Já a medida *F-Measure* representa uma média harmônica entre a *Precision* e o *Recall*, de modo que o valor de *F-Measure* considera um equilíbrio entre as duas medidas.

Os valores das medidas *HammingLoss*, *SubsetAccuracy*, *Precision*, *Accuracy*, *Recall* e *F-Measure* variam todas no intervalo $[0, 1]$. O valor ótimo para *HammingLoss* é 0 (zero), enquanto que esse valor é 1 (um) para as outras medidas.

Medidas de Avaliação Baseadas em Rótulos

Para classificadores binários, *i.e.* classificadores monorrótulo, os dois possíveis valores da classe são, geralmente, denominados positivo (+) e negativo (-). Uma maneira natural de avaliar classificadores monorrótulo é por meio dos valores de V_P, F_P, F_N e V_N, os quais correspondem, respectivamente, a quantidade de verdadeiros/falsos[7] positivos e falsos/verdadeiros negativos de um total de $N = V_P + F_P + F_N + V_N$ exemplos.

Em função desses valores, as medidas de avaliação de classificadores binários, apresentadas nas Equações 9, 10, 11 e 12, são calculadas.

$$\textit{Precision}(H, D) = \frac{V_P}{V_P + F_P} \tag{9}$$

$$Recall(H, D) = \frac{V_P}{V_P + F_N} \tag{10}$$

$$\textit{Accuracy}(H, D) = \frac{V_P + V_N}{N} \tag{11}$$

$$\textit{F-Measure}(H, D) = \frac{2V_P}{2V_P + F_P + F_N} \tag{12}$$

No grupo de medidas baseadas em rótulos, qualquer uma dessas medidas

[7]Quando um exemplo cuja classe real é positiva é classificado como positivo, ele é denominado verdadeiro positivo (V_P). Caso a classe real seja negativa e ele for classificado como positivo, ele é denominado falso positivo (F_P). Notação similar é utilizada no caso dos exemplos negativos.

de avaliação de classificação binária pode ser utilizada para calcular medidas, a elas relacionadas, de classificadores multirrótulo.

Para um classificador multirrótulo com $L = \{y_1, y_2, \ldots, y_q\}$, duas medidas baseadas em rótulo podem ser utilizadas, denominadas *macro-averaged* e *micro-averaged* (Tsoumakas et al., 2010), descritas a seguir.

Seja $B(V_P, F_P, F_N, V_N)$ uma medida de classificação binária, calculada em função de uma matriz de confusão. Seja $V_{P_{y_i}}$, $F_{P_{y_i}}$, $F_{N_{y_i}}$ e $V_{N_{y_i}}$ o número de verdadeiros/falsos positivos e falsos/verdadeiros negativos para o rótulo y_i do conjunto de rótulos L. As versões de *macro-averaged* e *micro-averaged* da medida B são definidas pelas Equações 13 e 14.

$$B_{macro} = \frac{1}{q} \sum_{i=1}^{q} B\left(V_{P_{y_i}}, F_{P_{y_i}}, F_{N_{y_i}}, V_{N_{y_i}}\right) \tag{13}$$

$$B_{micro} = B\left(\sum_{i=1}^{q} V_{P_{y_i}}, \sum_{i=1}^{q} F_{P_{y_i}}, \sum_{i=1}^{q} F_{N_{y_i}}, \sum_{i=1}^{q} V_{N_{y_i}}\right) \tag{14}$$

Para exemplificar, considere que foi avaliado um classificador multirrótulo H sobre um conjunto de 40 exemplos de teste com $L = \{a, b, c, d\}$. Para cada um dos rótulos simples em L é gerada uma matriz de confusão, apresentadas na Tabela 2, a partir da avaliação do classificador multirrótulo H.

Tabela 2: Matrizes de confusão para conjunto D com $L = \{a, b, c, d\}$

(a) Rótulo a

20 V_P	0 F_N
0 F_P	20 V_N

(b) Rótulo b

20 V_P	10 F_N
0 F_P	10 V_N

(c) Rótulo c

10 V_P	0 F_N
20 F_P	10 V_N

(d) Rótulo d

10 V_P	10 F_N
10 F_P	10 V_N

Na Tabela 3 são apresentados os resultados para B_{macro} e B_{micro}, considerando as medidas de classificação binária calculadas sobre as matrizes de confusão da Tabela 2.

Tabela 3: Valores das medidas *micro-avareged* e *macro-avareged* para as quatro medidas binárias apresentadas

	Acurácia	*Precisão*	*Recall*	*F-measure*
macro-avareged	0,69	0,75	0,83	0,70
micro-avareged	0,69	0,67	0,75	0,71

2.3 Estratégias para Geração de Conjuntos de Dados Sintéticos Multirrótulo

Como mencionado, várias estratégias tem sido propostas para a geração de *CD* sintéticos monorrótulo. Entretanto, são poucas as estratégias propostas para a geração de *CD* multirrótulo. Algumas dessas estratégias são descritas a seguir.

Chou and Hsu (2005) propõem construir um *CD* sintético por meio de determinadas funções para rotular exemplos. Esse *CD* é estendido de um gerador de conjuntos sintéticos sobre consumidores de uma empresa de turismo tradicional no aprendizado monorrótulo, descrito em (Agrawal et al., 1992). Esse conjunto possui $M = 9$ atributos, dos quais três são multivalorados. De modo geral, os atributos são gerados a partir de uma distribuição uniforme de um intervalo de valores definidos *a priori*.

No trabalho de Noh et al. (2004) foram estudadas distintas características de *CD* sintéticos para avaliação de árvores de decisão multirrótulo com seleção de atributos embutida (Liu and Motoda, 2008). Nesse trabalho também foram estudados conjuntos de dados sintéticos para investigar a capacidade das árvores em identificar um determinado atributo relacionado à classe durante a seleção de atributos. Noh et al. (2004) também apresentam como diferencial importante a formulação de várias funções para definir os valores dos atributos a partir de rótulos dos multirrótulos Y.

Neste trabalho utilizamos a estratégia proposta por Zhang et al. (2009) que propõem hiperesferas para gerar os multirrótulos dos exemplos. Nessa estratégia é construída inicialmente uma hiperesfera HS com raio r em um espaço d-dimensional, com $d < M$. Randomicamente são geradas q hiperesferas hs_k contidas em HS, com $L = \{y_1, y_2, \ldots y_q\}$, em que cada hs_k está associada a um rótulo y_k. Em seguida, os exemplos são gerados randomicamente dentro de HS com d atributos revelantes, de modo que cada exemplo E_i esteja associado a um subconjunto de rótulos Y_i correspondente a zero, uma ou mais hiper-esferas hs_k. Na Figura 1 é ilustrado um *CD* sintético que pode ser obtido a partir dessa estratégia considerando dois atributos relevantes X_1 e X_2 (Zhang et al., 2009). Caso um exemplo situar-se fora de todas as hiper-esferas hs_k, então o subconjunto de rótulos correspondente é vazio (zero rótulos). Convém ressaltar que, quando implementamos essa estratégia, consideramos procedimentos adicionais para evitar a geração de exemplos com zero rótulo.

Após, adicionam-se para cada exemplo $d/2$ atributos irrelevantes, gerados a partir de valores randômicos, e $d/2$ atributos redundantes, os quais replicam metade dos d atributos relevantes originais. Assim, cada exemplo do *CD* apresenta ao final $M = d + d/2 + d/2 = 2d$ atributos.

Nos 12 *CD* gerados no trabalho de Zhang et al. (2009), varia-se a quantidade

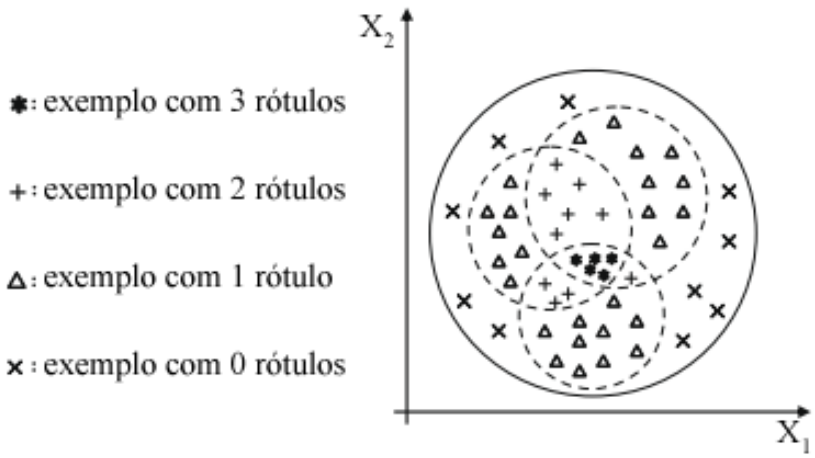

Figura 1: Exemplo de *CD* sintético multirrótulo proposto em (Zhang et al., 2009)

de atributos relevantes d nos valores 40, 60, 80 e 100 e a quantidade de rótulos simples q nos valores 10, 15 e 20. A hiperesfera HS tem raio $r = 1$ e centro na origem, de modo que os valores dos atributos variam entre -1 e 1. Para introduzir ruído e obter uma melhor aproximação de dados reais, a presença ou ausência de um rótulo y_k atribuído originalmente a um exemplo pode ser alterada no *CD* final com probabilidade $0,05$.

3 Projeto e Implementação de um Framework para Geração de Conjuntos de Dados Sintéticos

A seguir são descritas em detalhes a implementação de duas ferramentas, baseadas na proposta de Zhang et al. (2009), as quais estão integradas em um *framework* para geração de *CD* sintéticos multirrótulos.

Esses *CD* são gerados no formato requerido pelo Mulan[8], o qual consiste de um *framework* que implementa uma grande gama de métodos multirrótulo, os quais podem ser utilizados e ampliados, bem como disponibiliza diversas medidas de avaliação (Tsoumakas et al., 2011). Esse *framework* trabalha em conjunto com as classes Java[9] do Weka[10], um ambiente bastante conhecido e utilizado pela comunidade de aprendizado de máquina e de mineração de dados (Hall et al., 2009; Witten et al., 2011).

O formato requerido pelo Mulan envolve dois arquivos, dos quais um é do tipo `ARFF` (*Attribute-Relation File Format*) e o outro é do tipo `XML`[11] (*eXtensible Markup Language*). Assim, após um *CD* sintético com ruído e outro sem ruído

[8] http://mulan.sourceforge.net
[9] http://www.oracle.com/technetwork/java/index.html
[10] http://www.cs.waikato.ac.nz/ml/weka/
[11] http://mulan.sourceforge.net/format.html

ter sido criado utilizando as ferramentas desenvolvidas neste trabalho, ele pode ser diretamente utilizado como entrada para o Mulan.

A descrição do projeto e da implementação das ferramentas que integram o *framework* é apresentada nas próximas seções.

3.1 Ferramenta Hiperesferas

Nesta seção é descrita a implementação da Ferramenta Hiperesferas, que assim como em (Zhang et al., 2009) utiliza hiperesferas, no $\mathbb{R}^M$, para a geração do *CD* sintéticos multirrótulos.

A Ferramenta Hiperesferas, implementada em Java, recebe como entrada:

- M_{rel} - Número de atributos relevantes do conjunto de dados.
- M_{irr} - Número de atributos irrelevantes do conjunto de dados.
- M_{red} - Número de atributos redundantes do conjunto de dados.
- q - Número de rótulos do conjunto de dados.
- N - Número de exemplos do conjunto de dados.
- $maxR$ - Raio máximo das hiperesferas menores, *default* = 0,8.
- $minR$ - Raio mínimo das hiperesferas menores, *default* = $\left\lfloor \frac{\frac{q}{10}+1}{q} \right\rfloor$.
- μ - Taxa de ruído que será inserida nos multirrótulos do *CD*, *default* = 0,05.

Foi implementada uma crítica dos dados, de modo a não permitir valores inválidos para os parâmetros de entrada. Para que as entradas sejam válidas deve-se garantir que:

- Todos os valores sejam positivos e não nulos, com exceção do ruído μ que pode ser nulo
- $minR < maxR \leq 0,8$
- $0 < minR \leq \left\lfloor \frac{\frac{q}{10}+1}{q} \right\rfloor$
- $0 \leq \mu \ \leq 1$

A saída para cada execução da ferramenta consiste de dois *CD* sintéticos com N exemplos e M atributos, tal que $M = M_{rel} + M_{irr} + M_{red}$. Um dos *CD* contém ruído μ, enquanto o outro não contém ruído.

Vale ressaltar que esta implementação complementa o que foi proposto no trabalho de (Zhang et al., 2009), permitindo que o usuário escolha a quantidade de atributos relevantes, redundantes e irrelevantes do *CD*. Essa alteração faz com que a ferramenta se torne mais abrangente e oferece uma maior liberdade ao usuário para explorar diferentes características dos *CD*, utilizando diferentes combinações do número de atributos relevantes, irrelevantes e/ou redundantes.

Para a geração dos *CD*, são realizados, de uma forma sucinta, os seguintes passos:

1. Geração da hiperesfera principal no $\mathbb{R}^M$;
2. Geração de q hiperesferas no $\mathbb{R}^M$, internas à hiperesfera principal;
3. Geração de N pontos no $\mathbb{R}^M$;
4. Geração dos multirrótulos, conforme as hiperesferas menores;
5. Inserção do ruído μ.

A descrição detalhada da implementação é apresentada nas próximas seções.

3.1.1 Geração da Hiperesfera Principal

Inicialmente é gerada uma hiperesfera HS no $\mathbb{R}^M$, com centro na origem do plano cartesiano e raio $r_{HS} = 1$, ilustrada na Figura 2, onde no eixo x está representado o atributo $\mathbf{X}_1$ e no eixo y o atributo $\mathbf{X}_2$ do conjunto de dados multirrótulos, para $M = 2$.

3.1.2 Geração das Hiperesferas contidas na Hiperesfera Principal

Em seguida são geradas no $\mathbb{R}^M$ q hiperesferas menores contidas na hiperesfera principal, ou seja, todas as hiperesferas menores devem estar completamente contidas na hiperesfera principal. O conjunto de hiperesferas menores é denotado por $H_{hs} = \{hs_1, hs_2, \ldots, hs_q\}$, no qual cada hiperesfera $hs_i = (r_i, C_i)$ é caracterizada por um raio r_i e um centro distinto $C_i = (c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iM})$.

Para a geração de cada hiperesfera menor hs_i, primeiramente é gerado um valor de raio r_i aleatório com distribuição aproximadamente uniforme, entre *minR* e *maxR*. Em seguida é necessário gerar aleatoriamente os valores das coordenadas do centro C_i, de forma a garantir que a hiperesfera menor hs_i esteja contida na hiperesfera maior, como formulado pela condição 15

$$\forall\, i \in [1..q]\ c_{ij} \leq (1 - r_i), j = 1..M \tag{15}$$

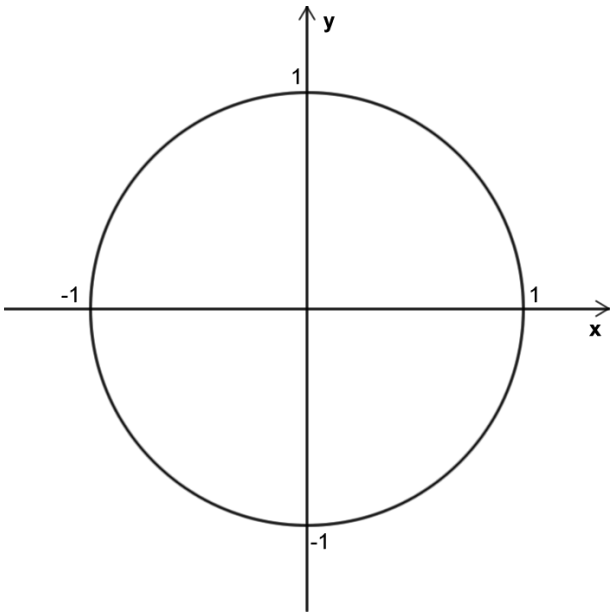

Figura 2: Hiperesfera principal HS, para $M = 2$

Entretanto, há um problema adicional, pois o fato de verificar a condição 15 não garante que o centro C_i das hiperesferas menores satisfaça a condição de hs_i estar contida em HS. Por exemplo, se for gerado um valor $c_{i1} = (1 - r_i)$, então os outros valores de c_{ij} para $j \neq 1$ devem, necessariamente, ser zero, ou seja, o resto dos valores das $M - 1$ coordenadas é fixo e igual a zero.

Assim, ao gerar cada C_i, $i = 1..q$, deve-se garantir que

$$\sqrt{c_{i1}^2 + c_{i2}^2 + \ldots + c_{iM}^2} \leq (1 - r_i). \tag{16}$$

O que é equivalente a

$$c_{i1}^2 + c_{i2}^2 + \ldots + c_{iM}^2 \leq (1 - r_i)^2. \tag{17}$$

$$\sum_{j=1}^{M} {c_{ij}}^2 \leq (1 - r_i)^2. \tag{18}$$

Portanto, o domínio das coordenadas do centro C_i é determinado pela Equação 18. Na Figura 3 é mostrado um exemplo gráfico para $M = 2$ no qual foi gerado um raio aleatório r_i e a única forma de garantir que a hiperesfera menor esteja totalmente contida na hiperesfera principal é que seu centro C_i esteja na área hachurada.

Analisando a Equação 17 conclui-se que a geração das coordenadas de cada centro C_i, $i = [1..q]$, das hiperesferas menores deve levar em conta o valor de r_i e os valores previamente gerados das coordenadas de C_i. Portanto, ao gerar cada coordenada c_{ij}, $j = [1..M]$ de forma aleatória, ela deve obedecer as

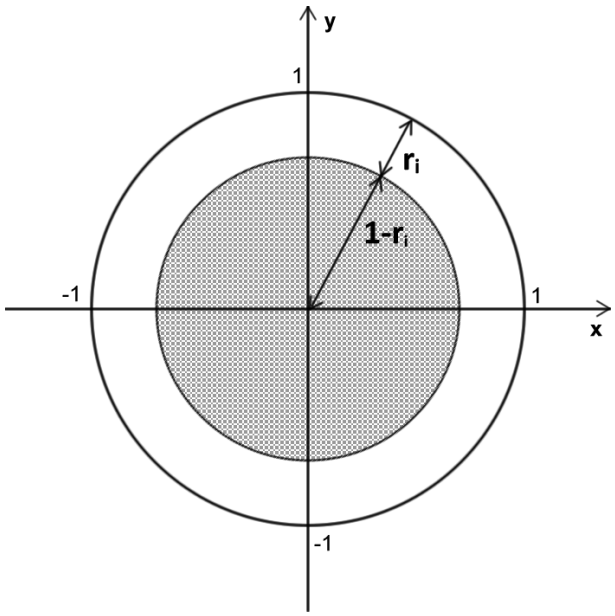

Figura 3: Domínio para geração de C_i, dado r_i, para $M = 2$

condições da Equação 19.

$$-\sqrt{(1-r_i)^2 - \sum_{s=1,s\neq j}^{M} (c_{is})^2} \leq c_{ij} \leq \sqrt{(1-r_i)^2 - \sum_{s=1,s\neq j}^{M} (c_{is})^2}. \tag{19}$$

Caso algum c_{is}, para $s \neq j$, ainda não tenha sido instanciado, o seu valor não é considerado.

Visto que os intervalos possíveis para a geração de cada coordenada c_{ij} diminuem a medida que uma nova coordenada é instanciada, é necessário evitar determinismo na geração das coordenadas do centro C_i da hiperesfera hs_i, ou seja, que c_{i1} sempre seja a primeira coordenada a ser gerada e c_{iM} a última. Para isso, a ordem de instanciação das coordenadas é realizada de forma aleatória. Em outras palavras, j não é variado em ordem crescente no intervalo de $[1..M]$, e sim de forma aleatória.

No Algoritmo 1 é apresentado de forma simplificada como é realizada a geração das hiperesferas $hs_i = (r_i, C_i)$, para $i = [1..q]$. Nesse algoritmo, a função $random(x, y)$ gera um valor aleatório, com distribuição aproximadamente uniforme, entre x e y, e as funções $atualizaminC(x)$ e $atualizamaxC(x)$ atualizam, respectivamente, o limite inferior e superior da próxima coordenada de C_i a ser instanciada, segundo a Equação 19, levando em consideração as coordenadas anteriormente geradas para geração das próximas coordenadas. Esses procedimentos são úteis para evitar que exemplos com zero rótulos sejam gerados pela ferramenta.

Algoritmo 1 Algoritmo para geração das hiperesferas menores

1: $h_{si} \leftarrow \emptyset$
2: $C_i \leftarrow \emptyset$
3: $r_i \leftarrow 0$
4: **for** $i = 1 \rightarrow q$ **do**
5: $\quad maxC \leftarrow (1 - r_i)$
6: $\quad minC \leftarrow -(1 - r_i)$
7: $\quad r_i \leftarrow random(minR, maxR)$
8: $\quad$ **for** $j = 1 \rightarrow M$ $(sendo\, j\, escolhido\, aleatoriamente)$ **do**
9: $\quad\quad c_{ij} \leftarrow random(minC, maxC)$
10: $\quad\quad C_i \leftarrow C_i \cup \{c_{ij}\}$
11: $\quad\quad minC \leftarrow atualizaminC(minC)$
12: $\quad\quad maxC \leftarrow atualizamaxC(maxC)$
13: $\quad$ **end for**
14: $\quad hs_i \leftarrow (r_i, C_i)$
15: **end for**
16: **return** $\{hs_1, hs_2, \ldots, hs_q\}$

3.1.3 Geração Direcionada dos Exemplos

Após criadas as hiperesferas é necessário popular a hiperesfera principal HS com N pontos. As coordenadas desses pontos representam os valores dos atributos dos exemplos E_i da base de dados. Cada exemplo é representado pela dupla ordenada $(\mathbf{x}_i, Y_i)$, com $i = [1..N]$, onde $\mathbf{x}_i$ representa o vetor dos valores dos atributos e $Y_i \subseteq L$ o conjunto de rótulos do i-ésimo exemplo. Assim, para cada exemplo E_i é necessário gerar os valores $(x_{i1}, x_{i2}, \ldots, x_{iM})$ dos atributos, $i = [1..N]$, e o multirrótulo correspondente Y_i.

De forma a garantir que todos os q rótulos estejam presentes nos multirrótulos, a geração dos N exemplos é realizada de forma direcionada tal que os valores dos atributos $(x_{i1}, x_{i2}, \ldots, x_{iM})$ de cada exemplo E_i, ainda que gerados de forma aleatória, representem um ponto que pertence pelo menos a uma das hiperesferas menores, mantendo as devidas proporções. Sendo assim, o risco de uma das hiperesferas estar vazia é nulo, e, ao mesmo tempo, a possibilidade de um ponto não pertencer a nenhuma hiperesfera hs_i, $i = [1..q]$, é excluída. Em outras palavras, todos os N exemplos gerados são rotulados com multirrótulos não vazios.

Como os raios r_i de cada hiperesfera hs_i, $i = [1..q]$, são diferentes, é necessário garantir que a proporção de exemplos em cada hiperesfera seja mantida, ou seja, hiperesferas de raio maior devem conter proporcionalmente mais pontos que hiperesferas de raio menor. Para isso, é utilizado o seguinte fator de balanceamento f definido pela Equação 20.

$$f = \frac{N}{\sum_{i=1}^{q} r_i}. \tag{20}$$

Então, para cada hiperesfera hs_i é gerada uma quantidade de N_i pontos internos a ela, com $N_i = f \times r_i$. Fazendo os arredondamentos necessários se tem a quantidade de exemplos N_i que serão gerados, de forma aleatória, dentro da hiperesfera hs_i, $i = [1..q]$.

Seguindo a mesma ideia utilizada na geração dos centros das hiperesferas, para criar um ponto $\mathbf{x}_k = (x_{k1}, x_{k2}, \ldots, x_{kM})$ aleatório dentro de uma hiperesfera hs_i de centro C_i e raio r_i, deve-se garantir que

$$\sqrt{(x_{k1} - c_{i1})^2 + (x_{k2} - c_{i2})^2 + \ldots + (x_{kM} - c_{iM})^2} \leq r_i. \tag{21}$$

O que é equivalente a

$$(x_{k1} - c_{i1})^2 + (x_{k2} - c_{i2})^2 + \ldots + (x_{kM} - c_{iM})^2 \leq r_i^2. \tag{22}$$

$$\sum_{j=1}^{M} (x_{kj} - c_{ij})^2 \leq r_i^2. \tag{23}$$

Diferentemente do método de geração dos centros C_i, para um dado $i = [1..q]$, o domínio para geração de $\mathbf{x}_k$ passou a ser uma hiperesfera menor hs_i. Porém, hs_i não está centrado na origem, então é necessário levar em consideração a posição do seu centro. Na Figura 4 é ilustrado um exemplo para $M = 2$ que mostra que para gerar um ponto aleatório $\mathbf{x}_k$ dentro da hiperesfera hs_i com centro C_i e raio r_i, deve-se garantir que $\mathbf{x}_k$ esteja na área hachurada.

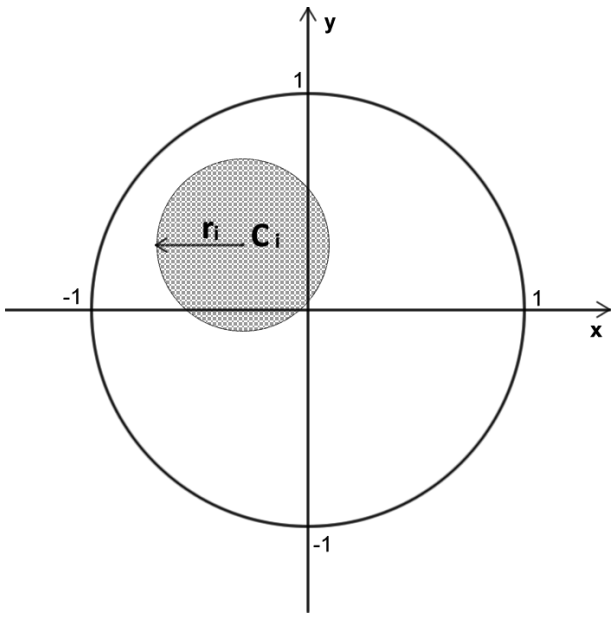

Figura 4: Domínio para geração de $\mathbf{x}_k$, dado r_i, para $M = 2$

Então, ao gerar aleatoriamente cada coordenada x_{k_j} de $\mathbf{x}_k$, $j = [1..M]$, é

necessário garantir que, $|x_{kj} - c_{ij}| \leq r_i$. Porém, analisando detalhadamente a geração aleatória de coordenadas, dado o caso que a primeira coordenada gerada seja $x_{k1} = c_{i1} + r_i$, então os outros valores de x_{kj}, para $j \neq 1$, devem ser necessariamente iguais a c_{ij}, para garantir que o ponto $\mathbf{x}_k$ seja interno à hs_i. Portanto, a geração de cada coordenada x_{kj} deve ser aleatória, mas levando em consideração as coordenadas anteriormente geradas.

Ao gerar cada coordenada x_{kj} de $\mathbf{x}_k$, dentro da hiperesfera hs_i, de forma aleatória com distribuição aproximadamente uniforme, $\forall j \in [1..M]$, deve-se restringir o intervalo conforme a Equação 24.

$$c_{ij} - \sqrt{r_i^2 - \sum_{s=1, s\neq j}^{M} (x_{ks} - c_{is})^2} \leq x_{kj} \leq c_{ij} + \sqrt{r_i^2 - \sum_{s=1, s\neq j}^{M} (x_{ks} - c_{is})^2}, \quad \forall i \in [1..q] \, e \, \forall k \in [1..N]. \tag{24}$$

Caso algum x_{ks}, para $s \neq j$, ainda não tenha sido instanciado, o seu valor não é considerado.

Analogamente à criação das hiperesferas, é necessário evitar determinismo na geração das coordenadas do ponto $\mathbf{x}_k$, ou seja, evitar que a coordenada x_{k1} sempre seja a primeira a ser gerada e x_{kM} a última. Isso porque os intervalos possíveis para a geração de cada coordenada x_{kj} diminuem a medida que uma nova coordenada é instanciada. Sendo assim, a ordem de instanciação das coordenadas é feita de forma aleatória, em outras palavras, j não varia em ordem crescente no intervalo de $[1..M]$, e sim de forma aleatória.

No Algoritmo 2 é apresentada de forma simplificada a geração dos N exemplos $\mathbf{x}_k$ dentro de cada uma das hiperesferas $hs_i = (r_i, C_i)$, para $i = [1..q]$. Nesse algoritmo é também utilizada a função $random(x, y)$ que gera um valor aleatório com distribuição aproximadamente uniforme entre x e y. As funções $atualizaminX(x)$ e $atualizamaxX(x)$ atualizam, respectivamente, o limite inferior e superior da próxima coordenada de $\mathbf{x}_k$ a ser instanciada dentro da hiperesfera hs_i correspondente, levando em consideração as coordenadas anteriormente geradas para geração das próximas coordenadas.

Após a geração dos N pontos, ou seja, N exemplos com M_{rel} atributos cada, adicionam-se para cada exemplo M_{irr} atributos irrelevantes, gerados a partir de valores randômicos, e M_{red} atributos redundantes, os quais replicam metade dos M atributos relevantes originais. A seleção de quais dos atributos relevantes serão replicados para gerar os atributos redundantes é também realizada de forma aleatória. Assim, cada exemplo do *CD* sintético apresenta ao final $M = M_{rel} + M_{irr} + M_{red}$ atributos.

Algoritmo 2 Algoritmo para geração dos exemplos nas hiperesferas

1: **for** $k = 1 \rightarrow N$ **do**
2: $\mathbf{x}_k \leftarrow \emptyset$
3: $maxX \leftarrow c_{ij} - r_i$
4: $minX \leftarrow c_{ij} + r_i$
5: **for** $j = 1 \rightarrow M$ $(sendo\, j\, escolhido\, aleatoriamente)$ **do**
6: $x_{kj} \leftarrow random(minX, maxX)$
7: $\mathbf{x}_k \leftarrow \mathbf{x}_k \cup \{x_{kj}\}$
8: $minX \leftarrow atualizaminX(minX)$
9: $maxX \leftarrow atualizamaxX(maxX)$
10: **end for**
11: **end for**
12: **return** $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$

3.1.4 Geração dos Multirrótulos

Após a geração dos valores dos atributos dos exemplos é preciso rotulá-los, ou seja, um exemplo com valores de atributos $\mathbf{x}_k$ contém o rótulo y_i no seu multirrótulo Y_k se tal ponto $\mathbf{x}_k$ for interno à hiperesfera hs_i.

Para isso é calculada a distância euclidiana de $\mathbf{x}_k$ até o centro C_i. Se essa distância for menor que o raio r_i, conclui-se que o exemplo $\mathbf{x}_k$ é interno à hs_i e $y_i \in Y_k$, caso contrário, $y_i \notin Y_k$. Ou seja, temos que $\forall k \in [1..N]$ $\mathbf{x}_k$ recebe o rótulo $y_i = 1$ se e somente se a Equação 25 for satisfeita,

$$\sqrt{(x_{kj} - c_{ij})^2} \leq r_i, \quad \forall i \in [1..q],\ \forall j \in [1..M],\ \forall k \in [1..N]. \tag{25}$$

O *CD* resultante, após a geração dos multirrótulos, é um conjunto de exemplos E_i, no formato atributo-valor, no qual cada exemplo está associado a um multirrótulo Y_i, com $i = [1..N]$, como mostrado na Tabela 1(b), página 4.

Após a geração do *CD* sintético, e com o intuito de obter uma melhor aproximação de dados reais, é gerado um segundo *CD* introduzindo um ruído μ no *CD* original. Esse ruído altera a presença ou ausência de um rótulo y_j, atribuído originalmente ao multirrótulo de um exemplo, com probabilidade μ. O programa tem como *default* um ruído $\mu = 0,05$ (5%). Como mencionado anteriormente, a saída da ferramenta consiste de um *CD* com M atributos com ruído e outro sem ruído. Quando $\mu = 0,0$ (0%) não é gerado o segundo *CD*.

A seguir é descrita a implementação da Ferramenta Hipercubos por nós proposta, baseada na proposta de Zhang et al. (2009), mas que utiliza hipercubos, no $\mathbb{R}^M$, para a geração do *CD* sintéticos multirrótulos.

3.2 Ferramenta Hipercubos

A principal motivação para o projeto e a implementação da Ferramenta Hipercubos é adaptar a estratégia proposta por Zhang et al. (2009), mas utilizando outra forma geométrica para representar a hipótese correspondente a cada rótulo que participa do multirrótulo. O uso de hipercubos para esse fim, é particularmente apropriada para avaliar algoritmos de *AM* multirrótulo que consideram a classificação dividindo o espaço utilizando hiperplanos, tais como árvores de decisão (Clare and King, 2001).

Semelhante à Ferramenta Hiperesferas, a Ferramenta Hipercubos, implementada em Java, recebe como entrada:

- M_{rel} - Número de atributos relevantes do conjunto de dados.
- M_{irr} - Número de atributos irrelevantes do conjunto de dados.
- M_{red} - Número de atributos redundantes do conjunto de dados.
- q - Número de rótulos do conjunto de dados.
- N - Número de exemplos do conjunto de dados.
- $maxR$ - Meia-aresta máxima dos hipercubos menores, *default* = 0,8.
- $minR$ - Meia-aresta mínima dos hipercubos menores, *default* = $\left\lfloor \frac{\frac{q}{10}+1}{q} \right\rfloor$.
- μ - Taxa de ruído que será inserida nos multirrótulos do *CD*, *default* = 0,05.

Foi implementada uma crítica dos dados, de modo a não permitir valores inválidos para os parâmetros de entrada. Para que as entradas sejam válidas deve-se garantir as mesmas condições descritas na Seção 3.1, página 11.

Da mesma maneira à ferramenta anterior, a saída desta ferramenta consiste de dois *CD* sintéticos com N exemplos e M atributos, tal que $M = M_{rel} + M_{irr} + M_{red}$, dos quais M_{red} são atributos redundantes e M_{irr} são atributos irrelevantes. Um dos *CD* contém ruído μ, enquanto o outro não contém ruído.

Assim como na implementação da Ferramenta Hiperesferas, a Ferramenta Hipercubos difere do que foi proposto no trabalho de Zhang et al. (2009), visto que também permite que o usuário escolha a quantidade de atributos relevantes, redundantes e irrelevantes do *CD*, além.

Para a geração dos *CD*, são realizados, de uma forma sucinta, os seguintes passos:

1. Geração do hipercubo principal no $\mathbb{R}^M$;

2. Geração de q hipercubos no $\mathbb{R}^M$, internos ao hipercubo principal;

3. Geração de N pontos no $\mathbb{R}^M$;

4. Geração dos multirrótulos, segundo os hipercubos menores;

5. Inserção do ruído μ.

Esses passos são descritos e detalhados a seguir.

3.2.1 Geração do Hipercubo Principal

Inicialmente é gerado um hipercubo HC no $\mathbb{R}^M$, com centro na origem do plano cartesiano e meia-aresta $e_{HC} = 1$, ilustrado na Figura 5, onde no eixo x está representado o atributo $\mathbf{X}_1$ e no eixo y o atributo $\mathbf{X}_2$ do conjunto de dados multirrótulos, para $M = 2$.

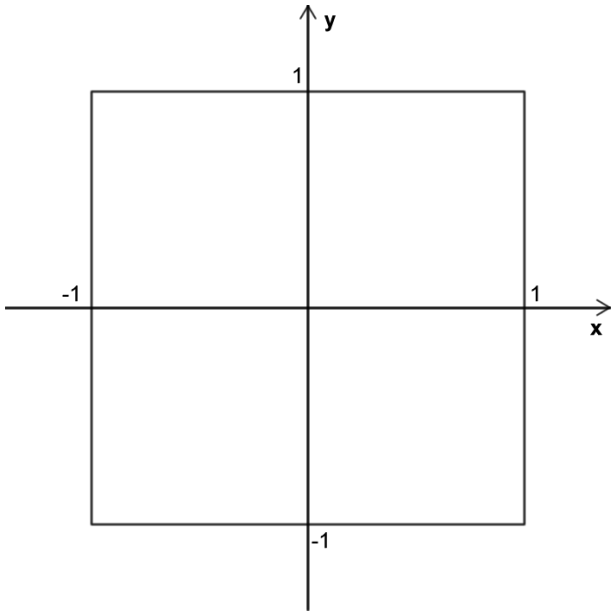

Figura 5: Hipercubo principal HC, para $M = 2$

3.2.2 Geração dos Hipercubos contidos no Hipercubo Principal

Semelhante à implementação da Ferramenta Hiperesferas, são gerados, no $\mathbb{R}^M$, q hipercubos menores contidos no hipercubo principal. O conjunto de hipercubos menores, $H_{hc} = \{hc_1, hc_2, \ldots, hc_q\}$, no qual cada hipercubo $h_{ci} = (e_i, C_i)$ é caracterizado pelo valor de metade de sua aresta, denotada por e_i, e a posição geométrica de seu centro $C_i = (c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iM})$.

Para a geração de cada hipercubo menor hc_i, primeiramente é gerado um valor de meia-aresta e_i aleatório com distribuição aproximadamente uniforme, entre *minE* e *maxE*. Em seguida é necessário gerar aleatoriamente os valores

das coordenadas do centro C_i, de forma a garantir que o hipercubo menor hc_i esteja contido na hipercubo principal, conforme a condição 26

$$\forall\, i \in [1..q]\ c_{ij} \leq (1 - e_i), j = 1..M \tag{26}$$

Diferentemente da Ferramenta Hipercubos, o fato de verificar a condição 26 garante que o centro C_i dos hipercubos menores satisfaz a condição de hc_i estar contida em HC. Portanto, o domínio das coordenadas do centro C_i é determinado pela condição 26. Na Figura 6 é mostrado um exemplo gráfico para $M = 2$, no qual foi gerada uma meia-aresta aleatória e_i e a única forma de garantir que o hipercubo menor esteja contido no hipercubo principal é que seu centro C_i esteja na área hachurada.

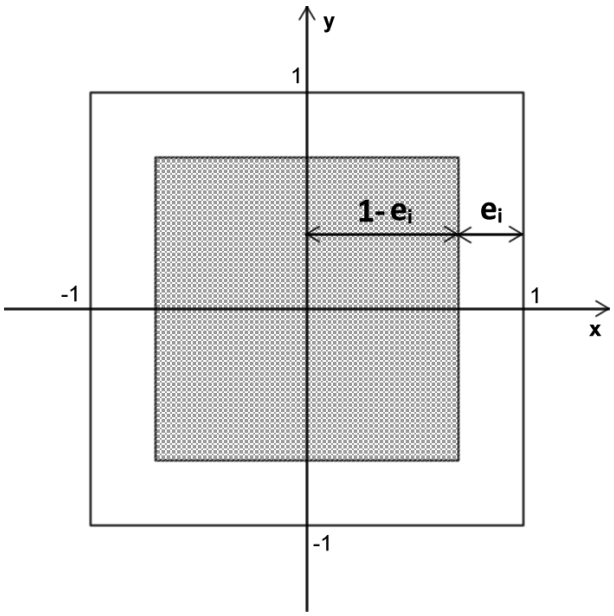

Figura 6: Domínio para geração de C_i, dado e_i, para $M = 2$

Diferentemente da ferramente Hiperesferas, o intervalo para a geração de cada coordenada do centro C_i para a geração de um hipercubo menor não varia, ou seja, ele se mantém o mesmo desde a primeira coordenada até a última. No Algoritmo 3 é apresentado de forma simplificada como é realizada a geração dos hipercubos $hc_i = (e_i, C_i)$, para $i = [1..q]$. A função $random(x, y)$ é a mesma utilizada no Algoritmo 1, página 15, *i.e.*, gera um valor aleatório, com distribuição aproximadamente uniforme, entre x e y.

3.2.3 Geração Direcionada dos Exemplos

Após criados os hipercubos, é necessário popular o hipercubo principal HC com N pontos. De forma análoga à Ferramenta Hiperesferas, esses pontos representam os valores dos atributos dos exemplos E_i da base de dados. Cada

Algoritmo 3 Algoritmo para geração dos hipercubos menores

1: $hc_i \leftarrow \emptyset$
2: $C_i \leftarrow \emptyset$
3: $e_i \leftarrow 0$
4: **for** $i = 1 \rightarrow q$ **do**
5: $\quad maxC \leftarrow (1 - r_i)$
6: $\quad minC \leftarrow -(1 - r_i)$
7: $\quad e_i \leftarrow random(minR, maxR)$
8: $\quad$ **for** $j = 1 \rightarrow M$ **do**
9: $\quad\quad c_{ij} \leftarrow random(minC, maxC)$
10: $\quad\quad C_i \leftarrow C_i \cup \{c_{ij}\}$
11: $\quad$ **end for**
12: $\quad hc_i \leftarrow (e_i, C_i)$
13: **end for**
14: **return** $\{hc_1, hc_2, \ldots, hc_q\}$

exemplo é representado pela dupla ordenada $(\mathbf{x}_i, Y_i)$, com $i = [1..N]$, onde $\mathbf{x}_i$ representa o vetor dos valores dos atributos e $Y_i \subseteq L$ o conjunto de rótulos do i-ésimo exemplo.

Assim como descrito na Seção 3.1.3, é necessário garantir que todos os q rótulos estejam presentes nos multirrótulos. Para isso, a geração dos N exemplos é realizada de forma direcionada tal que os valores dos atributos $(x_{i1}, x_{i2}, \ldots, x_{iM})$ de cada exemplo E_i, ainda que gerados de forma aleatória, representem um ponto que pertence pelo menos a um dos hipercubos menores, mantendo as devidas proporções.

Para garantir que a proporção de exemplos em cada hipercubo seja mantida é utilizado um fator de balanceamento f, semelhante à Equação 20, página 15, mas com a diferença que ao invés de somar os raios no denominador, nesse caso serão somadas as meias-arestas e_i. Então, para cada hipercubo hc_i é gerada uma quantidade de N_i pontos internos a ele, com $N_i = f \times e_i$. Fazendo os arredondamentos necessários tem-se a quantidade de exemplos N_i que são gerados, de forma aleatória, dentro do hipercubo hc_i, $i = [1..q]$.

Seguindo a mesma ideia utilizada na geração dos centros dos hipercubos, para criar um ponto $\mathbf{x}_k = (x_{k1}, x_{k2}, \ldots, x_{kM})$ aleatório dentro de um hipercubo hc_i de centro C_i e meia-aresta e_i, deve-se garantir que a condição 27 seja satisfeita.

$$|(x_{kj} - c_{ij})| \leq e_i, \forall i \in [1..q] \, e \, \forall j \in [1..M]. \tag{27}$$

Diferentemente do método de geração dos centros C_i, para um dado i, o domínio para geração de $\mathbf{x}_k$ passou a ser um hipercubo menor hc_i. Porém, hc_i não está centrado na origem, então é necessário levar em consideração a posição do seu centro. Na Figura 7 é ilustrado um exemplo para $M = 2$

que mostra que para gerar um ponto aleatório $\mathbf{x}_k$ dentro do hipercubo hc_i com centro C_i e meia-aresta e_i, deve-se garantir que $\mathbf{x}_k$ esteja na área hachurada.

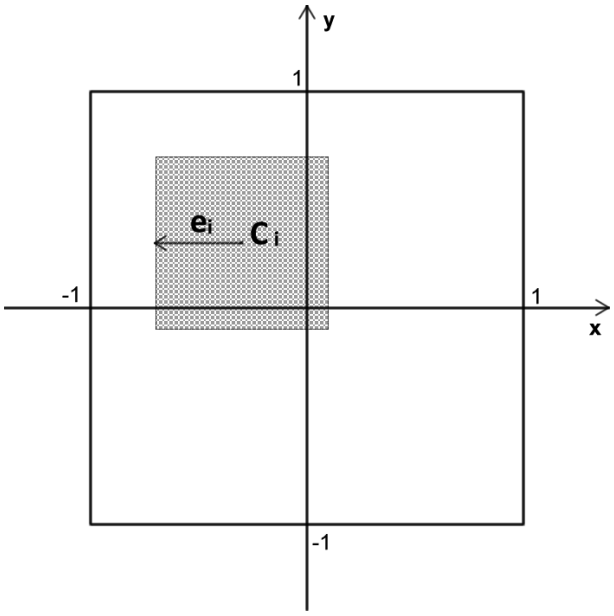

Figura 7: Domínio para geração de $\mathbf{x}_k$, dado e_i, para $M = 2$

Ao gerar cada coordenada x_{kj} de $\mathbf{x}_k$, dentro do hipercubo hc_i, de forma aleatória com distribuição aproximadamente uniforme $\forall j \in [1..M]$, deve-se restringir o intervalo conforme a condição 27.

No Algoritmo 4 é apresentada de forma simplificada a geração dos N exemplos $\mathbf{x}_k$ dentro de cada um dos hipercubos $hc_i = (e_i, C_i)$, para $i = [1..q]$. Nesse algoritmo é também utilizada a função $random(x, y)$ que gera um valor aleatório com distribuição aproximadamente uniforme entre x e y. Vale lembrar que como o intervalo para a geração de cada coordenada do exemplo $\mathbf{x}_i$ não varia, ou seja, ele se mantém o mesmo desde a primeira coordenada até a última, não é necessário o uso das funções $atualizaminX(x)$ e $atualizamaxX(x)$ como no caso das hiperesferas.

Algoritmo 4 Algoritmo para geração dos exemplos nos hipercubos

1: **for** $k = 1 \rightarrow N$ **do**
2: $\quad \mathbf{x}_k \leftarrow \emptyset$
3: $\quad maxX \leftarrow c_{ij} - e_i$
4: $\quad minX \leftarrow c_{ij} + e_i$
5: $\quad$ **for** $j = 1 \rightarrow M$ **do**
6: $\quad\quad x_{kj} \leftarrow random(minX, maxX)$
7: $\quad\quad \mathbf{x}_k \leftarrow \mathbf{x}_k \cup \{x_{kj}\}$
8: $\quad$ **end for**
9: **end for**
10: **return** $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$

Analogamente à ferramenta Hiperesferas, após a geração dos N pontos,

ou seja, N exemplos com M_{rel} atributos cada, adicionam-se para cada exemplo M_{irr} atributos irrelevantes, gerados a partir de valores randômicos, e M_{red} atributos redundantes, os quais replicam metade dos M atributos relevantes originais. A seleção de quais dos atributos relevantes serão replicados para gerar os atributos redundantes é também realizada de forma aleatória. Assim, cada exemplo do *CD* sintético apresenta ao final $M = M_{rel} + M_{irr} + M_{red}$ atributos.

3.2.4 Geração dos Multirrótulos

Semelhante à ferramenta Hiperesferas, após a geração dos valores dos atributos dos exemplos é preciso rotulá-los, ou seja, um exemplo com valores de atributos $\mathbf{x}_k$ contém o rótulo y_i no seu multirrótulo Y_k se tal ponto $\mathbf{x}_k$ for interno ao hipercubo hc_i.

Para isso é calculada a distância de $\mathbf{x}_k$ até o centro C_i, se essa distância for menor que o valor de meia-aresta e_i, conclui-se que o exemplo $\mathbf{x}_k$ é interno à hc_i e $y_i \in Y_k$, caso contrário, $y_i \notin Y_k$. Ou seja, temos que $\forall k \in [1..N]$ $\mathbf{x}_k$ recebe o rótulo $y_i = 1$ se e somente se a Equação 28 for satisfeita.

$$|(x_{kj} - c_{ij})| \leq e_i, \forall i \in [1..q] \, e \, \forall j \in [1..M]. \tag{28}$$

De forma análoga à ferramenta Hiperesferas, após a geração do *CD* sintético, e com o intuito de obter uma melhor aproximação de dados reais, é gerado um segundo *CD* introduzindo um ruído μ ao *CD* original. Esse ruído altera a presença ou ausência de um rótulo y_j, atribuído originalmente ao multirrótulo de um exemplo, com probabilidade μ. O programa tem como *default* um ruído $\mu = 0,05$ (5%). Quando $\mu = 0,0$ (0%) não é gerado o segundo *CD*.

3.3 O *Framework*

As duas ferramentas descritas nas Seções 3.1 e 3.2 estão integradas em um *framework*, que está disponibilizado para a comunidade em `http://sites.labic.icmc.usp.br/mldatagen/`. Ao acessar a página, o usuário encontra uma breve descrição do *framework* e de sua utilização, como apresentado na Figura 8.

Ao descer a página, o usuário tem acesso aos campos de configuração. O usuário deve preencher todos os dados e clicar em "Gerar". Na Figura 9 é apresentado a pagina de configuração do *framework*.

Uma vez que todos os campos estejam devidamente preenchidos o sistema realiza uma crítica de dados estática, segundo descrito na Seção 3.1, na página 11. Caso algum campo possua valores inválidos o sistema lança

Framework para a Geração de Conjunto de Dados Sintéticos para Aprendizado Multirrótulo

O framework, descrito em detalhes em um relatório técnico do ICMC-USP, permite a geração de Conjuntos de Dados (CD) sintéticos multirrótulo utilizando duas estratégias: hiperesferas ou hipercubos. Basicamente, essas estratégias geram randomicamente uma forma geométrica (hiperesfera ou hipercubo) para cada rótulo do CD, as quais são preenchidas com pontos (exemplos) gerados randomicamente. Após, cada exemplo é rotulado de acordo com a forma geométrica a qual pertence, o que determina o multirrótulo do exemplo.

Após escolher a estratégia a ser utilizada, devem ser definidos alguns parâmetros obrigatórios: número de atributos relevantes, número de atributos irrelevantes, número de atributos redundantes, quantidade de rótulos e quantidade de exemplos do CD. Também podem ser definidos alguns parâmetros opcionais, que possuem valores default: tamanho máximo e mínimo das hiperesferas/hipercubos menores, taxa de ruído e nome do CD.

A saída consiste de dois CD sintéticos, um sem ruído e outro com ruído, gerados pelo framework no formato requerido pelo Mulan, uma biblioteca que disponibiliza diversos métodos para o aprendizado multirrótulo. Esse formato envolve dois arquivos: um do tipo ARFF e outro do tipo XML, os quais podem ser diretamente utilizados como entrada para o Mulan.

Para gerar um CD sintético multirrótulo, configure os parâmetros abaixo e clique em "Gerar". Em seguida, clique em "Baixar CD Gerado" para fazer o download dos arquivos.

Figura 8: Página incial do *framework*

uma mensagem de erro, informando ao usuário qual dos parâmetros de entrada possui um valor não válido.

Uma vez que todos os parâmetros tenham sido preenchidos corretamente, o usuário é direcionado para uma página de saída, Figura 10, na qual há um *prompt* onde são impressos alguns dados importantes sobre a geração do *CD* como informações de cada uma das formas geométricas criadas. Para a Ferramenta Hiperesferas são mostradas as coordenadas dos centros de cada uma das hiperesferas menores e os valores de seus raios. De forma análoga, para a Ferramenta Hipercubos ao invés dos valores dos raios, são mostrados os valores das meia-arestas de cada hipercubo menor. Também é apresentado o balanceamento dos exemplos, *i.e.*, quantos exemplos foram criados dentro de cada hiperesfera/hipercubo, conforme descrito na Seção 3.1.3, página 15, e Seção 3.2.3, página 21.

Para fazer o *download* do *CD* gerado basta clicar em "Baixar *CD* Gerado". Será iniciado o *download* de um arquivo `<x>_<y>_<w>_<z>.zip`, no qual `<x>` é a ferramenta escolhida (Hiperesfera ou Hipercubos), enquanto que `<y>`, `<w>` e `<z>` são números que representam, respectivamente, a quantidade de atributos relevantes, irrelevantes e redundantes solicitados pelo usuário. Por exemplo, se o *framework* foi configurado com a Ferramenta Hiperesferas, 10 atributos relevantes, 5 atributos irrelevantes e 5 atributos redundantes, será gerado um arquivo `Hiperesferas_10_5_5.zip` contendo dois arquivos (ARFF

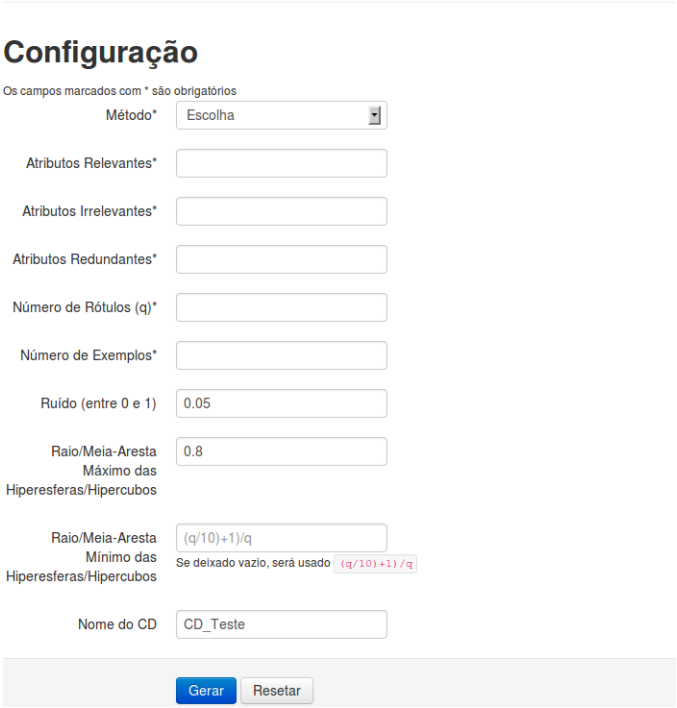

Figura 9: Página de configuração do *framework*

e XML) sem ruído e outros dois arquivos (ARFF e XML) com ruído.

4 Exemplo de Uso

Utilizando o *framework* implementado foram gerados 4 *CD*, 2 gerados com a ferramenta Hiperesferas e 2 com a ferramenta Hipercubos. Para geração desses 4 *CD* variou-se a quantidade de atributos relevantes M_{rel}, M_{irr} e M_{red} mas garantindo que M seja o mesmo para todos os conjuntos.

Foram gerados mais 2 *CD*, um com a Ferramenta Hiperesferas e o outro com a Ferramenta Hipercubos, com um número menor de atributos, visando analisar como a dimensionalidade pode influenciar nos resultados. A quantidade de rótulos simples q e a quantidade N de exemplos foi mantida constante em todos os 6 *CD*.

Para avaliar experimentalmente esses *CD*, foi utilizado o algoritmo de aprendizado multirrótulo *BRkNN-b* (Spyromitros et al., 2008), disponível no Mulan, que será brevemente introduzido na Seção 4.2.

4.1 Descrição dos Conjuntos de Dados

Todos os *CD* gerados possuem as seguintes características:

- $q = 10$ rótulos

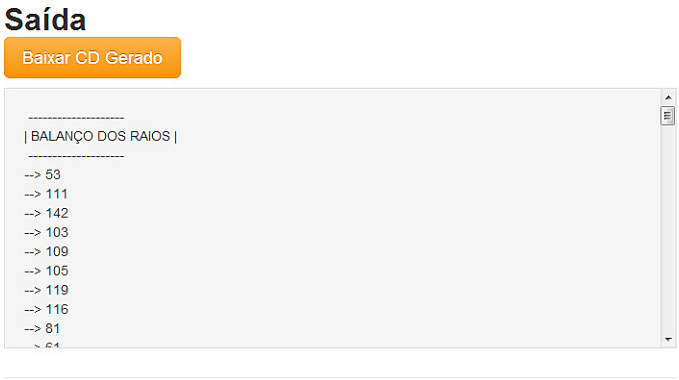

Figura 10: Página de saída do *framework*

- $N = 1000$ exemplos

Porém, com o objetivo de avaliar a influência dos diferentes tipos de atributos, nos 4 primeiros *CD* variou-se a quantidade de atributos relevantes M_{rel}, de atributos irrelevantes M_{irr} e de atributos redundantes M_{red}, garantindo que a soma total de atributos M seja constante e igual a 20.

Com o objetivo de analisar o comportamento da hiperesfera principal quando esta atinge seu volume máximo[12], os 2 *CD* restantes foram gerados com apenas 5 atributos relevantes, *i.e.*, $M_{rel} = 5$, $M_{irr} = 0$, $M_{red} = 0$. A configuração dos atributos de cada *CD* é descrita na Tabela 4.

Tabela 4: Distribuição dos atributos nos *CD* sintéticos

CD	Ferramenta	M_{rel}	M_{irr}	M_{red}	M
CD_A	Hiperesferas	20	0	0	20
CD_B	Hipercubos	20	0	0	20
CD_C	Hiperesferas	10	5	5	20
CD_D	Hipercubos	10	5	5	20
CD_E	Hiperesferas	5	0	0	5
CD_F	Hipercubos	5	0	0	5

4.2 Algoritmo Multirrótulo Utilizado

Para avaliar experimentalmente os *CD*, gerados pela ferramenta implementada, foi utilizado o algoritmo de aprendizado *BRkNN* (Spyromitros et al., 2008). O algoritmo *BRkNN* é uma adaptação do algoritmo *kNN* para classificação multirrótulo que utiliza o método de transformação de problema *Binary Relevance*, o qual transforma o problema multirrótulo em um ou mais problemas monorrótulos. Após essa transformação, utiliza o algoritmo *kNN* para a

[12] http://divisbyzero.com/2010/05/09/volumes-of-n-dimensional-balls/

resolução de cada problema monorrótulo separadamente. A idéia desse algoritmo é encontrar os *k* exemplos rotulados mais similares ao exemplo a ser classificado e, com base nos rótulos desses exemplos, é determinado o rótulo do exemplo a ser classificado. Logo, os resultados são posteriormente combinados para a solução final do problema multirrótulo.

Spyromitros et al. (2008) propõem duas extensões para melhorar o poder de predição do *BRkNN*: *BRkNN*-a e *BRkNN*-b, respectivamente. Ambas as extensões consideram um valor de confiança de cada rótulo para decidir os que farão parte do multirrótulo predito, a saber: a média do número de ocorrências dos rótulos nos *k* exemplos. Para o *BRkNN*-a, o procedimento para seleção dos rótulos é o mesmo do *BRkNN*: caso a confiança seja superior ao *threshold* 0,5, o rótulo é considerado parte do multirrótulo predito. Entretanto, esse procedimento de seleção de rótulos pode predizer multirrótulos vazios. Com o objetivo de anular essa possibilidade, caso nenhum rótulo seja selecionado, o de maior confiança é considerado parte do multirrótulo pelo *BRkNN*-a. Por outro lado, o *BRkNN*-b prevê antecipadamente quantos serão os rótulos que participarão do multirrótulo predito. A quantidade s de rótulos é a média aritmética arredondada das quantidades de rótulos dos k exemplos mais próximos. Assim, os que farão parte do multirrótulo serão os s de maior confiança.

Na avaliação realizada neste trabalho foi utilizado o algoritmo *BRkNN* com a extensão tipo b, $k = 11$ e os demais parâmetros com valores *default*, ou seja, a função de peso para a rotulação de exemplos é 1 para todos os k vizinhos mais proximos e a distância de cada vizinho mais próximo ao exemplo a ser rotulado também não interfere na estimativa do número de rótulos do multirrótulo predito (dos Reis et al., 2012). Todos os experimentos relacionados à avaliação dos classificadores construídos utilizando os seis *CD* artificiais foram realizados utilizando o método de avaliação 5x2-*fold cross-validation*, o qual realiza 5 replicações do método 2-*fold cross-validation* que divide um *CD* em dois conjuntos iguais para cada uma das execuções do algoritmo.

4.3 Resultados e Discussão

Para cada um dos *CD* gerados foram analisadas, em porcentagem, a frequência de ocorrência de cada rótulo e quantidade de exemplos que possuem uma certa quantidade de rótulos, podendo essa quantidade variar de 1 a 10. Na Tabela 5 estão apresentadas as frequências de cada um dos monorótulos.

Na Tabela 6 estão apresentados os valores de cardinalidade e densidade de rótulo de cada *CD*.

Na Tabela 7 estão apresentados as porcentagens de multirrótulos, de cada *CD*, com cardinalidade de rótulo 1 até 10.

Tabela 5: Frequência dos monorótulos

Rótulo/CD	CD_A	CD_B	CD_C	CD_D	CD_E	CD_F
$y1$	$6,50\%$	$38,20\%$	$55,30\%$	$10,90\%$	$28,40\%$	$4,00\%$
$y2$	$47,60\%$	$13,00\%$	$6,30\%$	$9,70\%$	$27,40\%$	$63,30\%$
$y3$	$24,90\%$	$9,40\%$	$8,80\%$	$22,50\%$	$72,90\%$	$36,10\%$
$y4$	$6,90\%$	$9,70\%$	$58,80\%$	$6,70\%$	$72,60\%$	$64,80\%$
$y5$	$50,00\%$	$15,50\%$	$7,80\%$	$55,30\%$	$44,60\%$	$81,00\%$
$y6$	$5,70\%$	$20,90\%$	$17,10\%$	$5,30\%$	$21,00\%$	$18,70\%$
$y7$	$15,10\%$	$34,20\%$	$17,70\%$	$7,50\%$	$29,40\%$	$46,10\%$
$y8$	$4,80\%$	$7,70\%$	$46,80\%$	$50,40\%$	$25,80\%$	$5,50\%$
$y9$	$22,10\%$	$6,60\%$	$11,60\%$	$23,60\%$	$57,70\%$	$7,60\%$
$y10$	$4,90\%$	$9,70\%$	$6,90\%$	$31,90\%$	$19,70\%$	$32,60\%$

Tabela 6: Valores de cardinalidade e densidade de rótulo

CD	Cardinalidade	Densidade
CD_A	1,885	0,188
CD_B	1,649	0,165
CD_C	2,371	0,237
CD_D	2,238	0,224
CD_E	3,995	0,399
CD_F	3,597	0,360

Pode ser observado nos primeiros quatro *CD* de dimensionalidade 20, que quanto maior a cardinalidade menor o número de multirrótulos gerados com essa cardinalidade. No caso do CD_A e CD_B, não foram criados multirrótulos com cardinalidade superior a 6, enquanto que para CD_C e CD_E não foram criados multirrótulos com cardinalidade superior a 7. No caso dos CD_E e CD_F de dimensionalidade 5, pode ser observado que até a cardinalidade 4 não se verifica que o número de multirrótulos gerados diminui monotonicamente com a cardinalidade do multirrótulo, mas isso acontece a partir de mutirrótulos com cardinalidade maior que 4. Uma possível explicação está relacionada com a *maldição da dimensionalidade*, a qual refere-se a fenômenos que acontecem em *CD* de alta dimensão, mas não em *CD* de baixa dimensão. O efeito da maldição da dimensionalidade é ainda um problema em aberto para a comunidade científica.

Após, foram gerados os classificadores para cada um dos *CD*. Como mencionado, essa avaliação foi realizada utilizando o método 5x2-*fold cross-validation*. Os valores médios das medidas de avaliação consideradas[13], juntamente

[13]Pode ser observado que não todas as medidas definidas Seção 2.2.2, página 5 foram consideradas. Isso porque em alguns casos obtem-se NaN (Not a Number). Por exemplo, para F-Meassure, caso Precision AND Recall sejam zero simultaneamente em algum dos *folds*, há uma divisão por zero que compromete o cálculo do valor médio. Vale observar que na versão de Mulan estava erradamente a condição Precision OR Recall, a qual foi substituída na nossa versão pela condição válida. Esse problema, bem como outros relacionados as medidas de avaliação multirrótulo, já foram reportados ao grupo responsável pelo desenvolvimento do Mulan, grupo com o qual os pesquisadores do LABIC mantém estreito contato.

Tabela 7: Porcentagem de multirrótulos gerados com cardinalidade de rótulo 1 até 10

CR/CD	1	2	3	4	5	6	7	8	9	10
CD_A	45,90%	30,90%	14,40%	6,90%	1,40%	0,50%	0,00%	0,00%	0,00%	0,00%
CD_B	58,60%	25,80%	9,80%	3,90%	1,70%	0,20%	0,00%	0,00%	0,00%	0,00%
CD_C	33,70%	25,70%	20,00%	13,30%	5,20%	1,90%	0,20%	0,00%	0,00%	0,00%
CD_D	37,90%	29,00%	15,20%	10,40%	4,70%	2,40%	0,70%	0,00%	0,00%	0,00%
CD_E	16,20%	16,60%	12,50%	16,40%	11,10%	10,80%	8,20%	5,10%	2,50%	0,60%
CD_F	14,50%	16,70%	20,30%	16,90%	14,20%	10,40%	6,10%	0,90%	0,00%	0,00%

com o desvio padrão entre parêntese, encontram-se na Tabela 8. Os melhores resultados estão destacados em cinza e foram todos obtidos com o CD_F. Vale observar que, exceto para as medidas Hamming Loss e Recall, os segundos melhores resultados foram obtidos com o CD_E. Ou seja, foi possível gerar melhores classificadores com os *CD* com menor dimensionalidade.

Tabela 8: Resultados da avaliação experimental

	Medidas Baseadas em Exemplos				
	Haming Loss	*Subset Accuracy*	*Precision*	*Recall*	*Accuracy*
CD_A	0,195 (0,002)	0,058 (0,006)	0,476 (0,006)	0,542 (0,013)	0,368 (0,004)
CD_B	0,155 (0,005)	0,123 (0,011)	0,529 (0,014)	0,749 (0,017)	0,472 (0,012)
CD_C	0,200 (0,005)	0,077 (0,013)	0,553 (0,019)	0,660 (0,006)	0,453 (0,012)
CD_D	0,165 (0,002)	0,101 (0,008)	0,602 (0,010)	0,813 (0,016)	0,521 (0,004)
CD_E	0,186 (0,005)	0,124 (0,014)	0,676 (0,012)	0,802 (0,015)	0,606 (0,010)
CD_F	0,120 (0,006)	0,228 (0,023)	0,759 (0,014)	0,929 (0,007)	0,709 (0,012)

	Medidas Baseadas em Rótulos			
	Micro Precision	*Micro Recall*	*Micro F-Measure*	*Macro Recall*
CD_A	0,486 (0,007)	0,565 (0,012)	0,522 (0,004)	0,292 (0,012)
CD_B	0,523 (0,015)	0,714 (0,018)	0,603 (0,013)	0,729 (0,015)
CD_C	0,566 (0,019)	0,685 (0,009)	0,620 (0,010)	0,362 (0,009)
CD_D	0,603 (0,010)	0,764 (0,020)	0,674 (0,005)	0,750 (0,021)
CD_E	0,732 (0,015)	0,845 (0,009)	0,784 (0,008)	0,781 (0,012)
CD_F	0,788 (0,013)	0,913 (0,008)	0,846 (0,008)	0,890 (0,012)

Para apresentar graficamente os resultados da Tabela 8, foram gerados gráficos radar para todas as medidas de avaliação que devem ser maximizadas (*i.e.* todas as medidas exceto *HammingLoss*) por meio do *framework* R[14]. Na Figura 11 estão apresentados os gráficos radares para os CD_A, CD_B, CD_C e CD_D, e na Figura 12, os gráficos radares para os CD_E e CD_F. Cada linha colorida representa o desempenho do classificador gerado utilizando cada um dos *CD* nas medidas de avaliação, enquanto que cada eixo representa uma medida de avaliação.

Analisando os resultados, é possível observar que os *CD* criados utilizando hipercubos apresentam melhores resultados que utilizando hiperesferas. Em

[14] http://www.r-project.org

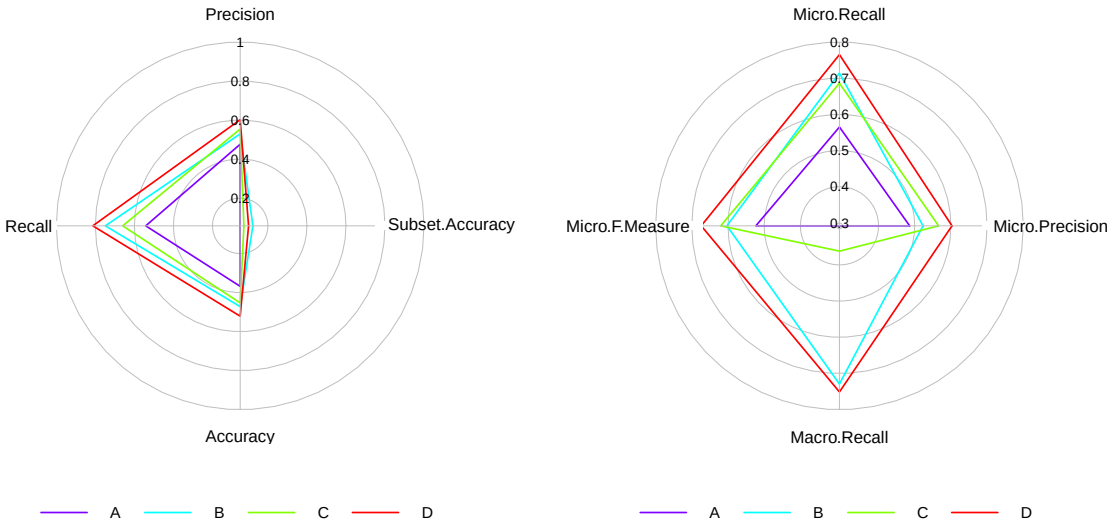

Figura 11: Gráficos radares dos CD_A, CD_B, CD_C e CD_D.

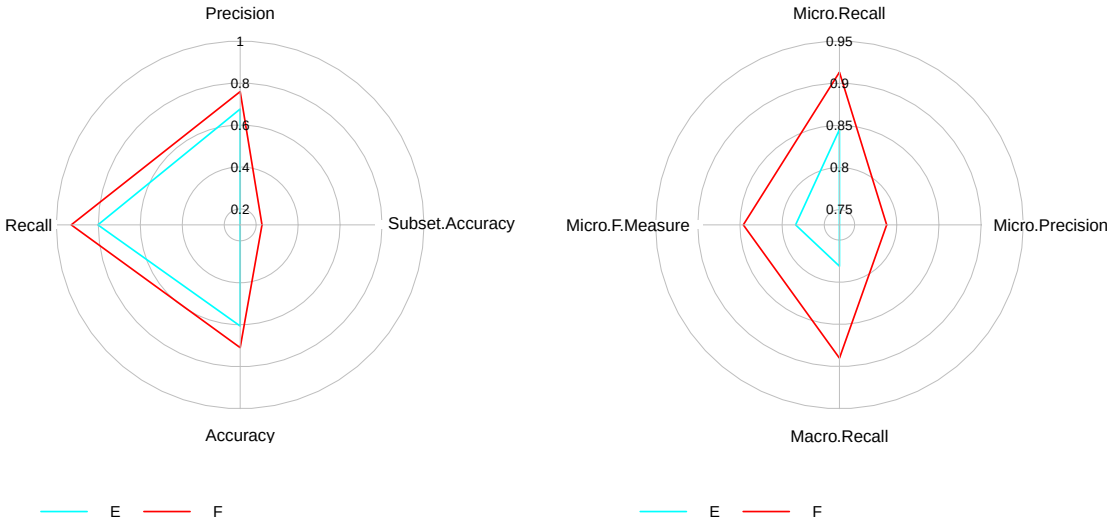

Figura 12: Gráficos radares dos CD_E e CD_F.

outras palavras, os classificadores gerados utilizando os CD_B e CD_D obtêm melhores resultados que utilizando os CD_A e CD_C respectivamente.

Um dos motivos é que o volume da hiperesfera principal cresce inicialmente com a dimensionalidade do *CD* até atingir um máximo valor para uma determinada dimensionalidade, a qual depende do volume inicial da hiperesfera. Após, o volume da hiperesfera decresce com a dimensionalidade e o volume tende a zero. Por exemplo, na ferramenta implementada, o raio da hiperesfera é 1. Nesse caso, a hiperesfera atinge seu volume máximo[15] na dimensão $M = 5$. Por outro lado, o volume do hipercubo cresce exponencialmente com a dimensionalidade. Ambos problemas estão relacionado com a maldição da dimensionalidade.

5 Conclusão

Neste trabalho foram estudadas diversas estratégias propostas na literatura para geração de *CD* sintéticos multirrótulo. Adicionalmente, estratégias e ferramentas utilizadas no aprendizado monorrótulo foram pesquisadas, apoiando eventuais extensões para aprendizado multirrótulo.

Com base na estratégia abordada no trabalho de Zhang et al. (2009) foram projetadas e implementadas duas ferramentas computacionais, Hiperesferas e Hipercubos. A partir da definição de parâmetros como M_{rel}, número de atributos relevantes do *CD*; M_{irr}, número de atributos irrelevantes do *CD*; M_{red}, número de atributos redundantes do *CD*; q, quantidade de rótulos do CD e N, quantidade de exemplos do *CD*, essas ferramentas geram um *CD* multirrótulo com $M = M_{rel} + M_{irr} + M_{red}$ atributos, no qual cada exemplo está rotulado com um multirrótulo cuja cardinalidade máxima é q. As ferramentas ainda possibilitam a inserção de um ruído μ, com o intuito de aumentar a aproximação de dados reais, gerando assim um segundo *CD*.

Cada *CD* sintético (com ou sem ruído) gerado pelas ferramentas está no formato Mulan[16], que envolve dois arquivos, dos quais um é do tipo ARFF e o outro é do tipo XML. Esses arquivos que descrevem o *CD* sintético gerado podem ser diretamente utilizados como entrada para o Mulan, uma biblioteca que disponibiliza diversos métodos para o aprendizado multirrótulo, a qual é muito utilizada pela comunidade. Ambas ferramentas foram integradas em um *framework*, no qual podem ser futuramente inseridas outras ferramentas para geração de CD multirrótulo.

Foi apresentado um exemplo de uso do *framework*, com o intuito de ilustrar a utilização das ferramentas para a geração de bases de dados multirrótulo.

[15] http://divisbyzero.com/2010/05/09/volumes-of-n-dimensional-balls/
[16] http://mulan.sourceforge.net

Analisando os resultados obtidos após gerar os classificadores *BRkNN* para cada um dos *CD*, é possível observar que os *CD* criados por meio da Ferramenta Hipercubos apresentam melhores resultados que os da Ferramenta Hiperesferas.

Além disso, a análise da influência da dimensionalidade sugere que ambas estratégias sofrem com a maldição da dimensionalidade. Um dos motivos é que o volume da hiperesfera principal cresce inicialmente com a dimensionalidade do *CD*, até atingir um valor máximo, e depois tende a zero. Enquanto isso, o volume do hipercubo cresce exponencialmente com a dimensionalidade.

Referências

Agrawal, R., Ghosh, S., Imielinski, T., Iyer, B., and Swami, A. (1992). An interval classifier for database mining applications. In *International Conference on Very Large Databases*, pages 560–573. Citado na página 9.

Alpaydin, E. (2010). *Introduction to Machine Learning*. MIT Press. Citado nas páginas 1 e 2.

Chou, S. and Hsu, C.-L. (2005). MMDT: a multi-valued and multi-labeled decision tree classifier for data mining. *Expert Systems with Applications*, 28(4):799–812. Citado nas páginas 2 e 9.

Clare, A. and King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–53. Citado na página 19.

Dietterich, T. G. (1990). Exploratory research in machine learning. *Machine Learning*, 5(1):5–10. Citado na página 1.

dos Reis, D. M., Cherman, E. A., Spolaôr, N., and Monard, M. C. (2012). Extensões do algoritmo de aprendizado de máquina multirrótulo BRkNN. In *Encontro Nacional de Inteligência Artificial*, pages 1–12. Citado na página 28.

Godbole, S. and Sarawagi, S. (2004). Discriminative Methods for Multi-labeled Classification. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Citado na página 6.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The Weka data mining software: An update. *SIGKDD Explorations*, 11(1):10–18. Citado na página 10.

Langley, P. (2000). Crafting papers on machine learning. In *International Conference on Machine Learning*, pages 1207–1212. Citado na página 1.

Liu, H. and Motoda, H. (2008). *Computational Methods of Feature Selection*. Chapman & Hall/CRC. Citado na página 9.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Education (ISE Editions). Citado nas páginas 1 e 2.

Noh, H. G., Song, M. S., and Park, S. H. (2004). An unbiased method for constructing multilabel classification trees. *Computational Statistics & Data Analysis*, 47(1):149–164. Citado nas páginas 2 e 9.

Spolaôr, N., Monard, M. C., and Lee, H. D. (2012). A systematic review to identify feature selection publications in multi-labeled data. Relatório Técnico do ICMC Nº 374. 31 pg. Universidade de São Paulo. Citado na página 3.

Spyromitros, E., Tsoumakas, G., and Vlahavas, I. (2008). An empirical study of lazy multilabel classification algorithms. In *Hellenic conference on Artificial Intelligence*, pages 401–406, Berlin, Heidelberg. Springer-Verlag. Citado nas páginas 26, 27, e 28.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3:1–13. Citado nas páginas 5 e 6.

Tsoumakas, G., Katakis, I., and Vlahavas, I. P. (2010). Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Citado nas páginas 1, 3, e 8.

Tsoumakas, G., Xioufis, E. S., Vilcek, J., and Vlahavas, I. P. (2011). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414. Citado na página 10.

Witten, I. H., Frank, E., Holmes, G., and Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., 3rd edition. Citado na página 10.

Zhang, M.-L., Peña, J. M., and Robles, V. (2009). Feature selection for multi-label naive bayes classification. *Information Sciences*, 179(19):3218–3229. Citado nas páginas vii, 2, 9, 10, 11, 12, 18, 19, e 32.