Aprendizado de máquina multirrótulo: explorando
a dependência de rótulos e o aprendizado ativo

Everton Alvares Cherman

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo

Everton Alvares Cherman

Orientadora: Profa. Dra. Maria Carolina Monard

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *EXEMPLAR DE DEFESA*.

USP – São Carlos
Novembro de 2013

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

C521a Cherman, Everton Alvares
Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo / Everton Alvares Cherman; orientadora Maria Carolina Monard. -- São Carlos, 2013.
129 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2013.

1. Aprendizado de máquina. 2. Aprendizado supervisionado. 3. Aprendizado multirrótulo. 4. Aprendizado ativo. 5. Inteligência artificial. I. Monard, Maria Carolina, orient. II. Título.

*Aos meus pais,
Iriuelto e Gertrudes.*

*A minha companheira,
Chris Tibes.*

*A minha orientadora,
Maria Carolina Monard.*

Agradecimentos

Diversas pessoas compartilharam a estrada da vida comigo. Algumas fizeram toda a diferença não só para realização deste trabalho, mas também para minha formação como pessoa. A essas pessoas devo meus sinceros agradecimentos.

Aos meus queridos pais, Irivelto e Gertrudes, por me agraciarem com todo o amor que um filho pode receber, pelos valores básicos de vida ensinados a mim com exemplos do dia a dia e por sempre me apoiarem nessa caminhada. Eu também agradeço minha irmã, Kamila, pelo exemplo de simplicidade de alma que cativa a todos em sua volta, principalmente a mim.

A minha companheira nessa jornada, Chris Tibes, por me apoiar, me incentivar e estar ao meu lado, não só fisicamente, desde antes do início deste projeto. Seu amor e seu apoio me incentivam a fazer sempre meu melhor e a crescermos juntos como pessoa por toda a estrada dessa vida. Estendo meus agradecimentos a sua família, Luiz, Terezinha e Louiza, por sempre me receberem como um filho em sua casa.

A minha orientadora, Maria Carolina Monard, a quem tenho profunda admiração e respeito. A senhora não só foi fundamental para o desenvolvimento deste trabalho, mas também para o meu desenvolvimento pessoal, principalmente quando os caminhos de minha vida pareciam como de um mar turbulento. A senhora é um exemplo de educadora, sabe considerar as individualidades de cada pessoa em sua volta como poucos nesse mundo. Estendo meus agradecimentos aos Professores Paulo e Huei, por terem sido meus primeiros mestres e terem me encaminhado à Professora Carolina.

A toda minha família, avós, avôs (*in memoriam*), tios e primos, por compartilharem diversos momentos importantes de minha vida desde o início da caminhada.

A todos os colegas e amigos do Labic, alguns em especial por terem contribuído para o desenvolvimento deste trabalho ou mesmo por me brindarem com suas amizades durante esses anos: Bruno Nogueira, Diego Silva, Ígor Braga, Jean Metz, Jorge Valverde, Marcos Cintra, Merley Conrado, Newton Spolaôr, Rafael Rossi, Ricardo Marcacini e Vinicius Souza. Agradeço também aos professores do Labic, Gustavo Batista e Solange Rezende.

Ao meu amigo e companheiro de república, Miguel Caro, pelo apoio típico de um irmão mais velho durante esses anos em São Carlos. Estendo meus agradecimentos a sua família, Maria, Mateo e Alejandro.

Ao Professor Grigorios Tsoumakas, pelo trabalho em colaboração realizado durante minha estadia em Thessaloniki. Estendo meus agradeci-

mentos a sua esposa, Theano, por ter me recebido na cidade, e a todos os amigos da Aristotle University of Thessaloniki: Anestis, Lefteris, Manolis, Margarita, Popi, Thanos e Yannis.

Ao Instituto de Ciências Matemáticas e de Computação (ICMC) pela infraestrutura e pelos Professores e Funcionários o compõem.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro fornecido a mim para o desenvolvimento do doutorado e do estágio no exterior. Considero a FAPESP um exemplo de instituição de fomento à pesquisa a ser seguida em todo o Brasil.

Resumo

Métodos tradicionais de aprendizado supervisionado, chamados de aprendizado monorrótulo, consideram que cada exemplo do conjunto de dados rotulados está associado a um único rótulo. No entanto, existe uma crescente quantidade de aplicações que lidam com exemplos que estão associados a múltiplos rótulos. Essas aplicações requerem métodos de aprendizado multirrótulo. Esse cenário de aprendizado introduz novos desafios que demandam abordagens diferentes daquelas tradicionalmente utilizadas no aprendizado monorrótulo. O custo associado ao processo de rotulação de exemplos, um problema presente em aprendizado monorrótulo, é ainda mais acentuado no contexto multirrótulo. O desenvolvimento de métodos para reduzir esse custo representa um desafio de pesquisa nessa área. Além disso, novos métodos de aprendizado também devem ser desenvolvidos para, entre outros objetivos, considerar a dependência de rótulos: uma nova característica presente no aprendizado multirrótulo. Há um consenso na comunidade de que métodos de aprendizado multirrótulo têm a capacidade de usufruir de melhor eficácia preditiva quando considerada a dependência de rótulos. Os principais objetivos deste trabalho estão relacionados a esses desafios: reduzir o custo do processo de rotulação de exemplos; e desenvolver métodos de aprendizado que explorem a dependência de rótulos. No primeiro caso, entre outras contribuições, um novo método de aprendizado ativo, chamado *score dev*, é proposto para reduzir os custos associados ao processo de rotulação multirrótulo. Resultados experimentais indicam que o método *score dev* é superior a outros métodos em vários domínios. No segundo caso, um método para identificar dependência de rótulos, chamado *UBC*, é proposto, bem como o *BR+*, um método para explorar essa característica. O método *BR+* apresenta resultados superiores a métodos considerados estado da arte.

Abstract

Traditional supervised learning methods, called single-label learning, consider that each example from a labeled dataset is associated with only one label. However, an increasing number of applications deals with examples that are associated with multiple labels. These applications require multi-label learning methods. This learning scenario introduces new challenges and demands approaches that are different from those traditionally used in single-label learning. The cost of labeling examples, a problem in single-label learning, is even higher in the multi-label context. Developing methods to reduce this cost represents a research challenge in this area. Moreover, new learning methods should also be developed to, among other things, consider the label dependency: a new characteristic present in multi-label learning problems. Furthermore, there is a consensus in the community that multi-label learning methods are able to improve their predictive performance when label dependency is considered. The main aims of this work are related to these challenges: reducing the cost of the labeling process; and developing multi-label learning methods to explore label dependency. In the first case, as well as other contributions, a new multi-label active learning method, called *score dev*, is proposed to reduce the multi-labeling processing costs. Experimental results show that *score dev* outperforms other methods in many domains. In the second case, a method to identify label dependency, called *UBC*, is proposed, as well as *BR+*, a method to explore this characteristic. Results show that the *BR+* method outperforms other state-of-the-art methods.

Esta tese foi preparada com o formator de texto $\text{\LaTeX}$. O modelo desenvolvido por Ronaldo Cristiano Prati foi utilizado. As citações bibliográficas seguem padrão *Apalike* do sistema Bib $\text{\TeX}$.

Sumário

Lista de Figuras	xv
Lista de Tabelas	xviii
Lista de Algoritmos	xix
Lista de Abreviaturas	xxi
Lista de Símbolos	xxiii
1 Introdução	1
1.1 Objetivos	3
1.2 Principais contribuições	6
1.3 Organização	8
2 Aprendizado Multirrótulo	9
2.1 Definições	9
2.2 Métodos de aprendizado multirrótulo	11
2.2.1 Métodos de transformação de problema	11
2.2.2 Métodos de adaptação de algoritmo	18
2.2.3 Classificação a partir de <i>ranking</i>	23
2.3 Avaliação do aprendizado	26
2.3.1 Medidas baseadas em exemplo	27
2.3.2 Medidas baseadas em rótulos	30
2.3.3 Classificadores <i>baseline</i>	32
2.4 Caracterização do conjunto de dados	37
2.5 Aplicações multirrótulo	38
2.6 Descrição dos conjuntos de dados utilizados neste trabalho . .	41
3 Dependência de Rótulos	43
3.1 Tipos de dependência de rótulos	44

3.2	<i>UBC: Identificando dependência de rótulos</i>	46
3.2.1	<i>Algoritmo UBC – Upper Bound Classifier</i>	47
3.2.2	<i>Avaliação experimental do UBC</i>	49
3.2.3	<i>Resultados e discussão</i>	50
3.3	<i>BR+: Explorando dependência de rótulos</i>	55
3.3.1	<i>Avaliação experimental do BR+</i>	59
3.3.2	<i>Resultados e discussão</i>	59
3.4	<i>Considerações finais</i>	67
4	Aprendizado Ativo Multirrótulo	69
4.1	<i>Conceitos básicos</i>	70
4.2	<i>Aprendizado ativo para dados multirrotulados</i>	71
4.2.1	<i>Abordagens manuais de rotulação</i>	72
4.2.2	<i>Requisições completas ou parciais de rótulos</i>	73
4.2.3	<i>Avaliação de exemplos não rotulados</i>	74
4.2.4	<i>Protocolo de experimentos</i>	83
4.3	<i>Avaliação experimental</i>	88
4.3.1	<i>Configuração dos experimentos</i>	88
4.3.2	<i>Resultados e discussão</i>	90
4.4	<i>Considerações finais</i>	98
5	Conclusões	101
5.1	<i>Classificadores <i>baseline</i> – Capítulo 2</i>	101
5.1.1	<i>Contribuições</i>	102
5.1.2	<i>Limitações e trabalhos futuros</i>	103
5.2	<i>Dependência de Rótulos – Capítulo 3</i>	103
5.2.1	<i>Contribuições</i>	104
5.2.2	<i>Limitações e trabalhos futuros</i>	105
5.3	<i>Aprendizado Ativo Multirrótulo – Capítulo 4</i>	106
5.3.1	<i>Contribuições</i>	106
5.3.2	<i>Limitações e trabalhos futuros</i>	107
A	Apêndice - Experimentos com Aprendizado Ativo Multirrótulo	109
A.1	<i>Protocolo de experimentos utilizando conjunto <i>independente</i> de teste</i>	109
A.2	<i>Protocolo de experimentos utilizando conjunto <i>remanescente</i> de teste</i>	116
	Referências Bibliográficas	121

Lista de Figuras

2.1	Ilustração do processo de treinamento e predição do método CC.	17
2.2	Hierarquia ilustrativa de rótulos e classificadores do método HOMER.	23
2.3	Principais medidas de avaliação do aprendizado multirrótulo.	27
2.4	Distribuição dos conjuntos de dados nos 64 trabalhos coletados na revisão sistemática.	36
2.5	Comparação de resultados reportados na literatura com os resultados obtidos pelo <i>baseline Geral_B</i>	36
2.6	Boxplot do caso ilustrativo.	38
2.7	Gráficos do tipo <i>boxplot</i> representando a distribuição de frequências dos rótulos de cada conjunto de dados.	42
3.1	Ilustração dos problemas binários utilizados no processo de treinamento do algoritmo <i>UBC</i>	48
3.2	Diferença entre <i>BR</i> e <i>UBC</i> para cada conjunto de dados e cada algoritmo base considerando <i>F-Measure</i> como medida de avaliação.	52
3.3	Gráficos radar apresentando valores de cinco medidas de avaliação para <i>BR</i> e <i>UBC</i> utilizando <i>SMO</i> como algoritmo base.	53
3.4	Gráficos radar apresentando valores de cinco medidas de avaliação para <i>BR</i> e <i>UBC</i> utilizando <i>J48</i> como algoritmo base.	54
3.5	Gráficos radar apresentando valores de cinco medidas de avaliação para <i>BR</i> e <i>UBC</i> utilizando <i>NB</i> como algoritmo base.	55
3.6	Ilustração do processo de treinamento e predição (sem atualização) do método <i>BR+</i>	56
3.7	Ilustração do processo de predição do <i>BR+</i> considerando uma ordem pré-definida.	58

3.8	Ranking de Friedman com pós-teste de Nemenyi para <i>SMO</i> como algoritmo base.	61
3.9	Ranking de Friedman com pós-teste de Nemenyi para <i>J48</i> como algoritmo base.	63
3.10	Ranking de Friedman com pós-teste de Nemenyi para <i>NB</i> como algoritmo base.	64
3.11	Ranking de Friedman com pós-teste de Nemenyi considerando os melhores resultados de cada método.	65
4.1	Ciclo padrão de aprendizado ativo <i>pool-based</i>	71
4.2	Custo geral de entender objetos de diferentes tipos de dados e custo de entender um rótulo.	74
4.3	Protocolo de experimentos para avaliação do classificador utilizando um conjunto <i>independente</i> e o conjunto de exemplos de consulta <i>remanescentes</i>	87
4.4	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>bibtex</i>	91
4.5	Ranking de Friedman com pós-teste de Nemenyi. Classificador <i>BR(SVM)</i> . Medida de avaliação <i>FV</i> . Protocolo de experimento <i>independente</i>	94
4.6	Ranking de Friedman com pós-teste de Nemenyi. Classificador <i>BR(SVM)</i> . Medida de avaliação <i>AULC</i> . Protocolo de experimento <i>independente</i>	96
4.7	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>corel5k</i>	96
4.8	Ranking de Friedman com pós-teste de Nemenyi. Classificador <i>RCut(LPBN)</i> . Medidas de avaliação <i>AULC</i> e <i>FV</i> . Protocolo de experimento <i>remanescente</i>	98
A.1	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>bibtex</i>	109
A.2	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>cal500</i>	110
A.3	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>corel16k001</i>	110
A.4	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>corel5k</i>	111
A.5	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>emotions</i>	111
A.6	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>enron</i>	112

A.7	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>llog-f</i>	112
A.8	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>medical</i>	113
A.9	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>ohsumed-f</i>	113
A.10	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>scene</i>	114
A.11	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>slashdot-f</i>	114
A.12	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>tmc2007-500</i>	115
A.13	Algoritmo de aprendizado <i>LIBLinear</i> . Protocolo de experimento <i>independente</i> . Conjunto de dados <i>yeast</i>	115
A.14	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>cal500</i>	116
A.15	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>corel16k001</i>	116
A.16	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>corel5k</i>	117
A.17	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>emotions</i>	117
A.18	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>enron</i>	117
A.19	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>llog-f</i>	118
A.20	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>medical</i>	118
A.21	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>ohsumed-f</i>	118
A.22	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>scene</i>	119
A.23	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>slashdot-f</i>	119
A.24	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>tmc2007-500</i>	119
A.25	Algoritmo de aprendizado <i>LPBHN-Rcut</i> . Protocolo de experimento <i>remanescente</i> . Conjunto de dados <i>yeast</i>	120

Lista de Tabelas

2.1	Exemplos rotulados no formato atributo-valor.	10
2.2	Exemplo ilustrativo de um conjunto de dados multirrótulo. .	12
2.3	Tabelas monorrótulo resultantes da aplicação do método <i>BR</i> no conjunto multirrótulo ilustrado na Tabela 2.2.	13
2.4	Tabela monorrótulo multiclasse resultante da aplicação do método <i>LP</i> no conjunto multirrótulo na Tabela 2.2.	13
2.5	Ilustração da construção de um meta conjunto de dados no algoritmo MetaC.	26
2.6	Exemplos das medidas de avaliação	29
2.7	Matrizes de confusão para um conjunto D com $L = \{a, b, c, d\}$. .	32
2.8	Valores das medidas <i>micro-avareged</i> e <i>macro-avareged</i> para as quatro medidas binárias apresentadas.	32
2.9	Ranking médio dos métodos <i>baseline</i>	35
2.10	Conjuntos de dados <i>benchmark</i> utilizados neste trabalho. . .	41
3.1	Resultados para <i>BR</i> e <i>UBC</i> utilizando doze conjuntos de dados. .	50
3.2	Influência da dependência de rótulos para cada conjunto de dados e algoritmo base.	51
3.3	Resultados utilizando <i>SMO</i> como algoritmo base.	60
3.4	Resultados utilizando <i>J48</i> como algoritmo base.	62
3.5	Resultados utilizando <i>NB</i> como algoritmo base.	64
3.6	Melhores resultados de cada método de aprendizado multirrótulo.	65
4.1	Exemplo ilustrativo do método de avaliação <i>conf</i>	77
4.2	Exemplo ilustrativo do método de avaliação <i>score</i>	78
4.3	Exemplo ilustrativo do método de avaliação <i>rank</i>	79
4.4	Exemplo ilustrativo do método de avaliação <i>conf</i> com aplicação das funções de agregação.	82

4.5	Exemplo ilustrativo do método de avaliação <i>score</i> com aplicação das funções de agregação.	82
4.6	Exemplo ilustrativo do método de avaliação <i>rank</i> com aplicação das funções de agregação.	82
4.7	Algoritmos de aprendizado ativo utilizados considerando diferentes funções <i>evaluate</i> e <i>aggregate</i>	89
4.8	Configurações de aprendizado ativo utilizadas nos experimentos.	90
4.9	Valores <i>FV</i> de eficácia na última iteração da curva de aprendizado para cada método de aprendizado ativo com o protocolo <i>independente</i>	92
4.10	Valores de <i>AULC</i> para cada método de aprendizado ativo com o protocolo <i>independente</i>	95
4.11	Valores <i>FV</i> de eficácia na última iteração da curva de aprendizado para cada método de aprendizado ativo com o protocolo <i>remanescente</i>	97
4.12	Valores de <i>AULC</i> para cada método de aprendizado ativo com o protocolo <i>remanescente</i>	97

Lista de Algoritmos

1	Esquema de votação do método <i>EPS</i>	16
2	Encontra o multirrótulo Z que maximiza <i>F-Measure</i>	34
3	Procedimento padrão para o aprendizado ativo	75
4	Avaliação de exemplos não rotulados pela função de avaliação (<i>query</i>).	76
5	Procedimento para composição do conjunto inicial de exemplos rotulados proposto e utilizado neste trabalho.	85

Lista de Abreviaturas

AULC	<i>Area Under the Learning Curve</i>
BR	<i>Binary Relevance</i>
CA	<i>Predição Com Atualização</i>
CC	<i>Chaining Classifiers</i>
ECC	<i>Ensemble of Chaining Classifiers</i>
EPS	<i>Ensemble of Pruned Sets</i>
FRI	<i>Fator de Rotulação Inicial</i>
FV	<i>Final Value</i>
kNN	<i>k-Nearest Neighbor</i>
LP	<i>Label Powerset</i>
LPBHN	<i>Label Propagation using Bipartite Heterogeneous Network</i>
MS	<i>Meta Stacking</i>
NB	<i>Naïve Bayes</i>
PCut	<i>Proportion-based cut</i>
PS	<i>Pruned Sets</i>
SA	<i>Predição Sem Atualização</i>
SVM	<i>Support Vector Machines</i>
RAKEL	<i>Random k-label sets</i>
RCut	<i>Rank-based cut</i>
SCut	<i>Score-based local optimization</i>
UBC	<i>Upper Bound Classifier</i>

Lista de Símbolos

Notação	Descrição
D	Conjunto de dados multirrótulo
D_u	Conjunto de dados não rotulado
D_l	Conjunto de dados rotulado
L	Conjunto de rótulos que participam dos multirrótulos
q	Número de possíveis rótulos que participam dos multirrótulos, <i>i.e.</i> , $q = L $
$CR(D)$	Cardinalidade de rótulos do conjunto de dados D
$DR(D)$	Densidade de rótulos do conjunto de dados D
y_j	Rótulo j do conjunto L , <i>i.e.</i> , $y_j \in L = \{y_1, y_2, \dots, y_q\}$
Y_i	Multirrótulo associado ao exemplo E_i
Z_i	Multirrótulo predito para o exemplo E_i
y'_j	Predição do monorrótulo $y_j \in L$
E_i	Exemplo no formato $(\mathbf{x}_i, Y_i)$
N	Número de exemplos no conjunto de dados
$\mathbf{x}_i$	Descritores do exemplo E_i da forma $(x_{i1}, x_{i2}, \dots, x_{iM})$
x_{ij}	Valor do atributo descritor j do exemplo E_i
M	Número de atributos descritores do conjunto de dados
$H(\mathbf{x})$	Classificador multirrótulo que retorna o multirrótulo de $\mathbf{x}$
H_0	Classificador multirrótulo de estágio inicial
H_1	Classificador multirrótulo de segundo estágio
H_{y_j}	Classificador monorrótulo binário para o rótulo y_j
f_b	Algoritmo de aprendizado monorrótulo binário
f_m	Algoritmo de aprendizado monorrótulo multiclasse
$r_i(y_j)$	Ranking do rótulo y_j para o exemplo E_i
n	Número de objetos a serem rotulados
c_o	Custo médio de entendimento de um objeto
c_l	Custo médio de entendimento de um rótulo
c_{lo}	Custo médio de atribuir um rótulo a um objeto
$query$	Função de seleção de um conjunto de objetos para rotulação
$oracle$	Função de rotulação de um conjunto de objetos
V_P	Número de exemplos verdadeiros positivos
V_N	Número de exemplos verdadeiros negativos
F_P	Número de exemplos falsos positivos
F_N	Número de exemplos falsos negativos

Introdução

ALGORITMOS de aprendizado de máquina supervisionado têm como objetivo identificar padrões de relações entre exemplos e suas classes. Um conjunto de exemplos para os quais a classe (ou rótulo) é conhecida é fornecido ao algoritmo de aprendizado supervisionado. O objetivo do algoritmo é construir um modelo que possa determinar automática e corretamente a classe de novos exemplos. Quando a classe dos exemplos assume valores discretos esse problema é chamado de classificação ([Mitchell, 1997](#); [Alpaydin, 2010](#); [Flach, 2012](#)).

Problemas de classificação podem ser encontrados em uma grande variedade de domínios, como categorização de textos de e-mails, notícias, páginas *web*; anotação de imagem, áudio e vídeo; dados biológicos e médicos; entre muitos outros. Cada exemplo representativo do domínio está associado a um conjunto de atributos descritores, bem como ao conceito que se deseja classificar automaticamente. Os rótulos dos exemplos são atribuídos tipicamente por anotadores humanos especialistas do domínio de aplicação. O tipo de especialização necessária para rotular os exemplos depende de cada domínio. O diagnóstico de um paciente, por exemplo, é atribuído por médicos dentro de sua especialidade. A classificação de músicas em gêneros musicais pode ser realizada por profissionais ou até por amantes da música. A categorização de e-mails ou imagens pessoais pode ser realizada pelo próprio usuário. O custo de se adquirir exemplos rotulados depende de cada domínio, mas invariavelmente se deseja minimizar o

custo associado ao processo de rotulação sem perder representatividade no conjunto de exemplos.

Atribuir um único rótulo a cada exemplo é considerada a abordagem padrão em aprendizado supervisionado, denominado aprendizado monorrótulo. Os métodos de aprendizado monorrótulo consideram que cada exemplo está associado a um único conceito meta (Mitchell, 1997). No entanto, é intuitivo para o ser humano associar suas observações a mais de um conceito. De fato, é crescente a quantidade de domínios de aplicação nos quais os exemplos estão associados a um conjunto de rótulos. Nesse contexto, a tarefa de classificação é chamada multirrótulo (Tsoumakas et al., 2009b; Carvalho e Freitas, 2009). Em classificação de notícias, por exemplo, 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 apresenta diabetes. Em bioinformática, a classificação multirrótulo é aplicada na predição das funções de genes, devido ao fato de que um gene pode influenciar em mais de uma função simultaneamente.

Esse novo contexto de aprendizado tem implicações nas soluções computacionais a serem utilizadas. Os rótulos, anteriormente representados apenas por uma dimensão, agora têm sua dimensão expandida, o que é refletido em todas as etapas de construção de um classificador, como a aquisição de exemplos rotulados, o desenvolvimento de novos algoritmos e a avaliação dos classificadores construídos. A seguir, são descritos alguns desafios de pesquisa relacionados ao aprendizado multirrótulo.

Aquisição de exemplos rotulados O processo de aquisição de rótulos para os exemplos é objeto de pesquisa recorrente em aprendizado monorrótulo. A obtenção de exemplos rotulados é tipicamente uma tarefa manual e custosa. Desse modo, a redução dos custos associados a esse processo é um importante desafio em aprendizado supervisionado. No contexto multirrótulo, o custo de rotulação é acentuado, pois mais rótulos são associados a cada exemplo. Desse modo, a necessidade de redução do esforço necessário para obtenção de um conjunto de dados multirrotulado e representativo é ainda maior, o que implica em novos desafios para o entendimento dos custos associados a esse contexto de aprendizado e o posterior desenvolvimento de abordagens para minimizar o esforço dos anotadores.

Desenvolvimento de algoritmos de aprendizado Após a obtenção dos dados rotulados, métodos de aprendizado podem ser aplicados para a geração de modelos preditivos. Em classificação multirrótulo, dois novos

desafios se destacam em relação ao aprendizado monorrótulo: i) escalabilidade; ii) dependência de rótulos. Como mais rótulos estão associados aos exemplos, existem mais relações entre os exemplos e os rótulos, as quais são necessárias para construir um modelo preditivo. A descoberta dessas relações implica em maior custo computacional, o que torna a escalabilidade dos métodos de aprendizado multirrótulo um aspecto importante para o desenvolvimento de novas abordagens.

Além das relações entre exemplos e rótulos, um novo tipo de relação surge em aprendizado multirrótulo: a dependência de rótulos. Esse tipo de relação pode ser entendido como a dependência que um rótulo exerce direta ou indiretamente sobre outro(s) rótulo(s). O consenso na comunidade é que considerar a dependência de rótulos é um fator chave para que os métodos de aprendizado multirrótulo usufruam da melhor eficácia preditiva possível no domínio de aplicação. A identificação dessa característica, bem como sua exploração pelo algoritmo de aprendizado, constitui um novo e importante desafio na predição do conjunto de rótulos que participam do multirrótulo de novos exemplos.

Avaliação dos classificadores construídos O processo de avaliação de classificadores no aprendizado multirrótulo apresenta desafios maiores que os presentes no aprendizado monorrótulo, pois a simples atribuição de predição correta ou incorreta não é suficiente para medir a eficácia preditiva de um classificador multirrótulo. Os exemplos multirrótulo podem ter sua predição *parcialmente* correta ou incorreta, o que torna necessária a adoção de novas medidas de avaliação. Além disso, em aprendizado monorrótulo, a eficácia de predição dos algoritmos propostos é frequentemente comparada à eficácia obtida por classificadores *baseline*, geralmente o classificador que sempre prediz a classe majoritária, ou seja, a classe mais frequente do conjunto de dados. No entanto, em aprendizado multirrótulo, identificar um classificador *baseline* se torna um desafio, pois existe a possibilidade de predições parcialmente (in)corretas.

1.1. Objetivos

Os desafios da área de aprendizado multirrótulo e seu potencial de aplicação têm despertado o interesse da comunidade pelo tema. Este trabalho está focado no entendimento dos novos problemas associados ao aprendizado multirrótulo, bem como no desenvolvimento de novas abordagens para esse tipo problema. Mais especificamente, neste trabalho, nos concentra-

mos em encontrar soluções para os seguintes desafios:

1. Contar com exemplos rotulados é um pré-requisito para qualquer abordagem de aprendizado supervisionado. No entanto, mesmo em aprendizado monorrótulo, o processo de rotulação é tipicamente manual e custoso. Em aprendizado multirrótulo esse custo é ainda mais acentuado pela necessidade de mapear múltiplos rótulos aos exemplos. Entender as implicações dessa nova dimensão aos custos relacionados à rotulação em problemas multirrotulados é essencial para minimizar o esforço dos anotadores e utilizar, por exemplo, métodos de aprendizado ativo. Nesse sentido, algumas perguntas norteiam este trabalho:

- *Como o anotador/rotulador humano deve realizar sua tarefa para minimizar o esforço necessário para concluí-la?*
- *Métodos de aprendizado ativo multirrótulo permitem reduzir a quantidade de exemplos a serem rotulados sem perder qualidade no modelo preditivo?*
- *Como avaliar os métodos de aprendizado ativo no cenário multirrótulo?*

2. Classificadores considerados *baseline* são importantes referências para identificar a eficácia de novas propostas. O erro majoritário, por exemplo, é uma medida em aprendizado monorrótulo fornecida por um classificador *baseline* que sempre prediz a classe majoritária do conjunto de dados, ou seja, que não considera os atributos descritores para fornecer sua predição. Novas propostas de algoritmos de aprendizado podem utilizar o erro majoritário como uma referência mínima de aprendizado a ser alcançada. Apesar de não ser obrigatório ser melhor que o *baseline*, casos em que um classificador não consegue superá-lo devem ser melhor explicados e, ao menos, apresentar resultados superiores em aspectos diferentes daqueles estritamente relacionados ao *baseline*. Nesse sentido, classificadores *baseline* para o aprendizado multirrótulo podem representar uma importante ferramenta de apoio à avaliação de novas propostas. Baseado nesse desafio, algumas perguntas norteiam este trabalho:

- *Qual abordagem baseline é adequada para o aprendizado multirrótulo?*
- *Como os resultados reportados na literatura se situam em relação aos resultados obtidos por um baseline?*

3. A dependência de rótulos é uma característica intrínseca ao problema de aprendizado multirrótulo e consiste, de maneira geral, na influência em que um rótulo exerce, direta ou indiretamente, sobre os outros rótulos. A opinião da comunidade de aprendizado multirrótulo é que resultados mais eficazes de predição podem ser obtidos por meio de métodos que consideram a dependência de rótulos no processo de aprendizado. De fato, existe um número crescente de trabalhos que evidenciam essa característica. No entanto, ainda não é claro qual é o limite da influência da dependência na eficácia de classificadores que a exploraram. Desse modo, as perguntas norteadoras relacionadas a esse tema são as seguintes:

- *Qual é a influência da dependência de rótulos no aprendizado multirrótulo?*
- *Quais são os limites da consideração da dependência de rótulos?*
- *Como explorar essa característica para melhorar a eficácia preditiva dos classificadores multirrótulo?*

Para responder a essas perguntas, algumas premissas amplamente adotadas, mesmo que implicitamente, em aprendizado multirrótulo foram consideradas para delimitação do escopo deste trabalho ([Read, 2010](#)):

1. *O conjunto de rótulos é conhecido de antemão, tem relevância para o domínio do problema e é interpretável por humano.*
2. *O número de possíveis rótulos é limitado, tipicamente não maior que o número de atributos descritores.*
3. *Cada exemplo está associado a um multirrótulo não vazio.*
4. *O número de atributos descritores não apresenta muita limitação. No entanto, não há a necessidade de considerar casos extremos de milhares de atributos, pois técnicas de seleção de atributos podem ser adotadas para a redução desse número.*
5. *Um algoritmo de aprendizado multirrótulo deve ser capaz de lidar com um número de exemplos de treinamento muito grande, potencialmente centenas de milhares.*

Sob essas premissas, um algoritmo de aprendizado deve gerar um classificador multirrótulo para um conjunto de dados de treinamento rotulado com o objetivo de prever o rótulo de novos exemplos.

1.2. Principais contribuições

Diversas contribuições referentes ao aprendizado multirrótulo foram originadas deste trabalho. As principais se referem aos desafios anteriormente expostos e são descritas sucintamente a seguir.

Baselines para o aprendizado multirrótulo Classificadores multirrótulo que desconsideram os atributos descritores foram elaborados para fornecer referências de *baseline* para a eficácia obtida por outros classificadores. O intuito da elaboração desse tipo de abordagem é identificar casos que necessitam de maior investigação por apresentarem resultados inferiores aos apresentados por *baselines*. Desse modo, *baselines* específicos para minimizar/maximizar determinadas medidas foram elaborados. A escolha do *baseline* a ser utilizado depende de qual medida o domínio da aplicação deseja minimizar/maximizar. Caso não se tenha definida uma medida em particular, um *baseline* geral chamado $Geral_B$, proposto também neste trabalho, pode ser utilizado (Metz et al., 2012). Além disso, por meio de uma revisão sistemática da literatura (Spolaôr et al., 2013a), os resultados do $Geral_B$ em dez conjuntos de dados frequentemente utilizados pela comunidade foram comparados aos resultados reportados na literatura por 64 trabalhos. De um total de 5342 valores de medidas coletadas, constatou-se que 12,8% não foram superiores aos valores das respectivas medidas obtidas pelo *baseline* $Geral_B$. Resultados inferiores ao $Geral_B$ não indicam necessariamente que o algoritmo de aprendizado avaliado não foi capaz de realizar sua tarefa adequadamente. Essa situação é um indicativo de alerta, ou seja, merece maior atenção e explicação dos pesquisadores quanto a esses resultados anômalos obtidos pelo algoritmo (Metz et al., 2013).

Aprendizado ativo multirrótulo Diferentes contribuições são apresentadas neste trabalho com intuito de reduzir o custo do processo manual de rotulação de exemplos multirrotulados. Basicamente, ao receber um lote de exemplos a serem rotulados, um rotulador humano pode realizar essa tarefa de duas maneiras distintas: foco por rótulo (*label-wise*) ou foco por exemplo (*object-wise*). Tal possibilidade é inexistente em aprendizado monorrótulo. A escolha da abordagem adequada depende do domínio de aplicação, mas para a maioria dos problemas a abordagem *object-wise* é mais adequada. Além disso, diferentes abordagens de aprendizado ativo multirrótulo são avaliadas com o intuito de selecionar os exemplos mais informativos a serem rotulados. Den-

tre essas abordagens, sugere-se a utilização do método *score dev*, proposto neste trabalho, por ter obtido resultados superiores à abordagem aleatória de maneira mais consistente. Desse modo, essa abordagem foi capaz de reduzir a quantidade de exemplos rotulados necessários para se obter bons classificadores. Um algoritmo recentemente proposto para inferência transdutiva baseado em grafo também foi utilizado com bons resultados neste trabalho. Algoritmos transdutivos são apropriados em casos que o conjunto de exemplos a ser rotulado é conhecido previamente. Não é de nosso conhecimento que essa abordagem tenha sido anteriormente analisada em aprendizado ativo multirrótulo. Este trabalho também contribuiu com aspectos de avaliação experimental dos métodos de aprendizado ativo, como a proposta de um critério chamado *Fator de Rotulação Inicial* para o processo de composição de um conjunto de exemplos rotulados anterior ao início do aprendizado ativo propriamente dito (Cherman et al., 2013b).

Entendimento e exploração da dependência de rótulos A dependência de rótulos é uma característica fundamental em aprendizado multirrótulo. Neste trabalho foram propostas duas abordagens para o avanço da área relacionado a esse tema. A primeira, denominada *UBC*, tem como objetivo medir o grau de dependência de rótulos em um conjunto de dados. Com essa abordagem foi possível identificar conjuntos de dados *benchmarks* que são positivamente influenciados pela dependência de rótulos, bem como outros conjuntos que apresentam pouca influência dessa característica. A presença de dependência de rótulos em um conjunto de dados poderia ser aferida com medidas clássicas da literatura da teoria de informação. No entanto, diferentemente do *UBC*, elas não fornecem uma estimativa do efeito da dependência de rótulos em termos de medidas de avaliação da eficácia preditiva do classificador multirrótulo. A segunda abordagem proposta se refere ao algoritmo *BR+*, o qual foi desenvolvido para explorar a dependência de rótulos presente nos conjuntos de dados e, assim, melhorar a eficácia dos classificadores gerados. O *BR+* apresentou resultados superiores ao método *BR*, que considera rótulos independentes, bem como em relação a outros classificadores considerados estado da arte na área que também tentam explorar a dependência de rótulos (Cherman et al., 2010a,b,c, 2011, 2012).

Além das contribuições relacionadas ao foco principal desta tese, o trabalho (Cherman et al., 2013a), relacionado ao aprendizado multirrótulo baseado em grafos, foi desenvolvido e é apresentado na Seção 2.2.2. Outras

contribuições relacionadas ao aprendizado multirrótulo também foram desenvolvidas nos seguintes trabalhos colaborativos: métodos de aprendizado multirrótulo baseados no paradigma *lazy* (Reis et al., 2012); abordagens para seleção de atributos multirrótulo (Spolaôr et al., 2012, 2013b,c); e *ML-datagen – framework* para geração de conjuntos de dados artificiais para pesquisa em aprendizado multirrótulo¹ (Tomás et al., 2013b,a).

Os algoritmos propostos e desenvolvidos no decorrer deste trabalho estão publicamente disponíveis².

1.3. Organização

O restante do trabalho está organizado da seguinte maneira:

Capítulo 2: Aprendizado Multirrótulo Uma revisão sobre os principais conceitos relacionados ao aprendizado multirrótulo é apresentada nesse capítulo. Além disso, propostas realizadas no desenvolvimento deste trabalho referente aos classificadores *baselines*, bem como estratégias de aprendizado multirrótulo baseadas em grafo, são descritas em conjunto com trabalhos relacionados da literatura.

Capítulo 3: Dependência de Rótulos Conceitos referentes à dependência de rótulos são inicialmente descritos nesse capítulo. Posteriormente, as abordagens *UBC* e *BR+*, propostas neste trabalho, são apresentadas em conjunto com a correspondente avaliação experimental.

Capítulo 4: Aprendizado Ativo Multirrótulo Nesse capítulo, inicialmente são abordados conceitos gerais sobre aprendizado ativo. Posteriormente, conceitos desse tema com foco no contexto multirrótulo são apresentados. Contribuições relacionadas ao entendimento dos custos envolvidos no processo de rotulação multirrótulo, bem como quanto ao protocolo experimental e novas abordagens para avaliação e realização dessa tarefa, são apresentadas nesse capítulo.

Capítulo 5: Conclusões Por fim, as considerações finais desta tese são apresentadas nesse capítulo. As principais contribuições provenientes do desenvolvimento deste trabalho são retomadas. Por fim, são descritas as limitações e os trabalhos futuros.

¹Disponível publicamente em <http://sites.labic.icmc.usp.br/mldatagen/>

²<http://www.labic.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/>

Aprendizado Multirrótulo

DEVIDO à crescente gama de aplicações compostas por exemplos multirrótulo, a quantidade de trabalhos relacionados ao aprendizado multirrótulo, foco de estudo deste trabalho, tem incrementado nos últimos anos. Com intuito de apresentar uma visão geral sobre essa área, neste capítulo são abordados os principais conceitos e definições relativos ao aprendizado multirrótulo, bem como os principais métodos propostos para esse tipo de aprendizado, além de medidas de avaliação e algumas aplicações. Algumas propostas deste trabalho também são descritas neste capítulo.

2.1. Definições

Para a tarefa de aprendizado supervisionado monorrótulo, a entrada do algoritmo consiste de um conjunto de N exemplos E_i , com $i = 1, \dots, N$, escolhidos de um domínio com uma distribuição $\mathcal{D}$ fixa, desconhecida e arbitrária, da forma $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ para alguma função desconhecida $y = f(\mathbf{x})$. Os $\mathbf{x}_i$ são tipicamente vetores da forma $(x_{i1}, x_{i2}, \dots, x_{iM})$ 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 2.1(a). Os valores (rótulos) y_i referem-se ao valor do atributo Y , frequentemente denominado classe.

Dado um conjunto D de exemplos a um algoritmo de aprendizado, um

classificador H , que aproxima f , i.e., $H(\mathbf{x}) \approx f(\mathbf{x})$, será induzido. O classificador consiste da *hipótese* feita sobre a verdadeira (mas desconhecida) função f . Dado um novo exemplo $\mathbf{x}$, o classificador, ou hipótese H , prediz o valor correspondente de y .

Assim, na classificação monorrótulo cada exemplo E_i está associado a um único rótulo y_i presente no conjunto de rótulos L , i.e., $y_i \in L$, com $|L| > 1$. Quando o conjunto de exemplos contém $|L| = 2$ rótulos o problema é chamado de classificação *binária*, enquanto se $|L| > 2$ rótulos, o problema é denominado classificação *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 $L = \{y_j : j = 1..q\}$ é o conjunto dos q rótulos simples que participam dos multirrótulos Y_i , $i = 1..N$, como representado na Tabela 2.1(b)¹.

Tabela 2.1: Exemplos rotulados no formato atributo-valor.

(a) Monorrótulo						(b) Multirrótulo					
	X_1	X_2	$\dots$	X_M	Y		X_1	X_2	$\dots$	X_M	Y
E_1	x_{11}	x_{12}	$\dots$	x_{1M}	y_1	E_1	x_{11}	x_{12}	$\dots$	x_{1M}	Y_1
E_2	x_{21}	x_{22}	$\dots$	x_{2M}	y_2	E_2	x_{21}	x_{22}	$\dots$	x_{2M}	Y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{N2}	$\dots$	x_{NM}	y_N	E_N	x_{N1}	x_{N2}	$\dots$	x_{NM}	Y_N

Dado um conjunto de exemplos, é importante distinguir 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 foram “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

¹Existe 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.

do modelo, mas são utilizados para o seu ajuste. Dessa maneira, esses exemplos são indiretamente “vistos” durante o processo de aprendizado, o que obriga que os exemplos de validação sejam distintos dos exemplos de teste.

2.2. Métodos de aprendizado multirrótulo

Os primeiros métodos de aprendizado multirrótulo foram propostos para a categorização de dados textuais em torno do ano de 1999. A classificação multirrótulo tem sido aplicada e desenvolvida para uma grande gama de aplicações, como anotação de imagens e vídeo (Zhang e Zhou, 2007a; Qi et al., 2007; Boutell et al., 2004), bioinformática (Vens et al., 2008; Blockeel et al., 2006; Clare e King, 2001), busca e mineração na *web* (Ueda e Saito, 2003), categorização de músicas em emoções (Trohidis et al., 2008; Li et al., 2006; Li e Ogihara, 2003) e marketing direcionado (Zhang, 2006). Os métodos de aprendizado multirrótulo podem ser categorizados de diversas maneiras. Neste trabalho será adotada a categorização amplamente utilizada pela comunidade que propõe três distintos grupos de algoritmos de classificação multirrótulo (Tsoumakas et al., 2009b):

1. Transformação de problema;
2. Adaptação de algoritmo;
3. Classificação a partir de *ranking*.

A abordagem de transformação do problema, como o próprio nome diz, transforma o problema multirrótulo em um ou mais problemas monorrótulo. Após essa transformação, algoritmos de classificação monorrótulo tradicionais são aplicados para resolução de cada problema monorrótulo separadamente. Por outro lado, a abordagem de adaptação de algoritmo considera o problema multirrótulo diretamente, sem realizar a transformação para problemas monorrótulo. Por fim, a última abordagem é relacionada à *ranking*, na qual a classificação multirrótulo é dada por meio da ordenação dos rótulos para um dado exemplo a ser predito e posterior realização de corte nesse *ranking* para determinar os rótulos considerados relevantes. A seguir são descritos alguns dos principais métodos de cada abordagem.

2.2.1 Métodos de transformação de problema

Vários métodos de transformação de problema foram propostos. A seguir, são descritos os métodos comumente utilizados que seguem esse pa-

radigma:

- *Binary Relevance* – *BR* (Tsoumakas et al., 2009b);
- *Label Powerset* – *LP* (Tsoumakas et al., 2009b);
- *RANdom k -label sets* – *RAKEL* (Tsoumakas e Vlahavas, 2007);
- *Pruned Sets* – *PS* (Read et al., 2008);
- *Classifiers Chain* – *CC* (Read et al., 2009);
- *Meta Stacking* – *MS* (Godbole e Sarawagi, 2004);
- *2BR* (Tsoumakas et al., 2009a).

Para a exemplificação de alguns dos métodos, é utilizado o conjunto de exemplos multirrótulo ilustrado na Tabela 2.2. Esse conjunto contém quatro exemplos rotulados com um ou mais rótulos do conjunto de rótulos simples $L = \{y_1, y_2, y_3, y_4\}$. Como a transformação do problema somente altera o espaço dos rótulos, para simplificar, é omitido o espaço de atributos descritores nas próximas tabelas.

Tabela 2.2: Exemplo ilustrativo de um conjunto de dados multirrótulo.

	Y
E_1	$Y_1 = \{y_2, y_3\}$
E_2	$Y_2 = \{y_1, y_3, y_4\}$
E_3	$Y_3 = \{y_4\}$
E_4	$Y_4 = \{y_2, y_3\}$

Método *Binary Relevance* O *BR* é um método simples e muito utilizado, o qual consiste em dividir o problema multirrótulo em $|L|$ problemas de classificação binária, um para cada rótulo contido em L . Na Tabela 2.3 são apresentados os quatro conjuntos monorrótulo resultantes da aplicação do método *BR* no conjunto multirrótulo ilustrado na Tabela 2.2. Esses conjuntos são utilizados para treinar $|L|$ classificadores binários, cada um considerando um dos conjuntos de dados resultantes da transformação ilustrada na Tabela 2.3. Dado um novo exemplo, a sua predição multirrótulo é formada pela união dos rótulos associados aos classificadores binários que classificam esse exemplo como positivo.

O *BR* apresenta complexidade computacional de $O(q \times f_b(M, N))$, onde $f_b(\bullet, \bullet)$ é a complexidade do algoritmo base de aprendizado binário em

Tabela 2.3: Tabelas monorrótulo resultantes da aplicação do método *BR* no conjunto multirrótulo ilustrado na Tabela 2.2.

	Y		Y		Y		Y
E_1	$\neg y_1$	E_1	y_2	E_1	y_3	E_1	$\neg y_4$
E_2	y_1	E_2	$\neg y_2$	E_2	y_3	E_2	y_4
E_3	$\neg y_1$	E_3	$\neg y_2$	E_3	$\neg y_3$	E_3	y_4
E_4	$\neg y_1$	E_4	y_2	E_4	y_3	E_4	$\neg y_4$

um conjunto de treinamento com N exemplos descritos por M atributos. Isso significa que *BR* é majoritariamente influenciado pela quantidade de rótulos e pela complexidade do algoritmo base binário utilizado. Além disso, o *BR* pode ser facilmente paralelizado, pois cada problema binário é independente, o que torna esse método altamente escalável. Apesar de sua simplicidade, o *BR* apresenta resultados adequados para diversos casos. No entanto, sua desvantagem é o fato de não considerar a dependência de rótulos na construção do modelo de classificação multirrótulo, já que cada classificador binário é construído independentemente dos outros.

Método *Label Powerset* O método *LP* consiste em transformar o problema multirrótulo em um problema monorrótulo multiclasse, no qual cada multirrótulo presente no conjunto de treinamento é mapeado em um valor do atributo classe. Na Tabela 2.4 é apresentado o conjunto monorrótulo multiclasse resultante da transformação dos exemplos descritos na Tabela 2.2 utilizando o método *LP*. Nessa tabela, a notação $y_{i,j,k}$ indica que o exemplo correspondente está rotulado com $y_i \wedge y_j \wedge y_k$.

Tabela 2.4: Tabela monorrótulo multiclasse resultante da aplicação do método *LP* no conjunto multirrótulo na Tabela 2.2.

	Y
E_1	$y_{2,3}$
E_2	$y_{1,3,4}$
E_3	y_4
E_4	$y_{2,3}$

Um novo exemplo é classificado de acordo com o valor de predição dado por um algoritmo multiclasse treinado com um conjunto de exemplos resultante da transformação ilustrada na Tabela 2.4.

O *LP* apresenta complexidade computacional de $O(f_m(M, N, 2^q))$, onde $f_m(\bullet, \bullet, \bullet)$ é a complexidade do algoritmo base de aprendizado multiclasse para um conjunto de treinamento com N exemplos, M atributos descritores e até 2^q possíveis combinações de rótulos. Tratar cada

combinação de multirrótulo como uma classe distinta traz implicações em termos de eficácia preditiva e complexidade computacional para esse método. Os conjuntos de dados com grande número de distintas combinações de multirrótulo resultam em um problema monorrótulo multiclasse com grande número de classes e altamente desbalanceado, pois a maioria dos rótulos está associada a poucos exemplos. Desse modo, além de afetar a eficácia preditiva do modelo criado, alguns algoritmos base são quadráticos (ou mais custosos) em relação a quantidade de classes, como é o caso do *support vector machine*, o que tornaria o treinamento do modelo custoso e até proibitivo computacionalmente para grandes conjuntos de dados. No entanto, o *LP* também é um método de transformação simples com a vantagem de considerar a dependência de rótulos. Para conjuntos de dados com pequeno número de combinações de rótulos, essa característica do método pode incrementar a eficácia do modelo preditivo gerado.

Random *k*-labelsets O método *RAKEL*, proposto por Tsoumakas e Vlahavas (2007), foi criado com o objetivo de evitar alguns dos problemas do *LP* mencionados anteriormente e ainda considerar a dependência de rótulos. Para tanto, *RAKEL* utiliza o conceito de *k-labelset*, o qual é definido como o subconjunto $L'_i \subseteq L$ com $k = |L'_i|$. O conjunto de todos os distintos *k-labelsets* em L é representado pelo termo L^k .

O algoritmo *RAKEL* iterativamente constrói um *ensemble* de m classificadores *LP*. A cada iteração, $i = 1..m$, um *k-labelset* $L'_i \subseteq L^k$ é selecionado aleatoriamente sem reposição. Utilizando o *k-labelset* L'_i , é treinado um classificador *LP*. O número de iterações m e o tamanho do conjunto de rótulos k são definidos pelo usuário. Quando $k = 1$ e $m = |L|$ o *RAKEL* obtém o mesmo *ensemble* de classificadores do *Binary Relevance*, enquanto que para $k = |L|$ e, conseqüentemente, $m = 1$ é obtido o classificador multi-classe do método *LP*. Dessa maneira, o método *RAKEL* pode reduzir a quantidade de combinações de rótulos possíveis para evitar o desbalanceamento excessivo de classes.

Para exemplificar, considere o problema multirrótulo ilustrado na Tabela 2.2 com $L = \{y_1, y_2, y_3, y_4\}$. Considere também um *ensemble* de $m = 2$ classificadores *LP* e *k-labelsets* de tamanho $k = 3$. Desse modo, os sub-problemas multirrótulos a serem resolvidos poderiam conter os seguintes *k-labelsets* $L'_1 = \{y_1, y_2, y_4\}$ e $L'_2 = \{y_2, y_3, y_4\}$ selecionados aleatoriamente. Nesse caso, dois classificadores *LP* seriam construídos, o primeiro para L'_1 , o qual desconsideraria o rótulo y_3 do conjunto de treinamento, e o segundo para L'_2 , o qual consideraria todos os rótulos exceto y_1 .

Para a classificação de um novo exemplo $\mathbf{x}$, cada modelo LP construído fornece decisões binárias relativas a cada rótulo y_i correspondente ao seu k -labelset. A partir disso, o *RAKEL* calcula a decisão média para cada rótulo e retorna resposta positiva se a média for maior que um valor de corte (*threshold*) t definido pelo usuário. Um valor intuitivo para t é 0,5, mas segundo os autores, o *RAKEL* obtém bons resultados para uma grande faixa de valores de t .

Apesar do método *RAKEL* reduzir as possíveis combinações de rótulos, ele ainda apresenta complexidade computacional semelhante à complexidade do método LP : $O(m \times f_m(M, N, 2^k))$. Quando $k = |L|$ e $m = 1$, a complexidade do método é igual à apresentada pelo LP . Desse modo, o parâmetro k tem grande influência na complexidade computacional do método.

Pruned Sets A motivação por trás do método *PS* está em identificar as principais relações de rótulos que participam no conjunto de dados multirrótulo (Read et al., 2008). Para tanto, o método realiza a poda de combinações de rótulos não frequentes, com o objetivo de diminuir a complexidade do problema. Numa segunda etapa, as combinações de rótulos podadas são divididas em subconjuntos menores e mais frequentes, o que torna possível retornar alguns exemplos ao conjunto de dados, garantindo perda mínima de informação. A operação de poda é controlada por um parâmetro p que determina o quão frequente uma combinação de rótulos deve ser para que seja mantida no conjunto de dados. Desse modo, o método preserva as principais relações de rótulos e o método LP pode ser aplicado para considerá-las. No entanto, um problema presente nesse método, assim como no LP , é a incapacidade de prever combinações de rótulos não vistas no conjunto de treinamento.

Nesse mesmo trabalho, também foi proposto o *Ensemble of Pruned Sets* – *EPS*. Os autores consideram que esse método pode auxiliar a evitar o efeito do *overfitting* do processo de poda e permitir a criação de novas combinações de rótulos no momento de classificar novos exemplos. A construção dos m classificadores que constituem o *ensemble* é realizada considerando subconjuntos dos exemplos de treinamento (os autores utilizaram 63%) para cada classificador. A predição de um novo exemplo segue o esquema de votação ilustrado no Algoritmo 1.

O estudo experimental nesse trabalho apontou que o método *EPS*, em geral, apresenta resultados melhores que os métodos *BR*, LP e *RAKEL*. Quanto à complexidade computacional, no pior caso (quando cada

```

Entrada: exemplo de teste  $E_i$ ;
ensemble de classificadores  $H_0, \dots, H_m$ ;
parâmetro de corte  $t$ .
Resultado: vetor  $z$  que representa o conjunto de rótulos  $Z_i$  preditos
para o exemplo  $E_i$ 
 $v \leftarrow (0, 0, 0, \dots, 0)$  // vetor de tamanho  $|L|$  que registra a soma de votos
de cada rótulo
for  $j \leftarrow 0$  to  $m$  do
     $w \leftarrow H_j(E_i)$  // classificador  $H_j$  retorna a predição representada por
    um vetor
     $v \leftarrow w + v$  // atualiza a soma dos votos com o vetor predito
end
for  $j \leftarrow 0$  to  $|L|$  do
     $z[j] \leftarrow (v[j] > t)?1 : 0$ 
end
Retorne  $z$ 

```

Algoritmo 1: Esquema de votação do método *EPS*.

exemplo apresenta uma distinta combinação de rótulos) o *PS* apresenta uma complexidade equivalente ao *LP*. No entanto, em aplicações reais é esperado que essa complexidade seja bem melhor, devido à poda realizada pelo método.

Chaining Classifiers Esse método tem como objetivo principal adicionar a característica de dependência de rótulos no método *BR*. Para tanto, o método *CC* (Read et al., 2009) utiliza $|L|$ classificadores binários assim como o método *BR*. Os classificadores binários são ligados em uma corrente (*chain*), utilizando uma ordem definida aleatoriamente, na qual cada classificador lida com um problema binário associado a um rótulo $y_j \in L$. O espaço de atributos descritores de cada classificador contido na corrente é estendido com os atributos dos rótulos anteriores na corrente.

A predição de um novo exemplo é realizada ativando cada classificador binário da corrente de classificadores seguindo a ordem definida no treinamento, de modo que os resultados dos classificadores anteriores são utilizados pelos próximos classificadores. A vantagem desse método é o fato de manter a simplicidade do método *BR* e considerar a dependência de rótulo no processo. O processo de treinamento e predição é ilustrado na Figura 2.1 considerando $L = \{y_1, y_2, y_3\}$ e $y_1 \prec y_2 \prec y_3$ como a ordem da corrente.

Nesse mesmo trabalho, o autor também propôs a utilização de *ensembles* com o método *CC*, denominado *Ensembles of Chaining Classifiers* – *ECC*. Para tanto, o autor realiza o treinamento de m classificadores *CC*, cada um treinado com um subconjunto de exemplos de

Figura 2.1: Ilustração do processo de treinamento e predição do método CC.

treinamento e com uma ordem aleatória distinta de encadeamento dos classificadores binários. A predição de um novo exemplo é realizada utilizando um esquema de votação análoga ao método *EPS* – Algoritmo 1.

Nos resultados experimentais apresentados em (Read et al., 2009), tanto o método *CC* quanto o método *ECC* apresentaram resultados superiores a métodos como *BR*, *LP*, *RAKEL* e *EPS*. Além disso, o método *CC* apresenta uma complexidade computacional semelhante ao *BR*. A diferença está na adição dos atributos-rótulos aos atributos-descritores. Desse modo, o *CC* apresenta a complexidade $O(q \times f_b(M + q, N))$. Na prática, o número de rótulos q do problema é muito menor que o número de atributos descritores. Além disso, assim como o *BR*, o método *CC* pode ser facilmente paralelizado para o treinamento dos modelos preditivos.

Meta Stacking Esse foi um dos primeiros métodos a explorar a dependência de rótulos no processo de aprendizado multirrótulo (Godbole e Saurawagi, 2004). Os autores assumem que a classificação do rótulo y_j pode ser um bom indicador para a classificação do rótulo y_k , com $k \neq j$. Desse modo, uma maneira de melhorar o processo de aprendizado é incrementar em $|L|$ atributos o conjunto de atributos descritores dos problemas binários² $H_{y_j,1}$ de todos os rótulos $y_j \in L$. Os valores desses novos atributos estão relacionados ao resultado de classificação do exemplo correspondente. No entanto, essa abordagem causa uma dependência cíclica, a qual pode ser quebrada por uma etapa anterior que utilize um classificador $H_{y_j,0}$ sem a necessidade de aumentar o espaço de atributos descritores.

² $H_{y_j,k}$ representa o classificador binário que responde pelo rótulo y_j na etapa k do processo de aprendizado.

A predição de um novo exemplo é realizada primeiramente utilizando o classificador $H_{y_j,0}$ para todo rótulo $y_j \in L$, o que produz os valores necessários para, finalmente, o classificador $H_{y_j,1}$ ser aplicado e o resultado final ser obtido. Os autores utilizam os *scores* fornecidos pelo método SVM para preencher os novos atributos criados relacionados aos rótulos, tanto no treinamento quanto no momento da predição de um novo exemplo.

Como mencionado, esse método tem a vantagem de considerar indiretamente a dependência de rótulos no processo de aprendizado, o que pode melhorar a eficácia do processo. No entanto, ele apresenta a desvantagem de um possível *overfitting* do classificador, já que os *scores* pelos quais é realizado o treinamento na segunda etapa são obtidos da predição do próprio conjunto de treinamento. Esse método foi avaliado comparativamente com o método CC em (Read et al., 2009) e apresentou resultados menos eficazes que o método CC. Quanto à complexidade computacional, pode-se considerar que o método apresenta duas vezes a complexidade do *BR*, dado que é necessário o treinamento de dois modelos preditivos.

2BR O método *Meta Stacking* foi generalizado em (Tsoumakas et al., 2009a) a fim de considerar qualquer classificador monorrótulo que forneça *scores* relacionados às predições. A generalização do *MS* foi chamada de *2BR*. Além disso, os autores também propuseram diferentes abordagens para evitar o *overfitting* presente no *MS*. Uma das abordagens proposta consiste em dividir o conjunto de treinamento em F partições (*folds*) disjuntas de tamanho aproximadamente iguais. Esse processo é realizado separadamente para cada rótulo e o meta-conjunto de dados é construído concatenando os valores de *scores* obtido pela predição dos F *folds* utilizando o classificador³ H_0 . Dessa forma, é possível evitar o *overfitting* para a construção do classificador H_1 . No entanto, com algumas exceções, os resultados apresentados pelo *2BR* seguiram na mesma linha do *MS*.

2.2.2 Métodos de adaptação de algoritmo

Os métodos dessa abordagem, diferentemente dos métodos de transformação do problema, são específicos para tratar diretamente o problema de classificação multirrótulo. Diversos métodos foram propostos e a seguir são descritos os métodos de maior destaque na comunidade.

- AdaBoost Multirrótulo (Schapire e Singer, 2000);

³ H_k é o classificador multirrótulo da etapa k do processo de aprendizado.

- C4.5 Multirrótulo – C4.5M (Clare e King, 2001);
- k NN Multirrótulo (Zhang e Zhou, 2005; Spyromitros et al., 2008);
- Estratégias baseadas em grafo (Cherman et al., 2013a);
- *Back-Propagation Multi-Label Learning* – BP-MLL (Zhang, 2006);
- *Multi-class Multi-label Perceptron* – MMP (Crammer et al., 2003);
- *Multi-class Multi-label Associative Classification* – MMAC (Thabtah et al., 2004);
- *Hierarchy Of Multi-label classifERs* – HOMER (Tsoumakas et al., 2008).

AdaBoost Multirrótulo Duas extensões do algoritmo AdaBoost (Freund e Schapire, 1995), denominadas respectivamente AdaBoost.MH e AdaBoost.MR, foram propostas para tratar o problema multirrótulo (Schapire e Singer, 2000). AdaBoost.MH foi desenvolvido para minimizar a medida *Hamming Loss*, enquanto que AdaBoost.MR foi projetado para escolher a hipótese que dispõe os rótulos corretos no topo do ranking.

C4.5M No trabalho de Clare e King (2001) é proposta uma modificação do algoritmo C4.5 (Quinlan, 1993) para a classificação de genes de acordo com suas funções, o qual é um típico problema multirrótulo. O algoritmo C4.5 utiliza a medida de entropia para escolher os nós de decisão das árvores. No entanto, essa medida foi originalmente concebida para tratar o problema monorrótulo. Assim, os autores adaptaram a fórmula do cálculo da entropia – Equação 2.1 – para permitir a sua utilização em problemas multirrótulo.

$$entropia = - \sum_{j=1}^{|L|} (p(y_j) \log p(y_j) + q(y_j) \log q(y_j)) \quad (2.1)$$

onde $p(y_j)$ é a frequência relativa do rótulo simples y_j e $q(y_j) = 1 - p(y_j)$.

Outra modificação proposta foi a utilização de nós folhas que representam não mais apenas um único rótulo, mas um conjunto de rótulos.

k NN Multirrótulo Vários métodos propostos são baseados no algoritmo monorrótulo k NN (Trohidis et al., 2008; Spyromitros et al., 2008; Brinker e Hüllermeier, 2007; Luo e Heywood, 2005; Zhang e Zhou, 2005). A primeira etapa de todos esses métodos é a mesma que a do k NN, *i.e.*, recuperar os k exemplos mais próximos. A diferença entre eles está na maneira como é calculado o conjunto de rótulos a ser predito.

O método *BRkNN* (Spyromitros et al., 2008) tem o mesmo comportamento que utilizar o *Binary Relevance* com o *kNN* como algoritmo-base. A diferença está na sua implementação, a qual não necessita realizar o cálculo dos vizinhos mais próximos para cada problema binário separadamente, mas sim uma única vez, o que melhora a eficiência computacional do método. Os autores também propuseram duas extensões: *BRkNN-a* e *BRkNN-b*. A primeira extensão tem como objetivo tratar a possibilidade do *BRkNN* disponibilizar como resposta um conjunto vazio de rótulos, ou seja, nenhum rótulo predito pra um determinado exemplo. Caso esse fato ocorra, o *BRkNN-a* fornece como resposta o rótulo com maior confiança de predição. Na segunda extensão, é calculado o tamanho médio m do conjunto de rótulos dos k exemplos recuperados e fornece como resposta os $\lceil m \rceil$ (inteiro mais próximo de m) rótulos com maior confiança de predição.

Outro método baseado em *kNN* é *MLkNN* (Zhang e Zhou, 2005), o qual utiliza o princípio *maximum a posteriori* para determinar o conjunto de rótulos do exemplo a ser classificado, baseado nas probabilidades a priori e posteriori da frequência de cada rótulo dos k vizinhos mais próximos.

Estratégias baseadas em grafo Recentemente, em (Cherman et al., 2013a), propomos a utilização de técnicas baseadas em grafo para a definição dos vizinhos mais próximos a serem considerados para o cálculo dos rótulos finais. Três abordagens foram avaliadas e comparadas com a estratégia tradicional do *BRkNN* de definição dos vizinhos: *ML ϵ* , *ML-MUT* e *MLnotMUT*.

Para determinar quais são os vizinhos mais relevantes para calcular o multirrótulo de novos exemplos, essas abordagens consideram a teoria de grafos, na qual os exemplos são os vértices e as arestas são determinadas conforme algum critério baseado na distância entre os exemplos.

Duas das principais heurísticas para a geração desses grafos são: abordagem baseada na similaridade mínima e abordagem baseada nos vizinhos mais próximos (Brito et al., 1997). A primeira abordagem, também conhecida como ϵ -vizinhança, consiste em criar uma aresta entre os exemplos E_i e E_j se $\| \mathbf{x}_i - \mathbf{x}_j \| \leq \epsilon$, sendo ϵ um limiar pré-definido. Ao se utilizar esta abordagem, se ϵ for um valor muito baixo, deve-se fazer um pós-processamento para que não haja vértices isolados no grafo. Por outro lado, se ϵ for um valor muito alto, a rede gerada pode ser muito densa. Devido à propriedade de simetria contida nesta

abordagem, o grafo gerado é considerado não direcionado. O algoritmo multirrótulo $ML\epsilon$ implementa essa abordagem.

Na abordagem baseada em vizinhos mais próximos existem duas variações para a construção do grafo: direcionado e não direcionado. No primeiro caso, escolhe-se inicialmente k vizinhos para cada exemplo E_i , e gera-se uma aresta direcionada entre E_i e E_j , se E_j estiver entre os k vizinhos mais próximos de E_i . Essa abordagem, utilizada pelo algoritmo multirrótulo $BRkNN-b$, é semelhante à abordagem do método kNN padrão, no qual o número n_E de arestas por vértice é igual a k , i.e., $n_E = k$. Para o segundo caso (grafo não direcionado), propomos em (Cherman et al., 2013a) a utilização de duas estratégias: vizinhos mútuos e vizinhos não mútuos.

Na estratégia vizinhos mútuos, uma aresta não direcionada entre E_i e E_j é criada s.s.s E_j está no conjunto de vizinhos mais próximos de E_i e, por sua vez, E_i também está no conjunto de vizinhos mais próximos de E_j . Por ser uma estratégia mais restritiva na criação de arestas, o grafo construído pode ser mais desconexo, i.e., $n_E \leq k$. O algoritmo $MLMUT$ implementa essa estratégia. Na estratégia vizinhos não mútuos, uma aresta não direcionada entre E_i e E_j é criada se E_j está no conjunto de vizinhos mais próximos de E_i ou se E_i está no conjunto de vizinhos mais próximos de E_j . Assim, o grafo criado com esta abordagem pode conter $n_E \geq k$ arestas por vértice. O algoritmo $MLnotMUT$ implementa essa estratégia. Em todos os casos, para classificar um novo exemplo $E = (\mathbf{x}, ?)$, ele é inserido no grafo como um novo vértice respeitando a abordagem que o algoritmo multirrótulo utiliza. O cálculo para a predição do multirrótulo considera os n_E exemplos conectados com E .

A complexidade do algoritmo $ML\epsilon$ é igual ao do $BRkNN$: $O(N \times M)$, onde N é o número de exemplos no conjunto de treinamento e M o número de atributos descritores do problema. Para os algoritmos $MLMUT$ e $MLnotMUT$, as complexidades são respectivamente $O(k \times N \times M)$ e $O(N^2 \times M)$. No entanto, a eficiência desses algoritmos pode ser melhorada utilizando recursos como estruturas de dados especiais (Liu et al., 2010).

Experimentos realizados com dez conjuntos de dados sugerem que as abordagens $MLMUT$ e $MLnotMUT$ são superiores à abordagem tradicional do kNN utilizada para definição dos vizinhos mais próximos. Os métodos $MLMUT$ e $MLnotMUT$ apresentaram os melhores resultados quando comparados aos métodos BR e $BRkNN$, principalmente se considerada as medidas de avaliação F -Measure e Accuracy, nas quais obtiveram o melhor desempenho médio. Em alguns casos, os métodos

por nós propostos obtiveram melhora com diferença estatisticamente significativa em relação ao *BR*. Vale destacar também que o método *MLMUT* obteve um *ranking* médio superior ao algoritmo multirrótulo *BRkNN-b* em todas as medidas de avaliação, o que sugere uma estratégia de construção de grafos mais eficaz que a usada pelo *BRkNN*.

Essas três propostas, *MLMUT*, *MLnotMUT* e *ML ϵ* , foram implementadas utilizando o *framework* *Mulan* (Tsoumakas et al., 2011) e encontram-se disponíveis em <http://www.labic.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/ENIAC2013.zip>

BP-MLL O BP-MLL (Zhang, 2006) é uma adaptação do algoritmo *back-propagation* para o aprendizado multirrótulo. A principal modificação proposta para o algoritmo é a introdução de uma nova função de erro que considera múltiplos rótulos no cálculo.

MMP O Multi-class Multi-label Perceptron, proposto por Crammer et al. (2003), é um algoritmo da família dos ranqueadores de rótulos baseado no perceptron (Haykin, 1998; Minsky e Papert, 1969). O MMP cria um perceptron para cada rótulo, mas a atualização dos pesos para cada perceptron é realizada de tal maneira que se obtenha um perfeito ranqueamento de todos os rótulos.

MMAC Esse algoritmo segue o paradigma de classificação associativa, o qual consiste na construção de um conjunto de regras de classificação por meio da mineração de regras de associação. O MMAC aprende um conjunto inicial de regras de classificação, remove todos os exemplos associados a esse conjunto e recursivamente aprende um novo conjunto de regras com os exemplos restantes até que não sobrem mais itens frequentes. Esses múltiplos conjuntos de regras devem conter regras com pré-condições similares, mas com diferentes rótulos no consequente. Para tais regras, é realizado um *merge* para gerar uma única regra multirrótulo.

HOMER Esse método foi proposto por Tsoumakas et al. (2008) com o objetivo de ser efetivo e computacionalmente eficiente na classificação multirrótulo em domínios com muitos rótulos. Para tanto, o método utiliza a estratégia *dividir para conquistar*, na qual é construída uma hierarquia de classificadores multirrótulo, cada um responsável por um conjunto muito pequeno de rótulos quando comparado ao conjunto total. Na Figura 2.2 é ilustrada uma hierarquia de rótulos e classificadores desse método.

Dessa maneira, o método HOMER melhora a performance preditiva a

Figura 2.2: Hierarquia ilustrativa de rótulos e classificadores do método HOMER (Tsoumakas et al., 2008).

custo de um algoritmo com complexidade linear para treinamento e logarítmica para teste em relação à $|L|$.

2.2.3 Classificação a partir de *ranking*

A ideia básica desse tipo de abordagem de classificação multirrótulo consiste basicamente em obter um *ranking* de rótulos para um determinado exemplo ou um *ranking* de exemplos para um determinado rótulo. A partir desse *ranking* é definido um corte para a definição de quais rótulos serão atribuídos aos exemplos a serem preditos⁴. As diversas técnicas propostas são variações dessa ideia básica. A seguir são apresentados as seguintes abordagens para classificação multirrótulo por meio de *ranking*:

- *Rank-based cut* – RCut (Yang, 2001);
- *Proportion-based cut* – PCut (Yang, 2001);
- *Score-based local optimization* – SCut (Yang, 2001);
- RTCut (Yang, 2001);
- SCutFBR (Yang, 2001);
- *Content-based MetaLabeler* – MetaC (Tang et al., 2009);

RCut Baseado no *ranking* de rótulos obtidos, um exemplo E_i é classificado com os t primeiros rótulos do *ranking*, onde t é um parâmetro definido pelo usuário. Geralmente, o t é definido utilizando o número médio de rótulos atribuídos aos exemplos do conjunto de treinamento, *i.e.*, a cardinalidade de rótulos. Supondo que os exemplos no conjunto de

⁴Esse tipo de estratégia também pode ser entendida como calibração de limiar.

treinamento apresentem em média 3,5 rótulos, o valor de t poderia ser definido como 3 ou 4.

PCut Nessa abordagem, os exemplos do conjunto de teste são ordenados para cada rótulo $y_i \in L$ pela confiança de predição nesse rótulo. Para os t primeiros exemplos do *ranking*, é atribuído o rótulo y_i . O parâmetro t é computado baseado na probabilidade *a priori* estimada no conjunto de treinamento. Essa abordagem é aplicável apenas para predição de exemplos em lote, *i.e.*, ela necessita de todos os exemplos de testes para a realização do *ranking* (cenário de inferência transdutiva) e, por esse motivo, raramente é utilizada em aplicações nas quais uma nova predição deve ser realizada exemplo a exemplo (cenário de inferência indutiva).

SCut Nessa abordagem é utilizado um conjunto de exemplos de validação para otimizar o corte t (*threshold* de cada rótulo para uma determinada medida de avaliação definida pelo usuário). A idéia básica é dividir o conjunto de treinamento em diferentes partições (*folds*) e utilizá-las para otimizar o valor de *threshold* de cada rótulo baseado na performance de uma determinada medida de avaliação sobre esse conjunto de validação.

RTcut Nessa abordagem, o *ranking* do rótulo y_j , $j = 1..q$, para um novo exemplo E_i é dado por um *score* sintético $ss(y_j, E_i)$ definido pela Equação 2.2.

$$ss(y_j, E_i) = r(y_j, E_i) + \frac{s(y_j, E_i)}{\max_{y_k \in L} s(y_k, E_i) + 1} \quad (2.2)$$

onde $r(y_j, E_i)$ é o *ranking* do rótulo y_j para o exemplo E_i e $s(y_j, E_i)$ é o *score* do rótulo y_j para o exemplo E_i .

Os autores comentam que, desse modo, é possível realizar um ajuste fino no *ranking* e suavizar o *trade-off* entre *precision* e *recall* existente no método RCut.

SCutFBR Quando um rótulo está associado a um número extremamente pequeno de exemplos de treinamento, a abordagem SCut corre o risco de sofrer o efeito de *overfitting*, pois os dados podem não ser suficientemente representativos para determinar o valor de *corte*, o que produziria um valor muito alto (restritivo) ou muito baixo (permissivo). Um valor de corte muito restritivo resultaria em muitos falso-negativos, ou seja, exemplos que deveriam ser considerados positivos, não são pelo fato do valor corte ser muito alto. Por outro lado, um valor muito

baixo resultaria em muitos falso-positivos, *i.e.*, muitos exemplos negativos são preditos como positivo devido a um corte muito permissivo. Para resolver esse problema, [Yang \(2001\)](#) propõe duas variações para o SCutFBR. A primeira variação, chamada SCutFBR.0, define o valor do corte como infinito, *i.e.*, classifica todos os exemplos para um determinado rótulo como negativo, o que minimiza o número de falso-positivo a custo de uma pequena queda no número de verdadeiro-positivo global, se considerado todos os rótulos, já que se trata de um rótulo raro. A outra alternativa, chamada de SCutFBR.1, define o valor de corte de maneira que apenas o exemplo do topo do *ranking* seja classificado como positivo, o que causaria menos prejuízo ao número de falso-negativo.

MetaC Essa abordagem consiste em definir os rótulos de um novo exemplo por meio de duas fases ([Tang et al., 2009](#)):

1. Obter um *ranking* dos rótulos para um novo exemplo;
2. Predizer o número de rótulos do topo do *ranking* a ser retornado como resposta.

Para a primeira fase, os *scores* de um classificador multirrótulo *Binary Relevance* são utilizados para obter o *ranking* dos rótulos para um determinado exemplo. Os autores dessa abordagem utilizaram como algoritmo-base o SVM e consequentemente os *scores* fornecidos por esse algoritmo para obter o *ranking*. No entanto, qualquer algoritmo-base que forneça *scores* pode ser utilizado. Na segunda fase, o número de rótulos que vai compor a resposta final é fornecido por meio de um modelo construído em duas etapas:

1. Construção de um meta conjunto de dados;
2. Aprendizado de uma meta-modelo.

A construção de um meta conjunto de dados na primeira etapa consiste em criar um novo atributo que representa os rótulos. Esse atributo irá conter a quantidade de rótulos simples que cada exemplo de treinamento possui. Assim, o(s) atributo(s) que representam os rótulos são descartados e esse novo atributo passa a ser a classe do problema. Na Tabela [2.5](#) é exemplificada a construção de um meta conjunto de dados.

Construído o meta conjunto de dados, um algoritmo de aprendizado multi-classe constrói um meta-modelo que faz a predição da quantidade de rótulos t de um novo exemplo e , desse modo, os t primeiros

Tabela 2.5: Ilustração da construção de um meta conjunto de dados no algoritmo MetaC.

(a) Conjunto de dados original		(b) Meta conjunto de dados	
	Y		$Meta-R\acute{o}tulo$
E_1	$Y_1 = \{y_2, y_3\}$	E_1	2
E_2	$Y_2 = \{y_1, y_3, y_4\}$	E_2	3
E_3	$Y_3 = \{y_4\}$	E_3	1
E_4	$Y_4 = \{y_2, y_3\}$	E_4	2

rótulos do *ranking* obtido na primeira fase são retornados como resposta final do problema multirrótulo.

2.3. Avaliação do aprendizado

Diversas medidas para a classificação multirrótulo, descritas a seguir, foram propostas para avaliar a eficácia do modelo gerado pelo algoritmo de aprendizado. Na definição dessas medidas, D representa um conjunto de dados multirrótulo composto por $|D| = N$ exemplos $(\mathbf{x}_i, Y_i)$, $i = 1 \dots N$ e $Y_i \subseteq L$ o conjunto de rótulos (multirrótulo) do i -ésimo exemplo. H é um classificador multirrótulo com $Z_i = H(\mathbf{x}_i)$ o multirrótulo predito por H para um dado $\mathbf{x}_i$.

As medidas de avaliação de classificação monorrótulo consideram apenas dois estados possíveis para a predição de um exemplo: correta ou incorreta. Nesse caso, medidas convencionais como *Accuracy*, *F-Measure* e área sob a curva ROC (*AUC*) são geralmente utilizadas (Flach, 2012). Por outro lado, a predição de um exemplo pode estar *parcialmente* correta na classificação multirrótulo, o que exige medidas de avaliação específicas para esse problema.

Diversas medidas específicas foram propostas para a avaliação da classificação multirrótulo. Conforme Zhang e Zhou (2013), essas medidas podem ser agrupadas em:

1. *Baseadas em exemplo*
2. *Baseadas em rótulo*

As medidas do primeiro grupo calculam a eficácia do classificador para cada exemplo multirrótulo separadamente para posteriormente calcular uma média sobre todos os exemplos. Já as medidas baseadas em rótulo são calculadas utilizando medidas de avaliação binárias monorrótulo aplicadas

para cada rótulo separadamente e uma média geral é posteriormente calculada. Nesse caso, a média sobre os valores obtida para cada rótulo pode ser calculada da maneira *Micro* ou *Macro*, como descrito em mais detalhes na Seção 2.3.2, página 30.

Além disso, cada grupo pode ser subdivido em medidas que avaliam a qualidade da *classificação* e medidas que aferem a qualidade da ordenação (*ranking*) dos exemplos. Na Figura 2.3, as principais medidas de avaliação de aprendizado multirrótulo são apresentadas e relacionadas aos seus respectivos grupos.

Figura 2.3: Principais medidas de avaliação do aprendizado multirrótulo. Figura adaptada de (Zhang e Zhou, 2013).

2.3.1 Medidas baseadas em exemplo

Uma medida tradicional utilizada para a avaliação de classificadores multirrótulo é a *Hamming-Loss* (Tsoumakas e Katakis, 2007), definida pela Equação 2.3.

$$Hamming-Loss(H, D) = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \Delta Z_i|}{|L|} \quad (2.3)$$

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 2.4.

$$Subset-Accuracy(H, D) = \frac{1}{N} \sum_{i=1}^N I(Z_i = Y_i) \quad (2.4)$$

onde I é uma função que mapeia verdadeiro em 1 e falso em 0. *Subset-Accuracy* é uma medida que não leva em conta flutuações ou resultados

parcialmente corretos, uma vez que a avaliação $Z_i = Y_i$ só possui duas possíveis respostas.

Medidas tradicionalmente relacionadas ao aprendizado monorrótulo foram adaptadas para o contexto multirrótulo para considerar exemplos com predição parcialmente correta. Essas medidas são descritas a seguir.

De maneira intuitiva, a medida *Accuracy*, definida pela Equação 2.5, representa os acertos obtidos da predição dos rótulos verdadeiros⁵ e também considera o fato de não predizer um rótulo que deveria ser predito.

$$Accuracy(H, D) = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (2.5)$$

A medida *Precision*, definida pela Equação 2.6, considera apenas se os rótulos preditos (positivamente) realmente deveriam ter sido preditos. Os rótulos não preditos não interferem no valor da medida.

$$Precision(H, D) = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap Z_i|}{|Z_i|} \quad (2.6)$$

A medida *Recall*, definida pela Equação 2.7, no entanto, considera a quantidade de rótulos que foram preditos ou deixados de ser preditos. 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.

$$Recall(H, D) = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap Z_i|}{|Y_i|} \quad (2.7)$$

Já a medida *F-Measure*, definida pela Equação 2.8, representa uma média harmônica entre a *Precisão* e o *Recall*, ou seja, um equilíbrio entre as duas medidas.

$$F-Measure(H, D) = \frac{1}{N} \sum_{i=1}^N \frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \quad (2.8)$$

Os valores das medidas *Hamming-Loss*, *Subset-Accuracy*, *Precision*, *Accuracy*, *Recall* e *F-Measure* variam todos no intervalo $[0, 1]$. Como mencionado, o valor ótimo para *Hamming-Loss* é 0 (zero), enquanto que esse valor é 1 (um) para as outras medidas.

Para ilustrar, na Tabela 2.6 é apresentada a predição de um conjunto de teste com cinco exemplos multirrótulos, considerando $L = \{a, b, c, d\}$ e as

⁵Considere “rótulos verdadeiros” como os rótulos corretos do multirrótulo de um exemplo. Na fase de treinamento, os rótulos verdadeiros são utilizados para treinar o classificador multirrótulo, enquanto que na fase de teste, os rótulos verdadeiros são utilizados para comparação com os rótulos preditos pelo classificador.

respectivas medidas de avaliação.

Tabela 2.6: Exemplos das medidas de avaliação

	Y_i	Z_i	$Y_i \cup Z_i$	$Y_i \cap Z_i$	<i>Acc.</i>	<i>Prec.</i>	<i>Recall</i>	<i>F</i>	<i>H. Loss</i>
E_1	$\{a, b\}$	$\{a, b\}$	$\{a, b\}$	$\{a, b\}$	1	1	1	1	0
E_2	$\{a, b\}$	$\{c, d\}$	$\{a, b, c, d\}$	$\{\emptyset\}$	0	0	0	0	1
E_3	$\{a, b, c\}$	$\{a, b\}$	$\{a, b, c\}$	$\{a, b\}$	0,67	1	0,67	0,8	0,25
E_4	$\{a, b\}$	$\{a, b, c\}$	$\{a, b, c\}$	$\{a, b\}$	0,67	0,67	1	0,8	0,25
E_5	$\{a, b\}$	$\{b, c\}$	$\{a, b, c\}$	$\{b\}$	0,33	0,5	0,5	0,5	0,5

Nessa tabela, na coluna Y_i é descrito o conjunto de rótulos verdadeiro de cada exemplo e na coluna Z_i o conjunto de rótulos predito por um classificador multirrótulo. As colunas restantes mostram as medidas de avaliação para cada um dos exemplos.

Considerando os exemplos individualmente, pode ser observado que somente o exemplo E_1 foi corretamente predito, enquanto que o exemplo E_2 mostra o pior caso de classificação multirrótulo, no qual não existem rótulos simples em comum. Para o exemplo E_3 , somente a medida de precisão é máxima, pois os rótulos verdadeiros $\{a, b\}$ foram preditos corretamente. Porém, não foi predito o rótulo c desse exemplo, o que influencia negativamente as outras medidas. Por outro lado, para o exemplo E_4 , somente a medida *Recall* é máxima, pois todos os rótulos verdadeiros $\{a, b\}$ foram preditos. As outras medidas foram influenciadas negativamente devido a predição do rótulo c , o qual não faz parte do conjunto de rótulos verdadeiros. Por fim, para o exemplo E_5 é observado um equilíbrio entre *Precision* e *Recall*, o que faz o valor de *F-Measure* ser igual a essas duas medidas.

Finalmente, o valor dessas medidas para o conjunto de treinamento com cinco exemplos é:

- *Accuracy*: 0,53;
- *Precision*: 0,63;
- *Recall*: 0,63;
- *F-Measure*: 0,63;
- *Hamming-Loss*: 0,4.

É possível realizar o *ranking* dos rótulos para um dado exemplo quando o algoritmo de aprendizado fornece, além do classificador, uma função que retorna valores (*scores*) para a ordenação dos rótulos conforme sua relevância para o exemplo. O *ranking* do exemplo E_i predito para um rótulo $y \in L$ é denotado por $r_i(y)$. Ao rótulo mais relevante é atribuída a primeira posição no *ranking*, e ao menos relevante a última posição no *ranking*. Algumas

medidas foram propostas para avaliar a qualidade dessa ordenação e são descritas a seguir.

A medida *1Error*, definida pela Equação 2.9, representa quantas vezes o rótulo ranqueado no topo não pertence ao conjunto Y_i de rótulos verdadeiros do exemplo considerado. Essa medida varia no intervalo $[0, 1]$.

$$1Error = \frac{1}{N} \sum_{i=1}^N I(\underset{y \in L}{argmin} r_i(y)) \quad (2.9)$$

Outra medida baseada em rankings é a *Coverage*, definida pela Equação 2.10, na qual a idéia intuitiva é identificar a posição no *ranking* antes da qual estão todos rótulos que deveriam ser preditos. Quanto mais perto do topo do *ranking* for essa posição, melhor será o valor dessa medida, pois significa que menos rótulos erroneamente ordenados estão presentes entre os rótulos verdadeiros.

$$Coverage = \frac{1}{N} \sum_{i=1}^N \max_{y \in Y_i} r_i(y) - 1 \quad (2.10)$$

A medida *Ranking-Loss* conta quantas vezes rótulos irrelevantes são ranqueados acima de rótulos relevantes, ou seja, quantas vezes um rótulo $y \in Y_i$ tem um ranking pior que um rótulo $y' \in L - Y_i$. A medida é definida pela Equação 2.11.

$$Ranking-Loss = \frac{1}{N} \sum_{i=1}^N \frac{1}{|Y_i| | \bar{Y}_i |} |(y_a, y_b) : r_i(y_a) < r_i(y_b), (y_a, y_b) \in Y_i \times (L - Y_i)| \quad (2.11)$$

Por fim, a medida *AveragePrecision*, definida pela Equação 2.12, representa, em média, a fração de rótulos em Y_i ranqueados acima de um determinado rótulo $y' \in L - Y_i$.

$$AveragePrecision = \frac{1}{N} \sum_{i=1}^N \frac{1}{|Y_i|} \sum_{y \in Y_i} \frac{|(y' \in Y_i : r_i(y') \leq r_i(y))|}{r_i(y)} \quad (2.12)$$

2.3.2 Medidas baseadas em rótulos

Para classificadores binários monorrótulo, as duas classes (rótulos) são, geralmente, denominadas positiva (+) e negativa (-). Uma maneira natural de avaliar esses classificadores é por meio dos valores de V_P , F_P , V_N e F_N , os quais correspondem, respectivamente, a quantidade de verdadeiros/falsos⁶

⁶Quando 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

positivo/negativo 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 2.13, 2.14, 2.15 e 2.16, são calculadas.

$$Precisao(H, D) = \frac{V_P}{V_P + F_P} \quad (2.13) \quad Recall(H, D) = \frac{V_P}{V_P + F_N} \quad (2.14)$$

$$Acuracia(H, D) = \frac{V_P + V_N}{N} \quad (2.15) \quad F(H, D) = \frac{2V_P}{2V_P + F_P + F_N} \quad (2.16)$$

A área sob a curva ROC (AUC) é uma medida tradicionalmente utilizada no aprendizado monorrótulo binário para avaliar a qualidade da ordenação (*ranking*) dos exemplos (Flach, 2012). Desse modo, ela também pode ser utilizada no contexto das medidas baseadas em rótulos de avaliação multirrótulo.

Para um classificador multirrótulo com $L = \{y_1, y_2, \dots, y_q\}$, duas medidas baseadas em rótulo podem ser utilizadas, denominadas *macro-averaged* e *micro-averaged* (Tsoumakas et al., 2009b).

Seja $B(V_P, V_N, F_P, F_N)$ uma medida de classificação binária, calculada em função de uma matriz de confusão. Seja $T_{P_{y_i}}$, $F_{P_{y_i}}$, $T_{N_{y_i}}$ e $F_{N_{y_i}}$ o número de verdadeiros/falsos positivos/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 2.17 e 2.18.

$$B_{macro} = \frac{1}{q} \sum_{i=1}^q B(T_{P_{y_i}}, F_{P_{y_i}}, T_{N_{y_i}}, F_{N_{y_i}}) \quad (2.17)$$

$$B_{micro} = B\left(\sum_{i=1}^q T_{P_{y_i}}, \sum_{i=1}^q F_{P_{y_i}}, \sum_{i=1}^q T_{N_{y_i}}, \sum_{i=1}^q F_{N_{y_i}}\right) \quad (2.18)$$

Conceitualmente, *macro* calcula a eficácia do aprendizado considerando pesos iguais para rótulos, enquanto que *micro* considera pesos iguais para exemplos. Em outras palavras, a abordagem *macro* calcula uma medida de avaliação binária para cada *rótulo* separadamente e então retorna a média sobre todas as medidas. Já a abordagem *micro* calcula o operador binário para cada *exemplo* e então retorna a média sobre todos os valores calculados. Isso significa que a abordagem *macro* é afetada igualmente por rótulos independente de suas frequências no conjunto de dados, o que pode ser um comportamento desejado quando se quer avaliar problemas com rótulos desbalanceados. Além disso, vale observar que para alguns operadores binários, *macro* e *micro* apresentam o mesmo valor, por exemplo:

$$Accuracy_{micro} = Accuracy_{macro} = (1 - HammingLoss).$$

exemplos negativos.

Para exemplificar, considere que foi aplicado um classificador multirrótulo H sobre um conjunto de 40 exemplos de teste com $L = \{a, b, c, d\}$. Para cada um dos rótulos simples $y_i \in L$ é gerada uma matriz de confusão, apresentadas na Tabela 2.7, a partir da avaliação do classificador multirrótulo H .

Tabela 2.7: Matrizes de confusão para um conjunto D com $L = \{a, b, c, d\}$.

(a) Rótulo a	(b) Rótulo b	(c) Rótulo c	(d) Rótulo d																
<table><tr><td>20 V_P</td><td>0 F_N</td></tr><tr><td>0 F_P</td><td>20 V_N</td></tr></table>	20 V_P	0 F_N	0 F_P	20 V_N	<table><tr><td>20 V_P</td><td>10 F_N</td></tr><tr><td>0 F_P</td><td>10 V_N</td></tr></table>	20 V_P	10 F_N	0 F_P	10 V_N	<table><tr><td>10 V_P</td><td>0 F_N</td></tr><tr><td>20 F_P</td><td>10 V_N</td></tr></table>	10 V_P	0 F_N	20 F_P	10 V_N	<table><tr><td>10 V_P</td><td>10 F_N</td></tr><tr><td>10 F_P</td><td>10 V_N</td></tr></table>	10 V_P	10 F_N	10 F_P	10 V_N
20 V_P	0 F_N																		
0 F_P	20 V_N																		
20 V_P	10 F_N																		
0 F_P	10 V_N																		
10 V_P	0 F_N																		
20 F_P	10 V_N																		
10 V_P	10 F_N																		
10 F_P	10 V_N																		

Na Tabela 2.8 são apresentados os resultados para B_{macro} e B_{micro} , considerando as medidas de classificação binária calculadas sobre as matrizes de confusão da Tabela 2.7.

Tabela 2.8: Valores das medidas *micro-averaged* e *macro-averaged* para as quatro medidas binárias apresentadas.

	Acurácia	Precisão	Recall	F
<i>macro-averaged</i>	0,69	0,75	0,83	0,70
<i>micro-averaged</i>	0,69	0,67	0,75	0,71

2.3.3 Classificadores *baseline*

A adoção de classificadores considerados *baselines* é uma prática comum em aprendizado monorrótulo. O classificador criado utilizando, por exemplo, a classe majoritária presente no conjunto de dados é frequentemente utilizado como uma referência de comparação com outros classificadores. Com a utilização dessa prática, é possível identificar se o erro dos classificadores apresenta algum ganho real se comparado, por exemplo, ao erro majoritário do conjunto de dados, obtido por um classificador *baseline* que não teve acesso aos atributos descritores.

Apesar de ser uma prática consolidada em classificação monorrótulo, o conceito de *baselines*, no estilo do erro majoritário, é pouco explorado na classificação multirrótulo. Desse modo, em (Metz et al., 2012), propomos abordagens *baselines* para classificação multirrótulo.

O primeiro aspecto a ser considerado na criação de tais *baselines* é a grande gama de medidas de avaliação presentes no aprendizado multirrótulo. No cenário monorrótulo, apesar da existência de mais de uma medida de avaliação, o erro⁷ é frequentemente adotado como uma métrica de

⁷Proporção de exemplos erroneamente classificados no conjunto de teste.

avaliação, o que torna o erro majoritário também uma prática comum de *baseline*. No entanto, em classificação multirrótulo, frequentemente são adotadas diversas medidas de avaliação para aferir diferentes aspectos da qualidade de predição dos classificadores. Desse modo, diferentes classificadores *baselines*, que consideram apenas os atributos descritores, podem ser elaborados. Cada *baseline* pode maximizar/minimizar uma medida de avaliação diferente.

A medida *Hamming-Loss*, definida anteriormente pela Equação 2.3, página 27, considera a porcentagem de erros cometidos pelo classificador. Nessa medida, o erro de predizer positivamente um rótulo que não deveria ser predito tem o mesmo peso de não predizer um rótulo que deveria ser predito, ou seja, a medida afere independentemente o erro de cada rótulo $y_j \in L$. Assim, um *baseline* que minimize a Equação 2.3 do *Hamming-Loss* consiste em minimizar o erro de cada rótulo $y_j \in L$, ou seja, é necessário definir um multirrótulo Z que minimize a equação. Como N e $|L|$ são constantes nessa equação, o *baseline* pode ser obtido minimizando $\sum_{i=1}^N |Y_i \Delta Z|$, onde

$$|Y_i \Delta Z| = \sum_{j=1}^{|L|} I(y_{ij} \neq z_j) \quad (2.19)$$

e y_{ij} se refere ao valor do rótulo y_j no multirrótulo Y_i do exemplo E_i , ou seja, o valor é positivo se $y_j \in Y_i$ e negativo caso contrário. Para minimizar *Hamming-Loss*, é necessário maximizar a possibilidade de que $y_j \in Z$, ou seja, todos os rótulos $y_j \in L$ com frequência maior que 50% devem ser incluídos em Z . Desse modo, o classificador *baseline* para *Hamming-Loss* fornecerá sempre o multirrótulo Z que contém os rótulos com frequência maior que 50% como predição para todos os exemplos.

A métrica *Subset-Accuracy*, definida na Equação 2.4, página 27, é uma medida de avaliação bastante rígida que requer que o multirrótulo predito seja exatamente igual ao multirrótulo verdadeiro. Desse modo, da maneira análoga à classe majoritária em aprendizado monorrótulo, essa medida pode ser maximizada predizendo sempre o *multirrótulo majoritário*, i.e., a combinação de rótulos mais frequente no conjunto de dados.

A medida *F-Measure* é frequentemente utilizada para aferir a qualidade de classificadores. Ela é definida como uma média harmônica entre *Precision* e *Recall*, definidas em classificação multirrótulo respectivamente pelas Equações 2.6 e 2.7, página 28. Definir um multirrótulo único Z que maximize a medida *Recall* é trivial. Pode-se simplesmente considerar $Z = L$, ou seja, predizer sempre todos os rótulos do problema. No entanto, encontrar um Z que maximize a medida *Precision* não é tão trivial, pois $|Z|$ pertence

ao denominador na equação que o define. Desse modo, um procedimento similar ao explicado a seguir para a medida *F-Measure* deve ser utilizado para encontrar um multirrótulo Z que maximize *Precision*.

Para maximizar *F-Measure*, é necessário encontrar um multirrótulo Z que maximize a Equação 2.8, página 28. Isso pode ser realizado maximizando $|Y_i \cap Z|$ com a menor quantidade de rótulos possíveis, já que $|Z|$ está presente no denominador da equação. O Algoritmo 2 é utilizado para encontrar tal Z que maximize a medida *F-Measure*. O método *baseline* para *F-Measure* é chamado de F_B .

```

Entrada:  $V$ : Lista de rótulos de  $L$  ordenada pela frequência;
 $D$ : Conjunto de dados multirrótulo
 $max \leftarrow 0$ 
 $Z \leftarrow Z_m \leftarrow \emptyset$ 
for  $i \leftarrow 1$  to  $q$  do
     $Z_m \leftarrow Z_m \cup \{V_i\}$ 
     $P \leftarrow F\text{-Measure}(Z_m, D)$ 
    if  $max < P$  then
         $max \leftarrow P$ 
         $Z \leftarrow Z_m$ 
    end
end
Retorna  $Z$ 

```

Algoritmo 2: Encontra o multirrótulo Z que maximiza *F-Measure*

Neste trabalho, os métodos *baseline* para *Hamming-Loss*, *Subset-Accuracy* e *F-Measure* são chamados, respectivamente, HL_B , $SubAcc_B$ e F_B . Essas abordagens encontram *baselines* que maximizam/minimizam uma medida específica. No entanto, outras medidas não são necessariamente maximizadas ou minimizadas. Considerando essa desvantagem e o fato de que diversas medidas são frequentemente utilizadas no aprendizado multirrótulo para aferir a qualidade dos classificadores, em (Metz et al., 2012) propomos um método *baseline* geral, ou seja, projetado para não maximizar ou minimizar uma medida específica, mas para apresentar resultados adequados a um *baseline* para uma gama maior de medidas. Esse método foi chamado $Geral_B$. A idéia do $Geral_B$ é construir um *ranking* dos rótulos contidos em L de acordo com a frequência no conjunto de dados. Os t primeiros rótulos (mais frequentes) compõem o multirrótulo Z que será predito por esse *baseline*, onde t é o número inteiro mais próximo da cardinalidade (número médio de rótulos por exemplo) do conjunto de dados.

Os classificadores *baselines* específicos, bem como o $Geral_B$, foram implementados utilizando o *framework* *Mulan* e estão disponíveis em <http://www.labic.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/mlbaseline-code.zip>.

Uma análise comparativa entre os *baselines* específicos e o $Geral_B$ utilizando oito conjuntos de dados foi realizada em (Metz et al., 2012). O objetivo dessa análise é verificar a hipótese por trás de cada método *baseline*, ou seja, apresentar empiricamente que os *baselines* específicos apresentem melhor desempenho que os outros *baselines* em suas correspondentes medidas de avaliação, enquanto que o $Geral_B$ não é muito degradado para nenhuma medida considerada.

Na Tabela 2.9 é apresentado o *ranking* médio sobre todos os oito conjunto de dados de cada método *baseline* para as três medidas de avaliação consideradas: *Subset-Accuracy*, *Hamming-Loss* e *F-Measure*.

Tabela 2.9: Ranking médio dos métodos *baseline*.

	$SubAcc_B$	HL_B	F_B	$Geral_B$
<i>Subset-Accuracy</i>	1	3	4	2
<i>Hamming-Loss</i>	3	1	4	2
<i>F-Measure</i>	3	4	1	2

Nessa tabela, é possível observar que o $Geral_B$ apresentou o segundo melhor *ranking* médio para todas as medidas de avaliação, perdendo apenas para o *baseline* específico da medida correspondente. Os métodos específicos tiveram maior degradação nas medidas que fogem seu foco.

Com a utilização desse tipo de abordagem é possível obter mais evidências sobre o poder de aprendizado dos algoritmos multirrótulo. O $Geral_B$ é mais adequado sempre que se deseja considerar várias medidas de avaliação, enquanto que *baselines* específicos podem ser utilizados quando se deseja minimizar/maximizar uma determinada medida de avaliação.

Para comparar os resultados dos métodos *baselines* com resultados reportados na literatura, foi realizada uma revisão sistemática em aprendizado multirrótulo com o objetivo de coletar e categorizar resultados experimentais reportados nessa area (Spolaôr et al., 2013a). Um total de 1543 trabalhos foram inicialmente coletados nessa revisão, os quais foram posteriormente reduzidos a 64 utilizando critérios de exclusão.

Para exemplificar os dados coletados, na Figura 2.4 é apresentada a distribuições dos conjuntos de dados nos 64 trabalhos considerados.

Os resultados experimentais reportados nos 64 trabalhos considerados foram comparados com os resultados obtidos pelo método $Geral_B$. Do total de 5342 valores de medidas de avaliação coletados, 12,8% foram piores ou iguais aos resultados obtidos pelo método *baseline*. Os resultados reportados para o conjunto de dados *corel5k* foram em 43% dos casos iguais ou piores aos apresentados pelo $Geral_B$. Por outro lado, para *emotions* apenas 0,6% dos resultados reportados não foram melhores que os resultados do *baseline*. Na Figura 2.5 são apresentados os resultados para as oito

Figura 2.4: Distribuição dos conjuntos de dados nos 64 trabalhos coletados na revisão sistemática (Metz et al., 2013).

medidas de avaliação mais frequentemente utilizadas nos 64 trabalhos coletados⁸.

Figura 2.5: Comparação de resultados reportados na literatura com os resultados obtidos pelo *baseline* G_{eral}_B .

Esses resultados foram submetidos à publicação (Metz et al., 2013) e evidenciam a importância da utilização de *baselines*, em paralelo a comparações com outros classificadores estado da arte, como indicativo de aprendizado de novas propostas.

⁸Todos os resultados considerados encontram-se disponíveis em <http://www.labc.icmc.usp.br/pub/mcmcard/ExperimentalResults/ML2013.pdf>

2.4. Caracterização do conjunto de dados

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

A *cardinalidade de rótulo* representa a média do número de rótulos distintos atribuídos a cada exemplo. Essa medida é independente da quantidade de rótulos simples contidos no problema — Equação 2.20.

A *densidade de rótulo* 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 2.21. Ambas as métricas estão relacionadas por $CR(D) = |L|DR(D)$.

$$CR(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} |Y_i| \quad (2.20) \quad DR(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i|}{|L|} \quad (2.21)$$

Outra característica importante dos conjuntos de dados é a frequência de cada rótulo $y_i \in L = \{y_1, y_2, \dots, y_q\}$ presente no problema. O conhecimento dessa informação possibilita principalmente identificar o grau de desbalanceamento dos rótulos. No entanto, a quantidade de rótulos presente no conjunto de dados pode facilmente ultrapassar a casa das centenas, o que impossibilita, por exemplo, descrever essa informação em uma tabela descritiva do problema. Desse modo, como sugerido por Tsoumakas (2013), introduzimos neste trabalho medidas de *quartil* para caracterização da distribuição da frequência dos rótulos no conjunto de dados. O *quartil* é um dos três valores que divide um conjunto ordenado de dados em quatro partes iguais, de modo que cada parte contém 25% dos indivíduos da amostra. No caso multirrótulo, o conjunto ordenado contém as frequências dos rótulos como indivíduos. Dessa forma, as seguintes medidas podem ser extraídas dos conjuntos de dados multirrótulo.

Min: Frequência mínima – Frequência do rótulo associado ao menor número de exemplos no conjunto de dados.

1Q: 1º Quartil – Frequência do rótulo até o qual se encontram 1/4 dos valores das frequência ordenada dos rótulos.

Med: 2º Quartil ou Mediana – Frequência do rótulo até o qual se encontram 2/4 dos valores das frequência ordenada dos rótulos, *i.e.*, a mediana das frequências dos rótulos.

$3Q$: 3º Quartil – Frequência do rótulo até o qual se encontram $3/4$ dos valores das frequência ordenada dos rótulos.

Max : Frequência máxima – Frequência do rótulo associado ao maior número de exemplos no conjunto de dados.

Para ilustrar, considere um conjunto de dados que contém 100 exemplos com $|L| = 9$ rótulos. Considere também o seguinte vetor ordenado de frequência de cada rótulo: $[5, 8, 10, 12, 15, 30, 35, 45, 55]$. Nesse caso, frequência mínima (Min), 1º Quartil ($1Q$), mediana (Med), 3º Quartil ($3Q$) e frequência máxima (Max) seriam respectivamente 5, 10, 15, 35 e 55. Essas informações podem ser representadas visualmente utilizando gráficos do tipo *boxplot*. Na Figura 2.6, o gráfico *boxplot* representativo desse caso ilustrativo é apresentado.

Figura 2.6: Boxplot do caso ilustrativo.

2.5. Aplicações multirrótulo

O aprendizado multirrótulo tem sido utilizado em variados domínios nos últimos anos. Nesta seção são descritas algumas das aplicações realizadas utilizando aprendizado multirrótulo, bem como os conjuntos de dados *benchmark* correspondentes e suas características.

Marketing direto Um exemplo dessa aplicação foi apresentado em (Zhang, 2006) e tem como objetivo identificar quais produtos um determinado cliente está interessado em comprar baseado no seu histórico e em dados demográficos. Nesse trabalho foi utilizado um conjunto de dados com 19 rótulos que representam categorias de produtos e foi construído um conjunto de dados contendo respostas de clientes quanto

ao interesse nos produtos. Como o cliente pode estar interessado em mais de um grupo de produtos ao mesmo tempo, essa é uma tarefa de aprendizado multirrótulo. Dessa maneira, criado o modelo multirrótulo, foi possível oferecer aos clientes um conjunto de produtos que potencialmente lhe interessavam.

Categorização de documentos textuais Diversos trabalhos relacionados à categorização de documentos textuais foram realizados. Um exemplo desse tipo de aplicação é a categorização de notícias. Uma notícia pode pertencer simultaneamente à categoria *Ciência/Tecnologia* e à *História/Cultura*, como no caso de uma notícia sobre a *Máquina de Anticítera*⁹, um artefato antigo para auxílio à navegação. Outro conjunto de dados multirrótulo muito utilizado em categorização de textos é o *Reuters Corpus Volume I – RCV1*, o qual contém mais de 800.000 notícias classificadas e disponibilizadas pela empresa *Reuters, Ltd*¹⁰. Esse tipo de aplicação apresenta em torno de 100 rótulos e 47.000 atributos descritores, dependendo da versão do conjunto de dados, o que torna necessária a aplicação de algoritmos computacionalmente eficientes.

Classificação de e-mails Um dos projetos relacionados, denominado *UC Berkeley Enron Email Analysis*, tem como objetivo classificar os e-mails da empresa Enron¹¹ em diversas categorias¹², tais como: *Negócios da empresa*, *Puramente pessoal*, *Pessoal mas com contexto de negócio*, *Humor*, *Admiração*, entre outros. As categorias são organizadas em uma hierarquia de dois níveis e um exemplo também pode ser associado a mais de uma dessas categorias. O conjunto de dados possui 1702 exemplos, 53 rótulos e, em média, 3,4 rótulos por exemplo.

Classificação de páginas Web Um exemplo desse tipo de aplicação é realizado pela *Yahoo! Directory*¹³, o qual disponibiliza um diretório de categorias e *websites* ligadas ao domínio “*yahoo.com*”. As categorias desse diretório são organizadas em uma hierarquia, a qual contém no primeiro nível 14 categorias, como *Artes e Humanidade*, *Negócios e Economia*, *Computadores e Internet*, entre outros.

Anotação de imagem e vídeo Esse tipo de tarefa consiste em identificar automaticamente por meio de classificação multirrótulo objetos ou contextos contidos em imagens e vídeos. Uma aplicação denominada

⁹http://en.wikipedia.org/wiki/Antikythera_mechanism

¹⁰<http://www.reuters.com/>

¹¹<http://www.enron.com/>

¹²http://bailando.sims.berkeley.edu/enron/enron_categories.txt

¹³<http://dir.yahoo.com/>

Scene (Boutell et al., 2004), consiste em um problema de classificação semântica de cenas, na qual uma cena da natureza pode conter múltiplos objetos tais como um campo com montanhas ao fundo. Um conjunto de dados com 2407 imagens foi construído para esse problema, contendo 6 rótulos e, em média, 1,1 rótulos por exemplo. Esse tipo de aplicação pode ser estendida para categorização de cenas de vídeos (Worring et al., 2007). O conjunto de dados *Mediamill*¹⁴ é um exemplo de dados desse tipo de aplicação, na qual foram registradas 85 horas de vídeo de fontes de notícias. No total, 43907 quadros foram capturados e categorizados em 101 rótulos/conceitos como *Futebol*, *Comida*, *Praia*, entre outros. Cada quadro contém, em média, 4,4 rótulos nesse conjunto de dados.

Identificação de funções de proteínas Uma aplicação muito utilizada em biologia consiste em identificar automaticamente funções de proteínas. Essa é uma tarefa multirrótulo, pois uma determinada proteína tem potencial para ter mais de 190 funções (Elisseeff e Weston, 2001). O conjunto de dados *Yeast* é um exemplo desse tipo de aplicação. Nesse conjunto de dados são registradas informações sobre a estrutura molecular e sobre funções da levedura *Saccharomyces cerevisiae* inteiramente sequenciada. O conjunto contém 2417 exemplos, rotulados com até 14 funções extraídas do primeiro nível do catálogo de funções *FunCat*¹⁵. Os exemplos contém, em média, 4,2 rótulos nessa conjunto de dados.

Sugestões de Tags em sistemas Web 2.0 A sugestão de *Tags* pode ser descrita como o processo de atribuir a objetos de informação pequenas descrições textuais ou palavras-chaves (chamadas de *Tags*). É uma abordagem simples de organização de informações que foi muito utilizada na ultima década. Publicações científicas, por exemplo, são frequentemente acompanhadas de palavras-chave que podem ser atribuídas livremente pelos autores ou selecionadas de uma ontologia, por exemplo a *ACM Computing Classification*¹⁶. Um exemplo de conjunto de dados de sugestões de *Tags* é o *Delicious*¹⁷, o qual tem como objetivo atribuir *tags* aos *websites* favoritos em um sistema Web 2.0. O desafio desse tipo de aplicação é o grande número de *tags* envolvidas e a necessidade de rápida predição online das *tags*. O *benchmark* desse conjunto de dados contém 983 rótulos e 160105 exemplos com 19 ró-

¹⁴<http://www.science.uva.nl/research/mediamill/>

¹⁵http://mips.helmholtz-muenchen.de/proj/funcatDB/search_main_frame.html

¹⁶<http://www.acm.org/about/class/how-to-use>

¹⁷<http://del.icio.us>

tulos por exemplo, em média.

2.6. Descrição dos conjuntos de dados utilizados neste trabalho

Vários conjunto de dados *benchmarks* relacionados às aplicações de diferentes domínios foram disponibilizados pela comunidade. Na Tabela 2.10 são descritas as características dos conjuntos de dados utilizadas neste trabalho. A primeira coluna *ID* se refere ao identificador do conjunto de dados, o qual será utilizado para simplificar a referência aos conjuntos de dados utilizados no decorrer do trabalho. A coluna *Nome* se refere a maneira normalmente utilizada pela comunidade para referenciar o conjunto de dados; a coluna *Domínio* é relacionada ao tipo de dados utilizado; as colunas *#Ex* e *#Atrib* indicam respectivamente o número de exemplos e de atributos no conjunto de dados; a coluna $|L|$ apresenta a quantidade de rótulos presentes no problema; e as colunas $CR(D)$ e $DR(D)$ se referem respectivamente à cardinalidade de rótulo e à densidade de rótulo do conjunto de dados. As últimas cinco colunas apresentam as informações que caracterizam a distribuição de frequência dos rótulos, descritas na Seção 2.4, página 37. Todos os conjuntos foram obtidos por meio dos websites Mulan¹⁸ e Meka¹⁹.

Tabela 2.10: Conjuntos de dados *benchmark* utilizados neste trabalho.

ID	Nome	Domínio	#Ex	#Atrib	$ L $	$CR(D)$	$DR(D)$	#Dist	Distribuição de rótulos				
									Min	$1Q$	Med	$3Q$	Max
#1	bibtex	texto	7395	1836	159	2.402	0.015	2856	51	61	82	129	1042
#2	cal500	música	502	68	174	26.044	0.150	502	5	15	39	109	444
#3	corel16k	imagem	13811	500	161	2.867	0.018	4937	25	67	115	264	3170
#4	corel5k	imagem	5000	499	374	3.522	0.009	3175	1	6	15	39	1120
#5	emotions	música	593	72	6	1.869	0.311	27	148	166	170	185	264
#6	enron	texto	1702	1001	53	5.31	0.064	753	1	13	26	107	913
#7	llog	texto	1460	1004	75	1.18	0.02	304	1	4	11	22	171
#8	medical	texto	978	1449	45	1.245	0.028	94	1	2	8	34	266
#9	ohsumed	texto	13929	1002	23	1.663	0.007	1147	135	386	712	1220	3952
#10	scene	imagem	2407	294	6	1.074	0.179	15	364	404	429	432	533
#11	slashdot	texto	3782	1079	22	1.18	0.05	156	0	26	179	250	584
#12	tmc2007	texto	28596	500	22	2.158	0.098	1341	403	548	1483	2914	16918
#13	yeast	biologia	2417	103	14	4.237	0.303	198	34	324	659	953	1816

Na Figura 2.7 são apresentados gráficos do tipo *boxplot* que representam a distribuição de frequência dos rótulos de cada conjunto de dados.

¹⁸<http://mulan.sourceforge.net/datasets.html>

¹⁹<http://meka.sourceforge.net/#datasets>

#1) bibtex

#2) cal500

#3) corel5k

#4) corel16k

#5) emotions

#6) enron

#7) llog

#8) medical

#9) ohsumed

#10) scene

#11) slashdot

#12) tmc2007

#13) yeast

Figura 2.7: Gráficos do tipo *boxplot* representando a distribuição de frequências dos rótulos de cada conjunto de dados.

Dependência de Rótulos

DEPENDÊNCIA de rótulos é uma característica intrínseca do problema de aprendizado multirrótulo. O benefício que pode ser obtido pela consideração das relações entre os rótulos no processo de aprendizado é um consenso na comunidade da área. Mais do que isso, a dependência de rótulos é um fator essencial para diferenciar o aprendizado multirrótulo do aprendizado monorrótulo. Um problema de aprendizado que não possui dependência de rótulos, *i.e.*, os rótulos que participam dos multirrótulos são independentes, pode ser resolvido com os métodos de aprendizado monorrótulo existentes de maneira isolada para cada rótulo (abordagem *Binary Relevance* padrão). Em outras palavras, o problema pode ser resolvido com a junção de vários problemas monorrótulo independentes.

No entanto, a maioria dos problemas multirrótulo apresenta algum grau de dependência de rótulo, o que a torna importante para obtenção de modelos preditivos eficazes. Desse modo, alguns desafios relacionados a esse tema se tornam importante em aprendizado multirrótulo, tais como seu entendimento, identificação e exploração.

Neste capítulo são descritos conceitos e trabalhos relacionados à dependência de rótulo. Os métodos *UBC* e *BR+*, propostos neste trabalho, são apresentados. O primeiro método consiste da simulação de um classificador multirrótulo em condições ideais que tem como objetivo mensurar o grau de dependência de rótulos nos conjuntos de dados. O *BR+* é um algoritmo de aprendizado multirrótulo que foca na exploração dos benefícios

que a dependência de rótulos pode fornecer à eficácia preditiva do problema de classificação multirrótulo.

3.1. Tipos de dependência de rótulos

[Dembczynski et al. \(2010\)](#) propõem distinguir dois tipos diferentes de dependência de rótulos:

1. Dependência incondicional
2. Dependência condicional

A primeira é um tipo global de dependência, independente de qualquer observação. A segunda captura a dependência entre rótulos condicionada a um exemplo $\mathbf{x}$.

A dependência incondicional pode ser interpretada como a correlação direta entre rótulos. Nesse sentido, diversas medidas de correlação podem ser utilizadas para medir o grau de correlação entre rótulos, tais como correlação de Pearson e Informação Mútua ([Chekina et al., 2013](#)) e, assim, identificar a existência desse tipo de dependência no conjunto de dados. A existência de dependência condicional também poderia ser estimada com medidas de correlação considerando a observação $\mathbf{x}$. No entanto, calcular dessa maneira a dependência condicional teria um custo proibitivo computacionalmente.

Ambos tipos de dependência podem ser úteis para melhorar o poder preditivo de algoritmos de aprendizado multirrótulo. Porém, é mostrado em ([Dembczynski et al., 2010](#)) que são necessários diferentes algoritmos para explorar, respectivamente, dependência condicional e incondicional de rótulos.

Outra maneira de estimar a dependência de rótulos seria utilizando classificadores multirrótulo construídos de maneira análoga à proposta em ([Cherman et al., 2012](#); [Chekina et al., 2013](#)). A idéia é fornecer ao algoritmo de aprendizado a responsabilidade de aprender as dependências condicionais importantes ao problema. Essa estratégia será descrita em mais detalhes na Seção 3.2.

Diversos algoritmos foram propostos com o objetivo de explorar a dependência de rótulos presente em problemas multirrotulados. Um aspecto chave no processo de aprendizado dos métodos multirrótulo é a maneira como a dependência de rótulos é explorada. Em ([Zhang e Zhang, 2010](#)), os autores categorizam os métodos de aprendizado multirrótulo em três grupos, descritos a seguir, baseado na estratégia de dependência de rótulos que o método considera no processo de aprendizado.

Estratégia de primeira ordem. Os métodos que seguem esta estratégia consideram o aprendizado rótulo por rótulo, ignorando a coexistência de outros rótulos (Zhang e Zhou, 2007b; Boutell et al., 2004; Clare e King, 2001). O método *Binary Relevance* é o exemplo clássico desta estratégia, pois decompõe o problema multirrótulo em $|L|$ problemas monorrótulo binários independentes. A virtude desse tipo de estratégia é a simplicidade conceitual e a alta eficiência computacional. Por outro lado, a qualidade das previsões pode ser baixa devido ao pressuposto de independência entre os rótulos.

Estratégia de segundo ordem. Nesta estratégia, os métodos consideram relações pareadas entre os rótulos, como a relação de *ranking* entre o rótulo relevante e o rótulo irrelevante (Fürnkranz et al., 2008; Zhang, 2006; Elisseff e Weston, 2001) ou a relação de qualquer par de rótulos (Ueda e Saito, 2003; Zhu et al., 2005; Qi et al., 2007). A dependência de rótulo é explorada até certo ponto com esse tipo de estratégia, o que pode resultar em abordagens com boa eficácia de predição. No entanto, existem aplicações em que a efetiva exploração da dependência de rótulos necessita de estratégias que vão além das pressupostas pelas estratégias de segunda-ordem.

Estratégia de alta ordem. Os métodos desta estratégia consideram a dependência de rótulos considerando relações de alta ordem, tal como o estilo de ordem completa que considera todas as influências dos outros rótulos para cada rótulo (Godbole e Sarawagi, 2004; Tsoumakas e Vlahavas, 2007; Read et al., 2008; Cherman et al., 2012). Os métodos que seguem esta estratégia têm apresentado maior capacidade de modelar a dependência de rótulos do que os métodos das estratégias de primeira e segundo ordem. Por outro lado, requerem também um maior esforço computacional, o que torna os métodos de alta ordem menos escaláveis.

Quatro métodos se destacam nesse sentido: *Label Powerset (LP)*, *RAKEL*, *Pruned Sets (PS)*, *Classifiers Chain (CC)* e métodos baseados em *Stacking (MS ou 2BR)*, todos baseados em estratégias de alta ordem.

Em uma análise quanto ao tipo dependência de rótulo que cada método procura explorar, pode-se agrupar os quatro primeiros métodos (*LP*, *RAKEL*, *PS* e *CC*) no grupo que explora a dependência de rótulos condicional e o método original *MS* no grupo que explora a dependência incondicional (Dembczyński et al., 2012).

Como mencionado anteriormente, o método *LP* transforma o problema multirrótulo em um problema multiclasse, no qual os multirrótulos são ma-

peados em valores de meta-classes. O algoritmo de aprendizado busca mapear as relações entre os atributos descritores e as meta-classes. Desse modo, co-ocorrência entre dois ou mais rótulos são consideradas, o que o torna um algoritmo que explora a dependência de rótulos condicional. Os métodos *RAKEL* e *PS* seguem essa lógica de maneira análoga, pois também exploram a dependência de rótulos por meio do mapeamento de meta-classes.

O método *CC* explora a dependência de rótulos do tipo condicional por meio de classificadores binários, um para cada rótulo, conectados em cadeia. O primeiro classificador da cadeia é equivalente a um classificador binário gerado pelo método *BR*. O segundo classificador incorpora o atributo referente ao rótulo do classificador anterior em seus atributos descritores e assim por diante até o último classificador, o qual considera em seus atributos descritores todos os outros rótulos contidos no problema. Como os atributos descritores são considerados, em conjunto com os atributos de outros rótulos, considera-se que esse método explora a dependência de rótulos condicional.

De maneira diferente, o método original *MS* baseado em *stacking* explora dependência do tipo incondicional. Nesse caso, o modelo preditivo é criado em dois estágios. O primeiro considera o método *BR* tradicional. O segundo utiliza a saída do primeiro estágio como atributos descritores. Desse modo, os classificadores binários do segundo estágio consideram a dependência incondicional substituindo as previsões originais fornecidas pelo *BR* por novas previsões. Originalmente, o classificador da segunda etapa considera os valores de previsões dos rótulos da primeira etapa sem considerar os atributos descritores originais, por isso a dependência explorada nesse caso é incondicional. Essa transformação da previsão inicial também pode ser interpretada como um procedimento de regularização (Dembczyński et al., 2012).

A seguir são descritas as propostas deste trabalho relacionadas à identificação da dependência de rótulos nos conjuntos de dados, bem como à construção de classificadores para explorar essa característica.

3.2. *UBC*: Identificando dependência de rótulos

Diversos algoritmos foram propostos para explorar a dependência de rótulos nos conjuntos de dados multirrótulo. Em geral, a avaliação experimental é realizada comparando o algoritmo proposto com o método *BR* e um ou mais algoritmos que também consideram a dependência de rótulos no processo de aprendizado. Além disso, diversos conjuntos de dados são normalmente utilizados com objetivo de reportar melhoras na exploração

da dependência de rótulos em diferentes domínios.

No entanto, não é claro o quanto um problema multirrótulo pode se beneficiar de algoritmos que considerem a dependência de rótulos. Desse modo, apesar da comparação empírica entre diversos algoritmos competidores, não é possível identificar quão perto esses algoritmos estão de uma exploração da dependência de rótulos considerada ideal. A identificação, pelo menos estimada, de um limite máximo de exploração da dependência de rótulo pode indicar que ainda existe margem para melhoria de algoritmos em determinados conjuntos de dados. Desse modo, identificar (estimar) os limites da consideração da dependência de rótulos se torna um aspecto importante para a evolução dos algoritmos de aprendizado multirrótulo.

Como mencionado anteriormente, existem propostas na literatura para identificar o grau de dependência condicional e incondicional entre os rótulos. Entretanto, não é de nosso conhecimento a existência de alguma proposta que se concentre em estimar os limites da exploração dessa característica pelos algoritmos de aprendizado.

Neste trabalho propomos o algoritmo chamado *Upper Bound Classifier* – *UBC* – que, em conjunto com o método *BR*, é utilizado para estimar dependências dos tipos condicional e incondicional em conjuntos de dados. Os padrões de dependência de rótulos encontrados podem ser visualizados em forma de modelos simbólicos ou de influência sobre qualquer medida de avaliação de aprendizado multirrótulo. A seguir, o algoritmo *UBC* é descrito em detalhes.

3.2.1 Algoritmo *UBC* – *Upper Bound Classifier*

A idéia básica do algoritmo *UBC* é delegar ao algoritmo de aprendizado a responsabilidade de identificar a dependência de rótulos contida no problema multirrótulo. Para tanto, o algoritmo segue o paradigma de transformação de problema análogo ao *BR*: cria um problema de classificação binária para cada rótulo do conjunto de dados. É importante ressaltar que o paradigma de transformação de problema fornece maior flexibilidade, em relação à adaptação de algoritmo, pois possibilita a utilização de uma grande gama de classificadores monorrótulo binário propostos na literatura.

A diferença do *UBC* para o *BR* está nos atributos descritores utilizados no treinamento de cada modelo binário. No caso do *UBC*, para cada rótulo, os atributos descritores são incrementados com os atributos relacionados a todos os outros rótulos. Desse modo, é fornecida para cada algoritmo binário toda a informação necessária para identificar, por meio de seu *bias*, padrões de dependência de rótulo contidos nos dados.

Na Figura 3.1 é ilustrado esse processo de treinamento do *UBC* conside-

rando um problema com $L = \{y_1, y_2, y_3\}$.

Fase de treinamento:

Figura 3.1: Ilustração dos problemas binários utilizados no processo de treinamento do algoritmo *UBC*.

O algoritmo *UBC* possui a capacidade de identificar padrões de dependência condicional e também incondicional. Para exemplificar, considere um algoritmo monorrótulo que segue o paradigma de regras de produção. Quando a regra encontrada não envolver os atributos descritores originais, mas apenas rótulos, a dependência encontrada é do tipo incondicional. Caso contrário, quando os rótulos e também os atributos descritores compõem a regra, o padrão encontrado se refere ao tipo condicional de dependência. De maneira análoga, algoritmos de aprendizado de outros paradigmas podem ser utilizados para encontrar padrões de dependência conforme seus respectivos *bias*. Desse modo, a hipótese por trás do *UBC* é a mesma que rege os algoritmos de aprendizado supervisionado: é possível identificar padrões de interesse contidos nos dados, de dependência de rótulos neste caso, por meio dos métodos de inferência (aprendizado).

A utilização de métodos de aprendizado supervisionado do paradigma simbólico no *UBC* já representaria o início do entendimento dos padrões de dependência no conjunto de dados, pois esse tipo de algoritmo possibilita a interpretação dos modelos criados e padrões encontrados (Rezende, 2003).

No entanto, o *UBC* pode ser utilizado em experimentos para estimar o ganho máximo, medido por métricas de avaliação multirrótulo, a ser obtido quando considerados os padrões de dependência de rótulos. Para tanto, o algoritmo *UBC* deve ser avaliado como um classificador multirrótulo padrão. A diferença para os outros classificadores é que para a predição de cada rótulo binário de um exemplo, o *UBC* necessita o valor verdadeiro (correto) dos outros rótulos, pois esse tipo de informação é fornecido a ele no momento do treinamento. Obviamente, em aplicações reais, esse tipo de informação não estará disponível, pois é exatamente o que se quer prever. Porém, é possível realizar uma simulação com o conjunto de teste disponí-

vel, fornecendo a situação desejada pelo *UBC*.

Assim, para o classificador binário H_{y_j} realizar a predição da relevância do rótulo y_j para o exemplo E_i , são necessários, além dos atributos descritores, os valores corretos do conjunto de rótulos $L - \{y_j\}$ do exemplo E_i . Desse modo, toda a informação necessária para simular um cenário ideal de exploração de dependência de rótulos é fornecida ao algoritmo.

Qualquer medida de avaliação multirrótulo pode ser utilizada para quantificar a eficácia do classificador ideal simulado. Além disso, uma comparação com o método *BR*, que considera a independência dos rótulos, pode ser realizada para estimar a margem disponível no conjunto de dados para explorar a dependência de rótulo. Espera-se que a eficácia do *UBC* seja melhor que a eficácia do *BR* para dados de conjunto de treinamento representativos do problema. Quanto maior for a diferença entre a eficácia do *UBC* e do *BR* maior será a influência da dependência de rótulos identificada no conjunto de dados.

A complexidade computacional do *UBC* é semelhante à complexidade do *BR*. A diferença é a adição de $|L| - 1$ atributos descritores para cada problema binário criado. Portanto, a complexidade do algoritmo *UBC* é $O(|L| \times f_b(N, M + |L| - 1))$, onde $f_b(\bullet, \bullet)$ é a complexidade do algoritmo base de aprendizado monorrótulo binário.

A seguir, é apresentada a avaliação experimental do algoritmo *UBC* utilizando diversos conjuntos de dados *benchmark* disponíveis.

3.2.2 Avaliação experimental do *UBC*

Os experimentos realizados neste trabalho utilizando o algoritmo *UBC* têm como objetivo identificar o grau de dependência de rótulos em conjuntos de dados tradicionalmente utilizados pela comunidade, descritos anteriormente na Seção 2.6, página 41. Todos os conjuntos de dados foram divididos em partições de treino e teste utilizando o método de validação cruzada *5x2-fold cross-validation* com *folds* pareados, ou seja, a todos os algoritmos foram apresentados as mesmas partições.

Os experimentos foram realizados utilizando o método *BR*, disponível no framework *Mulan*, e o método *UBC*, o qual foi implementado utilizando esse mesmo *framework*¹. Ambos necessitam de algoritmos base de aprendizado monorrótulo, pois seguem o paradigma de transformação do problema. Para tanto, três algoritmos frequentemente utilizado pela comunidade e implementados no *framework Weka* foram considerados: i) *Support Vector Ma-*

¹O *UBC* encontra-se disponível em <http://www.labc.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/ubc.zip>

chines (SMO^2); ii) Árvore de Decisão (*J48*); e iii) Naïve Bayes (*NB*). Todos os algoritmos foram executados utilizando a configuração padrão do *Weka*.

3.2.3 Resultados e discussão

Na Tabela 3.1 são apresentados os resultados da aplicação do *BR* e *UBC* em doze conjuntos de dados, descritos na Tabela 2.10, página 41. Os valores nessa tabela se referem à medida de avaliação baseada em exemplos *F-Measure*, frequentemente utilizada na literatura.

Tabela 3.1: Resultados para *BR* e *UBC* utilizando doze conjuntos de dados. Os valores se referem à medida de avaliação *F-Measure* baseada em exemplos. A coluna *Dif* indica a diferença existente entre *UBC* e *BR*.

ID	Nome	<i>SMO</i>			<i>J48</i>			<i>NB</i>		
		<i>BR</i>	<i>UBC</i>	<i>Dif</i>	<i>BR</i>	<i>UBC</i>	<i>Dif</i>	<i>BR</i>	<i>UBC</i>	<i>Dif</i>
#1	bibtex	0.381	0.413	0.03	0.351	0.423	0.07	0.276	0.295	0.02
#2	cal500	0.334	0.575	0.24	0.344	0.584	0.24	0.337	0.598	0.26
#3	corel16k	0.049	0.421	0.37	0.084	0.372	0.29	0.204	0.399	0.20
#4	corel5k	0.134	0.435	0.30	0.077	0.405	0.33	0.184	0.378	0.19
#5	emotions	0.583	0.711	0.13	0.528	0.705	0.18	0.623	0.696	0.07
#6	enron	0.503	0.606	0.10	0.489	0.626	0.14	0.333	0.348	0.01
#8	medical	0.754	0.832	0.08	0.762	0.830	0.07	0.264	0.466	0.20
#9	ohsumed	0.426	0.453	0.03	0.435	0.606	0.17	0.400	0.526	0.13
#10	scene	0.609	0.846	0.24	0.555	0.847	0.29	0.562	0.689	0.13
#11	slashdot	0.465	0.834	0.37	0.322	0.782	0.46	0.426	0.608	0.18
#12	tmc2007	0.688	0.721	0.03	0.649	0.700	0.05	0.607	0.610	0.00
#13	yeast	0.608	0.936	0.33	0.547	0.934	0.39	0.541	0.788	0.25

Como mencionado anteriormente, é esperado que o classificador multirrótulo *UBC* apresente resultados iguais ou superiores aos apresentados pelo classificador multirrótulo *BR*. Porém, exceções podem ocorrer por se tratar de métodos que fazem inferência/generalização sobre os dados, que por sua vez podem não ser representativos.

No entanto, como é possível observar na Tabela 3.1, considerando os valores de *F-Measure*, o algoritmo *UBC* gerou classificadores sempre semelhantes ou superiores aos gerados pelo *BR*, já que todos os valores de *Dif* são não-negativos. Na Figura 3.2 é apresentada uma análise visual desses resultados.

Os conjuntos de dados foram agrupados em três categorias conforme a diferença apresentada entre *UBC* e *BR*, ou seja, de acordo com o grau de dependência de rótulos em cada conjunto de dados.

Categoria 1 : $Dif < 0,1$ – Nula ou baixa influência da dependência de rótulos.

Categoria 2 : $0,1 \leq Dif < 0,2$ – Intermediária influência da dependência de rótulos.

²Implementação do *Weka* para *Support Vector Machines*.

Categoria 3 : $Dif \geq 0,2$ – Alta influência da dependência de rótulos.

Na Tabela 3.2, cada conjunto de dados é associado a sua correspondente categoria de acordo com o grau de dependência de rótulos identificado por cada algoritmo base.

ID	Nome	SMO	J48	NB
#2	<i>cal500</i>	•••	•••	•••
#3	<i>corel16k</i>	•••	•••	•••
#13	<i>yeast</i>	•••	•••	•••
#4	<i>corel5k</i>	•••	•••	••
#10	<i>scene</i>	•••	•••	••
#11	<i>slashdot</i>	•••	•••	••
#5	<i>emotions</i>	••	••	•
#6	<i>enron</i>	••	••	•
#9	<i>ohsumed</i>	•	••	••
#8	<i>medical</i>	•	•	•••
#1	<i>bibtex</i>	•	•	•
#12	<i>tmc2007</i>	•	•	•

Tabela 3.2: Categoria de influência da dependência de rótulos para cada conjunto de dados e algoritmo base. Os conjuntos de dados estão ordenados pelas categorias a eles associados. Categoria 1: •; Categoria 2: ••; Categoria 3: •••.

Na primeira categoria podem ser incluídos nitidamente os conjuntos *bibtex* e *tmc2007*, para os quais todos os classificadores criados com os três algoritmos apresentaram $Dif < 0,1$. O conjunto *medical* poderia também ser incluído nessa categoria, pois apesar da grande melhora apresentada pelo *UBC* ($Dif = 0,20$) utilizando *NB* como algoritmo base, os resultados ainda foram bem abaixo dos resultados de *BR* utilizando os outros algoritmos base. Desse modo, pode-se considerar que esses três conjuntos de dados não apresentam influência da dependência de rótulos que pode ser explorada pela visão dos algoritmos base utilizados. Em outras palavras, não há margem para melhoria da eficácia de predição nesses conjuntos por meio da exploração da dependência de rótulos.

Os resultados com os conjuntos *ohsumed*, *enron* e *emotions* se encaixam na Categoria 1 apenas para um algoritmo base cada. Esses conjuntos ficam mais bem classificados na Categoria 2, pois apresentam resultados que encaixam nessa categoria para 2/3 dos algoritmos base.

Os conjuntos *slashdot*, *scene* e *corel5k* apresentam resultados de segunda categoria de influência da dependência de rótulos para um algoritmo base. No entanto, para os outros algoritmos existe claramente margem significativa para explorar a dependência de rótulos nesses conjuntos. Todos os outros conjuntos de dados apresentaram margem de Categoria 3 para exploração da dependência de rótulo. Desse modo, pelo menos 75% dos conjuntos utilizados apresentam intermediária ou alta influência da dependência de rótulos, o que significa que classificadores multirrótulo que

consideram essa característica podem usufruir de melhor eficácia preditiva nesses conjuntos.

Na Figura 3.2, a margem para a exploração da dependência de rótulos fica mais evidente. Nessa figura, o círculo inferior e a flecha superior representam respectivamente o resultado de *F-Measure* obtido pelos classificadores *BR* e *UBC* para cada algoritmo base considerado.

Figura 3.2: Diferença entre *BR* e *UBC* para cada conjunto de dados e cada algoritmo base considerando *F-Measure* como medida de avaliação. O círculo inferior e a flecha superior representam respectivamente os resultados para *BR* e *UBC*.

Para realizar uma análise global, incluindo outras medidas de avaliação, foi gerado um gráfico do tipo radar para cada conjunto de dados e cada algoritmo base. Cada gráfico radar apresenta os valores da eficácia de *BR* e *UBC* para cinco medidas de avaliação: *Micro-F1*, *Macro-F1*, *F-Measure*, *Subset-Accuracy* e *Accuracy*. Nas Figuras 3.3, 3.4 e 3.5 são apresentados os gráficos radar para cada algoritmo base considerado nos experimentos.

O comportamento apresentado nos gráficos radar é análogo ao observado considerando apenas a medida *F-Measure*. Seis conjuntos de dados (*cal500*, *corel16k*, *corel5k*, *scene*, *slashdot* e *yeast*) apresentam alta influência da dependência de rótulos, três conjuntos (*emotions*, *enron* e *ohsumed*) apresentam intermediária influência, enquanto que os outros três (*bibtex*, *medical* e *tmc2007*) apresentam influência quase nula dessa característica.

O *UBC* pode incorrer em erros de generalização caso seja construído utilizando amostras de dados que não são representativas. Esse é um problema que pode ser encontrado por qualquer algoritmo de aprendizado. Mesmo assim, o *UBC* fornece boas evidências da presença ou ausência de dependência de rótulos nos problemas multirrótulos. Além disso, as evidências

Figura 3.3: Gráficos radar apresentando valores de cinco medidas de avaliação para *BR* e *UBC* utilizando *SMO* como algoritmo base.

encontradas não são apresentadas como correlações entre pares de rótulo, por exemplo, mas na forma do impacto nas medidas de avaliação. Essa evidência torna o usuário do algoritmo (pesquisador) mais consciente do benefício em termos objetivos (medida de avaliação) que o problema multirótulo pode usufruir se for considerada a dependência de rótulos de maneira eficaz.

Por meio dos experimentos realizados neste trabalho foram identificadas evidências claras de que os conjuntos *bibtex*, *medical* e *tmc2007*, por exemplo, são pouco influenciados caso a dependência de rótulos for considerada. O pesquisador, consciente dessa característica do conjunto de dados, pode focar os esforços (cognitivos e computacionais) de melhoria da eficácia preditiva desses problemas em métodos alternativos aos que exploram a dependência de rótulos. O pesquisador poderia, por exemplo, focar na utilização de comitê de classificadores. Por outro lado, diversos conjuntos de dados apresentam grande influência da dependência de rótulos e, portanto, esforços para uma exploração eficaz dessa característica devem ter melhor custo-benefício.

Pode-se argumentar que o esforço computacional realizado *apenas* para

Figura 3.4: Gráficos radar apresentando valores de cinco medidas de avaliação para *BR* e *UBC* utilizando *J48* como algoritmo base.

tomar consciência da dependência de rótulos não se justifica, já que esse esforço poderia estar sendo diretamente utilizado para treinar e ajustar modelos preditivos reais. No entanto, além dos benefícios comentados anteriormente, o modelo preditivo gerado pelo *UBC* pode ser utilizado para explorar a dependência de rótulo na prática. Para tanto, o algoritmo *BR+*, proposto neste trabalho e descrito a seguir, deve ser utilizado.

Figura 3.5: Gráficos radar apresentando valores de cinco medidas de avaliação para *BR* e *UBC* utilizando *NB* como algoritmo base.

3.3. *BR+*: Explorando dependência de rótulos

O método *BR+*, implementado no *framework* Mulan³, pode ser considerado uma extensão do método *BR* com o objetivo de considerar a relação de dependência de rótulos na construção do modelo de classificação. A ideia principal do método é encaminhar ao algoritmo de aprendizado a responsabilidade de também identificar as dependências de rótulos do conjunto de dados. Para isso, o *BR+* também constrói $|L|$ problemas de classificação binária, cada qual com o conjunto de treinamento com atributos descritores expandidos com as informações de rótulos, tal como é realizado pelo *UBC*. O problema binário correspondente à predição do rótulo y_j , além de conter o conjunto padrão de atributos descritores X também é expandido com $|L| - 1$ atributos x_{y_k} com $k = 1..q$ e $y_k \neq y_j$. Em outras palavras, cada exemplo do conjunto de treinamento contém também atributos descritores que representam todos os rótulos, exceto o próprio rótulo a ser predito. Na

³O *BR+* encontra-se disponível em <http://www.labc.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/brplus.zip>

Figura 3.6 é ilustrado esse processo de treinamento.

A classificação de um novo exemplo é realizada de maneira análoga ao método *UBC*. A diferença é que, no método *BR+*, os classificadores construídos não podem contar com os valores dos descritores relacionados aos rótulos, pois esses valores não são conhecidos para novos exemplos, o que causa uma dependência cíclica. Para quebrar essa dependência, o método *BR+* realiza uma estimativa de um conjunto inicial $\Omega = \{y'_1, y'_2, \dots, y'_q\}$ dos valores dos rótulos por meio de outro classificador multirrótulo H_0 qualquer. Esses valores estimados, exceto o correspondente ao rótulo a ser predito, são incorporados ao novo exemplo como valores dos atributos descritores e, assim, é realizada a predição do rótulo correspondente ao classificador construído utilizando o respectivo conjunto de treinamento. Na Figura 3.6 é ilustrado esse processo de predição do *BR+* considerando um problema ilustrativo com $L = \{y_1, y_2, y_3\}$.

Figura 3.6: Ilustração do processo de treinamento e predição (sem atualização) do método *BR+*.

A abordagem de predição descrita na Figura 3.6 considera uma estratégia sem atualização dos valores inicialmente preditos, ou seja, os valores $\Omega = \{y'_1, y'_2, \dots, y'_q\}$ estimados inicialmente são utilizados por todos os classificadores binários do *BR+* sem realizar atualização conforme novas predições de cada classificador binário são obtidas. Essa abordagem de predição é chamada *Predição Sem Atualização (SA)*.

No entanto, o conjunto de valores Ω pode ser atualizado conforme novas predições são realizadas. Desse modo, na medida em que cada classificador binário H_{y_j} , com $j = 1..q$, é executado, o conjunto Ω é atualizado com o novo valor do rótulo obtido por esse classificador. Por exemplo, para a predição do rótulo y_1 , são adicionados ao exemplo de teste os valores $y'_2, \dots, y'_q$. Após classificar o exemplo utilizando o classificador H_{y_1} , a nova predição y''_1 do rótulo y_1 substitui a predição anterior y'_1 no conjunto Ω , i.e., $\Omega = \{y''_1, y'_2, \dots, y'_q\}$.

Caso o próximo rótulo a ser predito for y_2 , então os valores $y''_1, y'_3, \dots, y'_q$ são adicionados ao exemplo de teste e a predição y''_2 do classificador H_{y_2} substitui a anterior y'_2 no conjunto Ω . Esse processo é realizado para todos os rótulos em L obtendo, assim, a predição de todos os valores do exemplo de teste. Essa abordagem é chamada neste trabalho de *Predição Com Atualização (CA)*.

Como pode ser observado, a estimativa SA é independente da ordem em que os classificadores binários correspondentes a cada rótulo são executados para obter a predição final. Porém, isso não acontece na predição *CA*. Nesse caso, o conjunto Ω de rótulos estimados é atualizado a cada rótulo predito, o que torna importante a definição da ordem em que esses classificadores binários serão executados.

Neste trabalho foram propostas duas maneiras para definir a ordem de utilização dos classificadores e, conseqüentemente, a atualização dos valores do conjunto Ω :

1. Ordem pré-definida ou estática – *Est*;
2. Ordem dinâmica – *Din*.

No primeiro caso, a ordem é definida previamente, no processo de treinamento do modelo, baseada em algum critério. Essa ordem, uma vez definida, é utilizada para a predição de todos os novos exemplos. No segundo caso, a ordem é definida dinamicamente no momento do processo de predição de um novo exemplo, segundo algum critério.

Neste trabalho, dois critérios para definição da ordem de predição estática *Est* foram avaliados:

1. **Rótulo mais frequente primeiro – $F+$** : os $L = \{y_1, y_2, \dots, y_q\}$ são ordenados pela frequência decrescente que aparecem no conjunto de treinamento. Essa ordem é utilizada para a execução dos respectivos classificadores H_{y_j} .
2. **Rótulo menos frequente primeiro – $F-$** : a ordem definida por esse critério é a ordem inversa ao critério $F+$.

No caso da ordem dinâmica (*Din*), na predição de um novo exemplo, a ordem em que cada classificador binário H_{y_j} é executado está relacionada com a confiança com que os valores da estimativa inicial $\Omega = \{y'_1, y'_2, \dots, y'_q\}$ foram obtidos. Nesse caso, essas q estimativas iniciais são ordenadas considerando confiança de predição crescente. Essa ordem é utilizada para a execução dos respectivos classificadores H_{y_j} . Em outras palavras, o classificador H_{y_j} será o primeiro a ser executado caso o valor de confiança de y'_j seja

o menor do conjunto Ω , e será o primeiro valor a ser atualizado, com base nos outros valores y'_i , $i \neq j$, os quais apresentam maior confiança. Assim por diante, cada valor é atualizado, sendo o último valor a ser atualizado o que tiver a maior confiança.

Na Figura 3.7 é ilustrado o processo de atualização da predição considerando uma ordem pré-definida.

Figura 3.7: Ilustração do processo de predição do *BR+* considerando uma ordem pré-definida.

O *BR+* apresenta desvantagens relacionadas a tempo de execução, já que necessita de um estimador multirrótulo inicial para realizar as predições finais, duplicando o tempo de execução do *BR*. Essa desvantagem pode ser mitigada por meio da utilização de recursos de paralelização, o que, assim como no *BR*, é diretamente aplicável no *BR+*. Além disso, a qualidade das predições iniciais pode afetar a eficácia do *BR+*.

Uma das vantagens do método, como já mencionado, é o fato do *BR+* considerar a dependência de rótulos no processo de predição. A preocupação com essa característica é atribuída ao algoritmo monorrótulo de cada problema binário. Na teoria, o método é capaz de identificar tanto dependência condicional quanto incondicional. No entanto, na prática, os padrões de dependência incondicional são mais raros, pois são limitados a modelos que não consideram os atributos descritores, tal como o método *MS* realiza originalmente em sua segunda etapa. Além disso, diferentemente do *CC*, um único modelo preditivo é construído, pois não é necessário tomar decisões sobre a ordem em que os classificadores serão treinados. O *BR+* é influenciado pela ordem dos classificadores apenas na fase de predição e não no momento de treinamento dos modelos. Desse modo, diversas abordagens de predição podem ser avaliadas para um mesmo modelo do *BR+* sem que seja necessário reconstruir o mesmo.

Outra vantagem do método *BR+* é a possibilidade de utilizar o *UBC* para identificar tanto as características de dependência de rótulos do problema, como mencionado na seção anterior, quanto para identificar os limites do

próprio método *BR+*.

3.3.1 Avaliação experimental do *BR+*

Como já mencionado, diversos métodos desenvolvidos apresentam a preocupação com a característica de dependência de rótulos, tais como o *LP*, *RAKEL*, *PS*, *CC* e *MS*. Dentre eles, o método *CC* tem se destacado por apresentar os melhores resultados em termos de eficácia preditiva e complexidade computacional (Read et al., 2009, 2011). Os experimentos com o *BR+* foram realizados com o objetivo de compará-lo com os algoritmos *BR*, *LP*, tradicionalmente utilizados na literatura, bem com o método *CC*, considerado estado da arte na literatura. Todos os algoritmos se encaixam no grupo de transformação de problema e utilizam algoritmos base de aprendizado monorrótulo, o que os tornam mais simples e aplicáveis para uma maior gama de problemas. Assim como para os experimentos realizados com *UBC*, para todos os algoritmos de aprendizado multirrótulo foram utilizados *SMO*, *J48* e *NB* como algoritmos base.

O método *CC* foi executado utilizando os valores de parâmetros padrão propostos pelos autores do método (Read et al., 2009). Para o método *BR+*, experimentos iniciais foram realizados com oito conjuntos de dados avaliando as diferentes abordagens de predição. Os resultados desses experimentos foram publicados em (Cherman et al., 2012, 2010a,b). Em geral, como a abordagem com atualização das predições iniciais com ordem definida de maneira dinâmica (*Din*) foi a que apresentou os melhores resultados, ela foi utilizada como abordagem representante do *BR+* neste trabalho para realizar experimentos com os conjuntos de dados utilizados pelo *UBC*. Para a estimativa dos valores iniciais do *BR+*, as predições realizadas pelos métodos *BR*, *LP* e *CC* foram utilizadas. Desse modo, é possível verificar se o *BR+* é capaz de melhorar as predições dos outros métodos a ele fornecidas como estimativa inicial dos rótulos.

Os métodos foram avaliados por meio de validação cruzada *5x2-fold cross-validation* com partições pareadas, da mesma maneira realizada nos experimentos com *UBC*. A medida *F-Measure* foi escolhida para medir a eficácia de predição dos métodos, pois está entre as medidas mais utilizadas na literatura (Metz et al., 2013).

3.3.2 Resultados e discussão

Na Tabela 3.3, resultados considerando *SMO* como algoritmo base são apresentados. Os valores na tabela são apresentados no formato *med(rank)*, onde *med* é o valor absoluto médio sobre todas as partições da medida de

avaliação *F-Measure* e *rank* é a posição relativa do método em relação aos outros métodos para um conjunto de dados. A última coluna da tabela se refere ao resultado de *F-Measure* para o *baseline* $Geral_B$. Resultados não superiores aos apresentados pelo $Geral_B$ são destacados.

Tabela 3.3: Resultados utilizando SMO como algoritmo base. Os valores são apresentados no formato $med(rank)$, onde med é o valor da medida de avaliação *F-Measure* e *rank* é a posição relativa do método em relação aos outros métodos para um conjunto de dados. As células destacadas não apresentaram resultados superiores ao *baseline* $Geral_B$.

		BR	LP	CC	BR+ (BR)	BR+ (LP)	BR+ (CC)	$Geral_B$
#1	<i>bibtex</i>	0.381(2.0)	0.370(6.0)	0.377(4.0)	0.382(1.0)	0.374(5.0)	0.378(3.0)	0.099
#2	<i>cal500</i>	0.334(4.0)	0.338(3.0)	0.321(6.0)	0.351(1.0)	0.340(2.0)	0.333(5.0)	0.455
#3	<i>corel16k001</i>	0.049(6.0)	0.176(1.0)	0.158(4.0)	0.141(5.0)	0.173(2.0)	0.163(3.0)	0.164
#4	<i>corel5k</i>	0.134(6.0)	0.160(4.0)	0.145(5.0)	0.182(1.0)	0.164(3.0)	0.171(2.0)	0.184
#5	<i>emotions</i>	0.583(6.0)	0.650(1.0)	0.609(4.0)	0.622(3.0)	0.625(2.0)	0.592(5.0)	0.296
#6	<i>enron</i>	0.503(6.0)	0.509(3.0)	0.505(5.0)	0.514(1.0)	0.509(3.0)	0.509(3.0)	0.417
#8	<i>medical</i>	0.754(6.0)	0.773(4.0)	0.770(5.0)	0.787(3.0)	0.789(1.5)	0.789(1.5)	0.232
#9	<i>ohsumed-f</i>	0.426(6.0)	0.459(2.0)	0.453(3.0)	0.441(4.0)	0.462(1.0)	0.438(5.0)	0.190
#10	<i>scene</i>	0.609(6.0)	0.733(1.0)	0.692(5.0)	0.700(4.0)	0.723(2.0)	0.703(3.0)	0.204
#11	<i>slashdot-f</i>	0.465(6.0)	0.522(2.0)	0.498(4.0)	0.469(5.0)	0.533(1.0)	0.506(3.0)	0.149
#12	<i>tmc2007-500</i>	0.688(5.0)	0.719(1.0)	0.688(5.0)	0.692(3.0)	0.700(2.0)	0.688(5.0)	0.429
#13	<i>yeast</i>	0.608(4.0)	0.615(2.0)	0.587(6.0)	0.617(1.0)	0.614(3.0)	0.592(5.0)	0.550
Ranking médio		5.3	2.5	4.8	2.6	2.3	3.5	

O classificador obtido com *BR+* (*LP*) seguido por *LP* e *BR+* (*BR*) apresentaram os melhores *ranking* médios utilizando SMO como algoritmo-base. Os resultados apresentados por esses três algoritmos foram claramente superiores aos apresentados pelos classificadores obtidos com *BR* e *CC*. O método *BR+* (*CC*) apresentou resultados intermediários de *ranking* médio.

Ainda considerando o *ranking* médio, um aspecto positivo do *BR+* é que na grande maioria dos casos ele foi capaz de melhorar as previsões fornecidas como estimativa inicial.

Para analisar a existência de diferença estatisticamente significativa entre os resultados, o teste de Friedman, com a hipótese nula de que todos os resultados são equivalentes, foi utilizado. Quando a hipótese nula é rejeitada pelo teste de Friedman, com confiança de 95%, o pós-teste de Nemenyi é utilizado para detectar quais diferenças entre os métodos são significativas (Demšar, 2006). De acordo com esse teste, a eficácia de dois métodos é significativamente diferente sempre que seus correspondentes *rankings* médios diferem por pelo menos um determinado valor de diferença crítica (CD). Os resultados do pós-teste de Nemenyi podem ser representados graficamente (Demšar, 2006), como mostrado na Figura 3.8 para os resultados da Tabela 3.3.

Nessa figura, a linha principal no eixo *X* representa o *ranking* médio obtido por cada método. Deve ser observado que valores menores de *ranking* médio são melhores. As linhas abaixo do eixo *X* conectam métodos que não

apresentam diferença estatisticamente significativa com nível de confiança de 95%. A diferença crítica (CD) é mostrada acima do eixo X .

Figura 3.8: Ranking de Friedman com pós-teste de Nemenyi para *SMO* como algoritmo base.

Na Figura 3.8 é possível observar que, ainda que sem diferença estatisticamente significativa para alguns casos, o *BR+* apresentou predições de qualidade superior àquelas fornecidas a ele como estimativa inicial. Essa característica é muito clara principalmente se considerado o método *BR+(BR)*, o qual apresenta diferença estatisticamente significativa em relação ao *BR*. Além disso, o *BR+(LP)* e o próprio *LP* também apresentaram resultados superiores com diferença estatisticamente significativa em relação ao método *CC*, o que pode ser considerado muito bom resultado, dado que este é considerado um método estado da arte na literatura.

Na comparação dos resultados individuais de cada classificador com os resultados obtidos pelo $Geral_B$, foram identificados casos em que o *baseline* não foi superado para três conjuntos de dados: *cal500*, *corel16k001* e *corel5k*. Para esses três conjuntos de dados, o *BR+* foi capaz de melhorar os resultados fornecidos a ele como estimativa inicial para a maioria dos casos. Mesmo assim, apenas para o conjunto de dados *corel16k001* o método *BR+*, em conjunto com *LP*, foi superior ao *baseline*.

Outro aspecto importante é a comparação com os resultados obtidos pelo *UBC*. Como pode ser observado na Figura 3.2 da página 52, os conjuntos *bibtex* e *tmc2007* tiveram o menor grau de melhora da eficácia preditiva pelos métodos que consideram a dependência de rótulos. Esses dois conjuntos foram os que apresentaram a menor margem de exploração da dependência de rótulo, o que mostra que, para esses conjuntos de dados, esforços para melhorar a qualidade de predição não devem ser empregados necessariamente na exploração da dependência de rótulos, mas em outras estratégias. Por outro lado, os conjuntos *corel16k*, *corel5k* e *scene* usufruíram de uma melhora consistente na qualidade de predição quando considerada a dependência de rótulos. Embora as melhoras para *corel16k* e *corel5k* não tenham sido na maioria dos casos suficientes para superar o *baseline*, esses resultados estão alinhados com a boa margem apresentada entre o *UBC* e o *BR* para esses conjuntos de dados – Figura 3.2.

No entanto, exceções também podem ser observadas. Os conjuntos *yeast*

e *cal500*, por exemplo, apresentaram pouca melhora de predição quando considerada a dependência de rótulo, apesar de serem bastante influenciados por essa característica segundo o *UBC*. Entre outros aspectos, essa constatação serve como evidência para que os esforços para explorar a dependência de rótulos nesses conjuntos de dados sejam empregados de maneira distinta à utilizada pelos algoritmos considerados neste trabalho. Os outros conjuntos de dados apresentaram uma melhora intermediária quando considerados os algoritmos que exploram a dependência de rótulos, o que está de acordo com o esperado pela análise realizada com o algoritmo *UBC*.

Os resultados considerando *J48* como algoritmo base são apresentados na Tabela 3.4.

Tabela 3.4: Resultados utilizando *J48* como algoritmo base. Os valores são apresentados no formato *med(rank)*, onde *med* é o valor da medida de avaliação *F-Measure* e *rank* é a posição relativa do método em relação aos outros métodos para um conjunto de dados. As células destacadas não apresentaram resultados superiores ao *baseline Geral_B*.

		<i>BR</i>	<i>LP</i>	<i>CC</i>	<i>BR+ (BR)</i>	<i>BR+ (LP)</i>	<i>BR+ (CC)</i>	<i>Geral_B</i>
#1	<i>bibtex</i>	0.351(4.0)	0.292(6.0)	0.345(5.0)	0.387(1.0)	0.375(3.0)	0.385(2.0)	0.099
#2	<i>cal500</i>	0.344(5.0)	0.330(6.0)	0.357(2.0)	0.354(3.0)	0.346(4.0)	0.358(1.0)	0.455
#3	<i>corel16k001</i>	0.084(5.0)	0.115(4.0)	0.074(6.0)	0.131(1.0)	0.118(3.0)	0.130(2.0)	0.164
#4	<i>corel5k</i>	0.077(5.0)	0.102(4.0)	0.076(6.0)	0.122(1.0)	0.109(3.0)	0.119(2.0)	0.184
#5	<i>emotions</i>	0.528(4.0)	0.522(5.0)	0.521(6.0)	0.556(1.0)	0.546(2.0)	0.539(3.0)	0.296
#6	<i>enron</i>	0.489(5.0)	0.420(6.0)	0.494(3.0)	0.507(1.0)	0.490(4.0)	0.506(2.0)	0.417
#8	<i>medical</i>	0.762(5.0)	0.751(6.0)	0.766(4.0)	0.795(2.0)	0.781(3.0)	0.798(1.0)	0.232
#9	<i>ohsumed-f</i>	0.435(5.0)	0.394(6.0)	0.439(4.0)	0.495(2.0)	0.491(3.0)	0.497(1.0)	0.190
#10	<i>scene</i>	0.555(6.0)	0.597(2.0)	0.604(1.0)	0.586(5.0)	0.594(3.0)	0.588(4.0)	0.204
#11	<i>slashdot-f</i>	0.322(6.0)	0.448(3.0)	0.412(4.0)	0.389(5.0)	0.457(2.0)	0.461(1.0)	0.149
#12	<i>tmc2007-500</i>	0.649(3.0)	0.602(6.0)	0.648(4.0)	0.656(1.0)	0.646(5.0)	0.654(2.0)	0.429
#13	<i>yeast</i>	0.547(2.0)	0.505(5.5)	0.518(4.0)	0.570(1.0)	0.505(5.5)	0.519(3.0)	0.550
Ranking médio		4.6	5.0	4.1	2.0	3.4	2.0	

Na comparação do *ranking* médio dos classificadores, a diferença entre o *BR+* e os outros métodos foi ainda mais acentuada utilizando *J48* como algoritmo base. O *ranking* médio dos classificadores criados pelas variações do *BR+* foi claramente superior ao *ranking* obtido pelos classificadores utilizados para as estimativas iniciais. Os métodos *BR+ (BR)* e *BR+ (CC)* apresentaram o melhor *ranking* médio, enquanto que *BR+ (LP)* obteve o terceiro melhor. O método *LP* apresentou os piores resultados, seguido do *BR* e do *CC*. Na Figura 3.9 é apresentado graficamente o *ranking* de Friedman com pós-teste de Nemenyi.

Os métodos *BR+ (BR)* e *BR+ (CC)* foram superiores com diferença estatisticamente significativa em relação aos classificadores utilizados por eles como estimativa inicial, respectivamente *BR* e *CC*. Esses resultados apontam grande evidência da capacidade do *BR+* em melhorar a qualidade de predição por meio da exploração da dependência de rótulos.

Figura 3.9: Ranking de Friedman com pós-teste de Nemenyi para *J48* como algoritmo base.

Os métodos de aprendizado multirrótulo avaliados não superaram o $Geral_B$ para quatro conjuntos de dados: *cal500*, *corel16k001*, *corel5k* e *yeast*. A exceção é o método *BR+* (*BR*), o qual foi o único a superar o *baseline* para o conjunto de dados *yeast*. Nos casos de *cal500* e *yeast* existe grande margem para melhoria da eficácia preditiva por meio da exploração da dependência de rótulos. Mesmo assim, os métodos que buscam explorar essa característica pouco conseguiram melhorar. A única melhora é observada para o método *BR+* (*BR*), o único a superar o *baseline* para esse conjunto. Para o conjunto de dados *corel16k001*, o qual também apresenta grande influência da dependência de rótulos, o método *BR+* foi capaz de melhorar a qualidade das predições, mas não o suficiente para superar o *baseline*.

Os resultados para todos os outros conjuntos de dados foram superiores ao *baseline*. Novamente, o conjunto de dados *tmc2007* foi pouco beneficiado pela consideração da dependência de rótulos, o que já era esperado pela análise realizado utilizando o *UBC*. No entanto, apesar da pequena influência da dependência de rótulos, o conjunto *bibtex* foi beneficiado pela consideração dessa característica, como pode ser observado pela diferença de 0,036 identificada entre o *BR* e o *BR+* (*BR*). Isso significa que o *BR+* (*BR*) foi capaz de explorar mais de 50% da margem identificada pelo *UBC* (0,07 – Tabela 3.1). Os outros conjuntos de dados também tiveram benefícios por considerar a dependência de rótulos. O maior destaque foi o conjunto de dados *ohsumed*, claramente o maior beneficiado pela dependência de rótulos. Para esse conjunto o *BR* apresentou 0,322 de *F-Measure*, enquanto que o *BR+* (*CC*) apresentou 0,461, uma diferença de 0,139. Isso representa mais de 80% de exploração da margem de 0,170 encontrada pelo *UBC*.

Na Tabela 3.5 são apresentados os resultados considerando *NB* como algoritmo base. O método *BR+* apresentou novamente as melhores posições de *ranking* médio. Especificamente, os melhores classificadores foram gerados pelo método *BR+* (*BR*), seguido do *BR+* (*LP*). Os métodos *BR+* (*CC*) e *LP* apresentaram resultados de *ranking* médio semelhantes. Os piores resultados foram obtidos pelos métodos *BR* e *CC*.

Na Figura 3.10 é apresentado o *ranking* de Friedman com pós-teste de Nemenyi. Apesar do teste de Friedman indicar diferença estatisticamente

Tabela 3.5: Resultados utilizando *NB* como algoritmo base. Os valores são apresentados no formato $med(rank)$, onde med é o valor da medida de avaliação *F-Measure* e $rank$ é a posição relativa do método em relação aos outros métodos para um conjunto de dados. As células destacadas não apresentaram resultados superiores ao *baseline* $Geral_B$.

		<i>BR</i>	<i>LP</i>	<i>CC</i>	<i>BR+ (BR)</i>	<i>BR+ (LP)</i>	<i>BR+ (CC)</i>	$Geral_B$
#1	<i>bibtex</i>	0.276(1.0)	0.216(6.0)	0.256(4.0)	0.257(3.0)	0.275(2.0)	0.247(5.0)	0.099
#2	<i>cal500</i>	0.337(1.0)	0.331(2.0)	0.319(3.0)	0.253(5.0)	0.258(4.0)	0.229(6.0)	0.455
#3	<i>corel16k001</i>	0.204(4.0)	0.155(6.0)	0.206(3.0)	0.220(1.0)	0.188(5.0)	0.213(2.0)	0.164
#4	<i>corel5k</i>	0.184(3.5)	0.107(6.0)	0.184(3.5)	0.207(2.0)	0.142(5.0)	0.210(1.0)	0.184
#5	<i>emotions</i>	0.623(5.0)	0.602(6.0)	0.628(4.0)	0.634(2.0)	0.639(1.0)	0.632(3.0)	0.296
#6	<i>enron</i>	0.333(6.0)	0.415(1.0)	0.339(5.0)	0.343(2.5)	0.343(2.5)	0.342(4.0)	0.417
#7	<i>medical</i>	0.264(6.0)	0.411(4.0)	0.270(5.0)	0.461(1.5)	0.456(3.0)	0.461(1.5)	0.232
#8	<i>ohsumed-f</i>	0.400(6.0)	0.439(4.0)	0.401(5.0)	0.502(2.0)	0.510(1.0)	0.501(3.0)	0.190
#10	<i>scene</i>	0.562(6.0)	0.632(4.0)	0.565(5.0)	0.686(1.0)	0.684(3.0)	0.685(2.0)	0.204
#11	<i>slashdot-f</i>	0.426(6.0)	0.526(2.0)	0.470(5.0)	0.526(2.0)	0.526(2.0)	0.523(4.0)	0.149
#12	<i>tmc2007-500</i>	0.607(2.0)	0.676(1.0)	0.595(3.0)	0.584(5.0)	0.594(4.0)	0.583(6.0)	0.429
#13	<i>yeast</i>	0.541(3.5)	0.567(2.0)	0.536(5.0)	0.541(3.5)	0.578(1.0)	0.534(6.0)	0.550
Ranking médio		4.2	3.7	4.2	2.5	2.8	3.6	

significativa entre os métodos, o pós-teste de Nemenyi não constatou essa diferença entre os pares comparados. No entanto, é possível observar que os métodos gerados pelo *BR+* foram superiores na posição absoluta do *ranking* aos métodos utilizados por ele como estimativa inicial.

Figura 3.10: Ranking de Friedman com pós-teste de Nemenyi para *NB* como algoritmo base.

Resultados não superiores ao $Geral_B$ foram identificados para cinco conjuntos de dados. Além de *cal500*, *corel16k001*, *corel5k* e *yeast*, o conjunto *enron* também está incluso nessa condição. Para *cal500* e *enron* todos os métodos não conseguiram superar os resultados do $Geral_B$. O método *LP* foi o único a não superar o *baseline* para o conjunto de dados *corel16k001*. Para *corel5k*, os métodos *BR+ (BR)* e *BR+ (CC)* foram os únicos a superar o $Geral_B$. No caso do conjunto de dados *yeast*, apenas os resultados de *LP* e *BR+ (LP)* foram superiores aos apresentados pelo *baseline*. Os outros conjuntos de dados tiveram resultados superiores ao *baseline* com todos os métodos avaliados.

À exceção de *bibtex* e *tmc2007*, todos os conjunto de dados foram beneficiados com a consideração da dependência de rótulos pelo *BR+*. O conjunto de dados *ohsumed* foi novamente destaque em relação a exploração da dependência de rótulos. O método *BR+ (LP)* obteve 0,510, enquanto que o *BR* apresentou 0,400: uma melhora de 0,110 de 0,126 possível identificado pelo *UBC*. Isso significa que o *BR+ (LP)* explorou cerca de 87% da dependência

de rótulos contida no conjunto de dados na visão do algoritmo base (*NB*). Para *slashdot*, três métodos também apresentaram melhora significativa na qualidade de predição ao considerar a dependência de rótulos: *LP*, *BR+* (*BR*) e *BR+* (*LP*).

Na Tabela 3.5, são apresentados os melhores resultados de cada método de aprendizado multirrótulo considerando todos os três algoritmos base utilizados neste trabalho.

Tabela 3.6: Melhores resultados de cada método de aprendizado multirrótulo. Os valores são apresentados no formato $med(rank)$, onde med é o valor da medida de avaliação *F-Measure* e $rank$ é a posição relativa do método em relação aos outros métodos para um conjunto de dados. As células destacadas não apresentaram resultados superiores ao *baseline Geral_B*.

		<i>BR</i>	<i>LP</i>	<i>CC</i>	<i>BR+</i> (<i>BR</i>)	<i>BR+</i> (<i>LP</i>)	<i>BR+</i> (<i>CC</i>)	<i>Geral_B</i>
#1	<i>bibtex</i>	0.381(3.0)	0.370(6.0)	0.377(4.0)	0.387(1.0)	0.375(5.0)	0.385(2.0)	0.099
#2	<i>cal500</i>	0.344(5.0)	0.338(6.0)	0.357(2.0)	0.354(3.0)	0.346(4.0)	0.358(1.0)	0.455
#3	<i>corel16k001</i>	0.204(4.0)	0.176(6.0)	0.206(3.0)	0.220(1.0)	0.188(5.0)	0.213(2.0)	0.164
#4	<i>corel5k</i>	0.184(3.5)	0.160(6.0)	0.184(3.5)	0.207(2.0)	0.164(5.0)	0.210(1.0)	0.184
#5	<i>emotions</i>	0.623(6.0)	0.650(1.0)	0.628(5.0)	0.634(3.0)	0.639(2.0)	0.632(4.0)	0.296
#6	<i>enron</i>	0.503(6.0)	0.509(3.0)	0.505(5.0)	0.514(1.0)	0.509(3.0)	0.509(3.0)	0.417
#8	<i>medical</i>	0.762(6.0)	0.773(4.0)	0.770(5.0)	0.795(2.0)	0.789(3.0)	0.798(1.0)	0.232
#9	<i>ohsumed-f</i>	0.435(6.0)	0.459(4.0)	0.439(5.0)	0.502(2.0)	0.510(1.0)	0.501(3.0)	0.190
#10	<i>scene</i>	0.609(6.0)	0.733(1.0)	0.692(5.0)	0.700(4.0)	0.723(2.0)	0.703(3.0)	0.204
#11	<i>slashdot-f</i>	0.465(6.0)	0.526(2.5)	0.498(5.0)	0.526(2.5)	0.533(1.0)	0.523(4.0)	0.149
#12	<i>tmc2007-500</i>	0.688(5.0)	0.719(1.0)	0.688(5.0)	0.692(3.0)	0.700(2.0)	0.688(5.0)	0.429
#13	<i>yeast</i>	0.608(4.0)	0.615(2.0)	0.587(6.0)	0.617(1.0)	0.614(3.0)	0.592(5.0)	0.550
Ranking médio		5.0	3.5	4.5	2.1	3.0	2.8	

As três primeiras posições no *ranking* médio foram obtidas novamente pelas variações de classificadores do método *BR+*. O método *LP* apresentou o quarto melhor *ranking* médio, enquanto que *CC* e *BR* foram os últimos colocados. Na Figura 3.11 é apresentado graficamente o *ranking* médio do teste estatístico de Friedman com pós-teste de Nemenyi.

Figura 3.11: Ranking de Friedman com pós-teste de Nemenyi considerando os melhores resultados de cada método.

A qualidade das melhores predições do método *BR+* (*BR*) foi superior à qualidade de predições dos métodos *CC* e *BR* com diferença estatisticamente significativa. Apesar de melhores em relação à posição no *ranking* médio, as variações do *BR+* não apresentaram diferença estatisticamente significativa em relação aos métodos avaliados. Os resultados ruins apresentados pelo *CC* podem ser explicados pela necessidade se estabelecer uma ordem prévia de encadeamento dos classificadores. Como essa ordem deve

ser estabelecida antes do treinamento do modelo preditivo, seria custoso computacionalmente avaliar as melhores ordens para cada conjunto de dados. Essa é uma importante desvantagem do CC em relação ao BR+, o qual não necessita de uma ordem pré-estabelecida no treinamento do modelo. A única ordem necessária para o BR+ é no momento da predição. Ela é necessária para atualizar os valores de predição inicial fornecidos ao método. Avaliar diferentes ordens de predição tem baixo custo computacional para a maioria dos algoritmos base, pois não é necessário treinar novamente o modelo preditivo.

Conforme esperado pelas características do método, os resultados do LP estão correlacionados ao tamanho do conjunto de dados, mais especificamente à quantidade de rótulos $|L|$ e à combinação de multirrótulos $\#Dist$ (ver Tabela 2.10). O método apresentou os piores resultados para os quatro maiores conjuntos em termos de número de rótulos (*bibtex*, *cal500*, *corel16k* e *corel5k*), resultados intermediários para conjuntos com intermediária quantidade de rótulos e combinação de multirrótulos (*enron*, *medical*, *ohsumed* e *slashdot*) e entre os melhores resultados para os conjuntos menores nessas características (*emotions*, *scene*, *tmc2007* e *yeast*). Além de problemas de eficácia, esse método não é facilmente escalável por meio de paralelização como são os métodos BR, CC e BR+.

Considerando os melhores resultados de cada método, apenas para o conjunto de dados *cal500* todos os métodos não superaram o *baseline*. No caso da *corel5k*, apenas os métodos BR+ (BR) e BR+ (CC) foram superiores ao *Geral_B*. Todos os outros resultados apresentados pelos métodos considerados foram superiores aos resultados do *baseline*. Esses resultados indicam forte evidência da capacidade do BR+ de melhorar a predição inicial a ele fornecida e, assim, explorar a dependência de rótulos mesmo em grandes conjuntos de dados.

Os conjuntos de dados *yeast* e *cal500* podem ser considerados casos especiais a serem estudados. Quanto à *yeast*, foi identificada uma das maiores margens para se explorar a dependência de rótulos, ou seja, é um conjunto de dados com forte influência dessas características. No entanto, os métodos que a exploram não foram capazes de melhorar a eficácia preditiva em relação ao BR, o método que considera independência entre rótulos. Uma hipótese para esse comportamento é a existência de uma dependência cíclica entre os rótulos que não é quebrada pelos atributos descritores. Desse modo, apesar da existência de determinados rótulos serem bons para determinar a existência de outros rótulos, e vice-versa, os atributos descritores não conseguem prever com qualidade mínima a existência de nenhum deles.

O conjunto de dados *cal500* apresenta uma situação semelhante à do conjunto *yeast*, porém com o agravante de nenhum método ser capaz de superar o *baseline* quando esse conjunto é considerado. Mais do que isso, existe uma diferença importante, de cerca de 0,10, entre o valor de *F-Measure* para o *Geral_B* e os outros métodos.

Apesar de serem negativos, esses dois casos mostram a importância de se considerar métodos *baseline*, bem como o *UBC*, quando se quer explorar a dependência de rótulos.

3.4. Considerações finais

Neste capítulo foram apresentadas e avaliadas empiricamente duas propostas deste trabalho: os métodos *UBC* e *BR+*. Essas propostas estão relacionadas com o entendimento e com a exploração da dependência de rótulos, uma característica intrínseca do problema multirrótulo.

O método *UBC* contribui para o entendimento da influência que a dependência de rótulos exerce sobre a eficácia de classificadores multirrótulo. Por meio da utilização desse método, é possível identificar conjuntos de dados que apresentam grande influência dessa característica e outros conjuntos que são menos ou quase nada influenciados. Diferentemente de outras abordagens que tentam medir a dependência de rótulos de conjuntos de dados, o *UBC* fornece esse indicativo em termos de medidas de avaliação, ou seja, identificar a dependência de rótulos que influencia diretamente na eficácia do classificador é o objetivo desse método. Essa tarefa é realizada por meio da utilização de um algoritmo base de aprendizado. Portanto, a identificação dos padrões de dependência de rótulos está condicionada à visão (*bias*) do algoritmo base utilizado e da qualidade dos dados fornecidos ao algoritmo.

O *UBC* pode ser utilizado para auxiliar no desenvolvimento de novos algoritmos ou no esforço de melhorar a eficácia preditiva de um domínio de aplicação específico. A utilização do *UBC* nos conjuntos de dados *yeast* e *cal500*, por exemplo, permitiu identificar que esses conjuntos apresentam uma das maiores influências da dependência de rótulos, porém nenhum dos algoritmos de aprendizado considerados conseguiu de fato explorar essa influência. O domínio de aplicação desses conjuntos de dados poderia ser investigado em maiores detalhes com intuito de desenvolver técnicas para explorar o tipo de dependência de rótulos presente nesses casos. Outro aspecto positivo e de utilização direta do *UBC* é que o modelo construído por esse método para o entendimento da dependência de rótulos pode também ser utilizado para explorar de fato essa característica, e, assim, obter melhor

eficácia preditiva para o domínio de aplicação. Para tanto, o método *BR+* deve ser considerado.

O método *BR+* foi aplicado a doze conjuntos de dados e, para diversos casos, apresentou resultados superiores a outros métodos tradicionais da literatura, como o método *CC*, considerado estado da arte em classificação multirrótulo. O fato do *BR+* necessitar de uma predição inicial poderia ser considerado um ponto negativo desse método. No entanto, foi observado que na grande maioria dos casos o *BR+* foi capaz de melhorar a qualidade da predição inicial a ele fornecida. Além disso, conjuntamente com o *UBC*, foi possível observar que quanto melhor a predição inicial fornecida melhor será a predição final realizada pelo *BR+*. Desse modo, o *BR+* tende sempre a se beneficiar da existência de novos algoritmos que forneçam melhores predições até o limite estabelecido pelo *UBC*. Quanto à escalabilidade do método, da mesma maneira que *CC* e *BR*, o *BR+*, bem como o *UBC*, podem se beneficiar facilmente de técnicas de paralelização para a construção de seu modelo preditivo e, assim, ser aplicado a grandes conjuntos de dados.

Aprendizado Ativo Multirrótulo

A condição básica para a aplicação de métodos de aprendizado supervisionado é contar com um conjunto de exemplos rotulados para treinamento. No entanto, para diversos problemas, a obtenção de rótulos é uma tarefa difícil para o rotulador, ou seja, é uma tarefa que consome tempo e/ou que é custosa financeiramente. Nesse sentido, métodos de aprendizado ativo têm sido propostos com o intuito de reduzir o esforço necessário para se obter um conjunto de dados rotulado, com o qual um algoritmo de aprendizado pode gerar um classificador com boa qualidade de predição. A ideia central desses métodos é permitir que o algoritmo de aprendizado escolha, dentro do conjunto de exemplos não rotulados, os exemplos mais informativos para serem rotulados. Esses exemplos são submetidos a um oráculo para rotulação (*e.g.* um anotador humano), o qual entende da natureza do problema. Desse modo, algoritmos de aprendizado ativo podem reduzir substancialmente a quantidade de exemplos rotulados necessários para se construir um bom classificador ([Settles, 2010](#)).

Considerando o cenário multirrótulo, o desenvolvimento e a aplicação de métodos de aprendizado ativo, bem como o próprio processo de rotulação manual dos dados, apresentam novas alternativas e novos desafios técnico-científicos. Apenas recentemente o cenário multirrótulo tem recebido atenção para o desenvolvimento de novas abordagens de aprendizado ativo. Três aspectos importantes de desafios relacionados a essa área podem ser destacados:

Minimizar o esforço do oráculo: *Quais são os custos de rotulação manual em diferentes tipos de dados multirrótulo? Quais abordagens são mais adequadas?* Independente da utilização do aprendizado ativo, as requisições feitas a um oráculo humano para rotular manualmente exemplos pode ser realizada em lotes. Diferentemente do contexto monorrótulo, a abordagem utilizada pelo rotulador para realizar sua tarefa tem influência no custo final de rotulação quando considerado o contexto multirrótulo.

Desenvolver novas abordagens de aprendizado ativo: *Quais abordagens de aprendizado ativo são adequadas para problemas de aprendizado multirrótulo?* Novas abordagens de aprendizado ativo multirrótulo podem ser desenvolvidas, e mesmo as abordagens do aprendizado ativo monorrótulo podem ser adaptadas para o contexto multirrótulo. No entanto, não há uma ideia consolidada de quais novas abordagens ou adaptações realizarão o aprendizado de maneira satisfatória no contexto multirrótulo.

Avaliar a eficácia do aprendizado ativo: *Como medir a eficácia dos métodos de aprendizado ativo no contexto multirrótulo?* Além da possibilidade de utilização de diversas medidas de avaliação, o aprendizado ativo multirrótulo apresenta novas variáveis no protocolo de experimentos para avaliar a eficácia das abordagens propostas, tais como a composição do conjunto de exemplos inicialmente rotulados.

Neste capítulo são apresentados primeiramente os principais conceitos de aprendizado ativo, bem como as abordagens já propostas pela comunidade no contexto multirrótulo. Posteriormente, são apresentadas as contribuições deste trabalho relacionadas aos desafios anteriormente citados.

4.1. Conceitos básicos

Aprendizado ativo é uma forma de aprendizado supervisionado no qual o próprio algoritmo de aprendizado tenta selecionar as amostras de exemplos (objetos) mais representativas para serem rotuladas. Desse modo, o algoritmo de aprendizado controla o processo de rotulação perguntando a um oráculo, tipicamente um especialista humano, a informação de rótulo das amostras selecionadas. Assim, diferentemente do modelo passivo de aprendizado supervisionado, no qual todos os rótulos são obtidos *apriori*, no aprendizado ativo o algoritmo de aprendizado escolhe interativamente quais são as amostras de dados a serem rotuladas.

Existem basicamente três cenários de aprendizado ativo ([Settles, 2010](#)):

membership query synthesis: Neste cenário, o algoritmo de aprendizado pode consultar qualquer exemplo não rotulado, bem como gerar sinteticamente exemplos que o algoritmo julgue necessário para o aprendizado.

stream-based: Este cenário considera dados que são recebidos de maneira sequencial. Neste caso, o algoritmo de aprendizado ativo deve decidir se cada exemplo individualmente deve ou não ser rotulado.

pool-based: No cenário *pool-based*, todo o conjunto de exemplos não rotulados D_U (*unlabeled pool*) é avaliado pelo algoritmo antes de selecionar um ou mais objetos para serem rotulados.

Na Figura 4.1 é mostrado o ciclo padrão de aprendizado ativo do cenário *pool-based*.

Figura 4.1: Ciclo padrão de aprendizado ativo *pool-based*. Adaptado de (Settles, 2010).

Este trabalho está focado no cenário *pool-based* devido a sua aplicabilidade a um grande número de problemas reais, tais como classificação de texto, classificação e recuperação de imagens, reconhecimento de fala e diagnóstico de câncer, entre outros.

4.2. Aprendizado ativo para dados multirrotulados

Como mencionado, existem alguns desafios que precisam ser considerados quando se pretende aplicar aprendizado ativo em dados multirrotulados. Nas próximas seções são descritos os desafios mais relevantes abordados neste trabalho.

4.2.1 Abordagens manuais de rotulação

Semelhante a um sistema de aprendizado ativo monorrótulo, um sistema de aprendizado ativo multirrótulo pode requisitar a rotulação de um ou mais objetos. Se a requisição é para apenas um objeto, o anotador irá observar (olhar, ler, ouvir, assistir) o objeto em uma tentativa de entendê-lo e caracterizá-lo como relevante ou não para cada um dos q possíveis rótulos $y_i \in \{y_1, y_2, \dots, y_q\}$ que participam dos multirrótulos.

No entanto, as requisições de rotulação podem também ser feitas para um lote (*batch*) de objetos. Por exemplo, a aquisição dos rótulos para a competição de anotação e recuperação de conceitos de fotos *ImageCLEF 2011*¹ foi realizada por meio do processo chamado *crowd-sourcing* utilizando lotes de 10 e 24 imagens (Nowak et al., 2011). Nesse caso, existem duas maneiras de proceder com a rotulação dos exemplos:

Object-wise ou Global. Nessa abordagem, o foco da rotulação é dada para cada objeto do lote a ser rotulado, *i.e.*, para cada objeto define-se a relevância (presença ou ausência) de cada um dos q rótulos do problema antes de processar (rotular) o próximo objeto. Essa abordagem também foi chamada de *Anotação Global* por Esuli e Sebastiani (2009).

Label-wise ou Local. Nessa abordagem, o foco da rotulação é dada para cada rótulo do problema. Desse modo, cada rótulo é processado pelo rotulador para determinar sua relevância em cada um dos n objetos do lote a ser rotulado. O foco de processamento é transferido para outro rótulo apenas após a avaliação completa de todos os exemplos em relação ao rótulo anterior. Essa abordagem também foi chamada de *Anotação Local* por Esuli e Sebastiani (2009).

O custo de utilização de cada abordagem depende principalmente do tipo do dado a ser rotulado. Considere a requisição para rotular n objetos com q possíveis rótulos. Seja c_o o custo médio de *entender um objeto*; c_l o custo médio de entender um rótulo; e c_{lo} o custo médio de decidir se um objeto deve ou não ser associado com um determinado rótulo. Se *desconsiderarmos os aspectos cognitivos e fisiológicos* do anotador que estão presente no processo de rotulação, tal como sua capacidade de memória de curto prazo, então, uma estimativa do custo total da rotulação do tipo *object-wise* seria:

$$n[c_o + q(c_l + c_{lo})] = nc_o + nqc_l + nqc_{lo}$$

De maneira análoga, a estimativa de custo no caso da rotulação *label-wise* seria:

¹<http://www.imageclef.org/2011>

$$q[c_l + n(c_o + c_{lo})] = qc_l + nqc_o + nqc_{lo}$$

Assumindo que o custo da rotulação *label-wise* é menor que o custo da rotulação *object-wise*, tem-se

$$qc_l + nqc_o + nqc_{lo} < nc_o + nqc_l + nqc_{lo}$$

$$qc_l + nqc_o < nc_o + nqc_l$$

$$n(q - 1)c_o < q(n - 1)c_l$$

$$c_o < \frac{q(n - 1)}{n(q - 1)}c_l \approx \frac{qn}{nq}c_l = c_l$$

Isso significa que a escolha da abordagem de rotulação tem forte dependência do custo de entendimento do objeto e do rótulo. Se o custo de entender o objeto (rótulo) é maior, então a abordagem *object (label) wise* deveria ser adotada.

Na Figura 4.2 é ilustrado² o custo de entender um rótulo, bem como o custo de entender objetos de diferentes tipos de dados. O custo de entender um objeto é, em geral, menor apenas para imagens, as quais o humano entende em uma fração de segundo, enquanto que vídeos são compostos por diversas imagens e opcionalmente também por áudio. O custo de entendimento de áudios e vídeos está fortemente ligado ao tamanho dos mesmos. O mesmo vale, em menor grau, para o entendimento de documentos. Desse modo, documentos, áudios e vídeos requerem um maior tempo de entendimento do que os conceitos típicos dos rótulos.

4.2.2 Requisições completas ou parciais de rótulos

Na tarefa clássica de aprendizado monorrótulo o sistema de aprendizado ativo requisita o valor da variável-alvo (classe) para um ou mais objetos. No contexto de aprendizado multirrótulo, o que o sistema pode requerer?

Normalmente, o sistema deveria requerer os valores de todas as variáveis-alvos binárias (rótulos), que compõem o problema multirrótulo, para um ou mais objetos. Nesse caso, um algoritmo de aprendizado multirrótulo incremental poderia atualizar o modelo atual baseado nos novos exemplos rotulados. Uma abordagem diferente dessa foi adotada por Qi et al. (2009).

²Por se tratar de uma figura meramente ilustrativa e com fins didáticos, desconsidere a magnitude das barras de custo. O interesse nessa figura é ilustrar graficamente quando é vantajoso utilizar determinada abordagem de rotulação manual.

Figura 4.2: Custo geral de entender objetos de diferentes tipos de dados e custo de entender um rótulo.

Eles propuseram um sistema que requisita os valores para apenas um subconjunto de rótulos e, posteriormente, infere os valores restantes baseado na correlação de rótulos.

Contar com valores de apenas um subconjunto de rótulos requer um algoritmo que é incremental em termos de exemplos de treinamento parcialmente rotulados. O algoritmo *Binary Relevance* é talvez o único algoritmo que satisfaz plenamente esse requisito, pois trata cada rótulo de maneira independente, além de ser um padrão que é frequentemente utilizado como referência.

Portanto, o desenvolvimento de estratégias que requisitem rotulação parcial dos objetos poderia ser um esforço válido, como estratégias que explorem a dependência de rótulos e ainda assim continuem satisfazendo o requisito de rotulação parcial. No entanto, o esforço de rotulação deve ser considerado, pois se o sistema requisita o rótulo do mesmo objeto em dois diferentes momentos, o custo de entendimento do objeto também incorre duas vezes. Como discutido na Seção 4.2.1, essa seria uma estratégia ineficiente para a maioria dos tipos de dados.

Baseado nessa análise, neste trabalho é utilizada a abordagem de requisição completa dos valores de rótulos para um dado objeto, a qual é adequada para a grande maioria dos problemas.

4.2.3 Avaliação de exemplos não rotulados

O aspecto chave dos algoritmos de aprendizado ativo monorrótulo é a maneira como eles avaliam, por meio de uma função de avaliação, o quão informativo os exemplos não rotulados são. Como existe somente uma variável-alvo no contexto monorrótulo, apenas uma função de avaliação dos exemplos para essa variável-alvo é suficiente para o aprendizado ativo. No

Algoritmo 3 é descrito o procedimento padrão composto pela função de avaliação (*query*), a qual calcula o valor de evidência para cada exemplo $E_i \in D_u$ e retorna os t exemplos com melhor evidência quanto à informação que carregam. Esses exemplos são rotulados pelo oráculo para compor o conjunto D_l de exemplos rotulados.

```

input :  $D_l$ : conjunto de exemplos rotulados;
         $L$ : conjunto de rótulos;
         $D_u$ : conjunto de exemplos não rotulados;
         $H$ : algoritmo de aprendizado multirrótulo;
         $f$ : classificador multirrótulo gerado por  $H$ ;
         $oracle$ : o rotulador;
         $r$ : número de iterações;
         $t$ : número de exemplos a serem rotulados a cada iteração.

for 1, 2, ...,  $r$  do
     $f \leftarrow H(D_l)$ 
     $\{E_i\}_{i=1}^t \leftarrow query(f, L, D_u, t)$ 
     $\{Y_i\}_{i=1}^t \leftarrow oracle(\{E_i\}_{i=1}^t)$ 
     $D_l \leftarrow D_l \cup \{(E_i, Y_i)\}_{i=1}^t$ 
     $D_u \leftarrow D_u - \{E_i\}_{i=1}^t$ 
end

```

Algoritmo 3: Procedimento padrão para o aprendizado ativo

No entanto, no contexto multirrótulo, cada variável-alvo (rótulo) deve ser avaliada, o que implica na geração de diversos valores de evidência que devem ser agregados para formar um único valor de estimativa da evidência para um dado objeto. Desse modo, em dados multirrótulo, a função de avaliação dos algoritmos de aprendizado ativo (*query*) é composta pelas duas seguintes funções:

Função de avaliação. Uma função $evaluate(E_i, y_j)$ para avaliar pares objeto-rótulo (E_i, y_j) e estimar o quão informativo o objeto E_i seria, caso rotulado, para diferenciar o rótulo y_j dos outros rótulos do problema. Essa função deve retornar um valor de evidência $e_{i,j}$ para todo objeto não rotulado $E_i \in D_u$ do conjunto de consulta e para todo rótulo $y_j \in L = \{y_1, y_2, \dots, y_q\}$ do problema.

Função de agregação. Uma função $aggregate(e_{i,1}, e_{i,2}, \dots, e_{i,q})$ para agregar os valores de evidência de um dado objeto E_i obtidos em cada par objeto-rótulo da função anterior. Desse modo, um único valor de evidência e_i é atribuído para o objeto não rotulado E_i .

No Algoritmo 4 é apresentada a função de avaliação (*query*), descrita no Algoritmo 3, a qual é composta pelas funções *evaluate* e *aggregate* utilizadas no contexto de aprendizado ativo multirrótulo. Após a avaliação da

evidência de cada objeto $E_i \in D_u$, os melhores t objetos são retornados pela função *best*.

```

input :  $D_u$ : conjunto de exemplos não rotulados;
         $L$ : conjunto de rótulos;
         $f$ : classificador multirrótulo;
         $t$ : número de exemplos a serem retornados para rotulação.
output: Os  $t$  exemplos não rotulados com melhores evidências

foreach  $E_i \in D_u$  do
  foreach  $y_j \in L$  do
     $e_{i,j} \leftarrow \text{evaluate}(D_u, f, E_i, y_j)$ 
  end
   $e_i \leftarrow \text{aggregate}(e_{i,1}, e_{i,2}, \dots, e_{i,q})$ 
end
return  $\text{best}((e_1, e_2, \dots, e_n), t, D_u)$ 

```

Algoritmo 4: Avaliação de exemplos não rotulados pela função de avaliação (*query*).

Dado o *framework* padrão de aprendizado ativo multirrótulo apresentado anteriormente, as soluções propostas na literatura e também neste trabalho se concentram em definir as abordagens utilizadas para as funções *evaluate* e *aggregate*. A seguir, as propostas da literatura, bem como as propostas deste trabalho, são apresentadas para cada uma dessas funções.

Propostas para a função de avaliação EVALUATE

Amostragem por incerteza. (conf): (Brinker, 2006; Esuli e Sebastiani, 2009; Singh et al., 2010).

A incerteza (ou confiança) de predição de um objeto é calculada baseada no *score* bruto retornado pelo classificador base. A natureza desse valor depende do *bias* do algoritmo de aprendizado, o qual pode ser baseado em distância da margem de decisão (*Support Vector Machines*), baseado em probabilidades (*Naïve Bayes*), entre outros. No caso da abordagem *Binary Relevance*, utilizada neste trabalho, um classificador base binário para cada rótulo é construído. Cada um desses classificadores fornece um valor *score* de predição para um dado objeto a ser predito. Quanto mais próximo do limiar de corte for esse valor retornado pelo classificador, mais incerto (menos confiante) o classificador está em relação à predição do rótulo para o objeto. Tipicamente, o valor de limiar é configurado em 0,5 em uma escala de 0 a 1,0. Nessa estratégia, os objetos com maiores incertezas na predição (mais próximos do limiar de corte) são os escolhidos pelo aprendizado ativo para rotulação pelo oráculo. A ideia intuitiva é que exemplos com predições

mais incertas têm maior potencial de melhorar o classificador caso rotulados pelo oráculo e utilizados pelo algoritmo de aprendizado para retreinar o modelo.

A desvantagem deste método é a escolha do limiar adequado, pois o valor padrão 0,5 é adequado somente no caso de *scores* calibrados, ou seja, nem sempre esse valor é adequado para todos os classificadores. Mesmo assim, essa é a estratégia mais utilizada como referência de aprendizado ativo pela comunidade, com valor 0,5 como limiar, e também é utilizada neste trabalho como base de comparação com outros métodos.

Na Tabela 4.1 é aplicado esse método a um exemplo ilustrativo que considera cinco objetos (E_1, E_2, E_3, E_4 e E_5) e três rótulos ($L = \{y_1, y_2, y_3\}$). Os valores de *score* bruto de predição ($f(y_j)$) para cada objeto E_i e cada rótulo y_j do problema são apresentados no lado esquerdo da tabela e processados conforme a estratégia da função de avaliação *conf*, chamada $evaluate_{conf}$. Os valores processados pela estratégia de avaliação são apresentados no lado direito da tabela como valores de evidência ($e_{i,1}, e_{i,2}$ e $e_{i,3}$), tal que $evaluate_{conf} = |f(y_j) - threshold|$ e $threshold = 0,5$.

Tabela 4.1: Exemplo ilustrativo do método de avaliação *conf*. Neste exemplo é considerado o limiar 0,5 para processar os valores brutos de *score* e obter os valores de confiança.

	score bruto				evaluate _{conf}		
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$
E_1	0,70	0,30	0,31	$\Rightarrow$	0,20	0,20	0,19
E_2	0,35	0,42	0,60		0,15	0,08	0,10
E_3	0,45	0,51	0,80		0,05	0,01	0,30
E_4	0,48	0,52	0,80		0,02	0,02	0,30
E_5	0,20	0,30	0,49		0,30	0,20	0,01

Como mencionado anteriormente, esses valores ainda devem ser agregados para um único valor e, assim, formar o valor de evidência final de cada objeto. As estratégias para agregação desses valores são discutidas posteriormente.

Amostragem por relevância (score): (Esuli e Sebastiani, 2009).

Essa estratégia tenta medir a probabilidade do rótulo ser relevante para o objeto. Para tanto, os valores brutos de *score* retornados pelos classificadores são diretamente utilizados como valor da função de avaliação *evaluate*. Altos valores de *score* significam maior probabilidade do rótulo (positivo) ser relevante para o objeto. Por outro lado, baixos valores de *score* significam menor probabilidade do rótulo (positivo) ser relevante para o objeto, ou seja, maior probabilidade do rótulo negativo ser relevante para o objeto. A ideia intuitiva é rotular

por meio do oráculo os objetos que tenham a maior probabilidade de conter rótulos positivos, ou mesmo negativos, dependendo da situação desejada. Esse método tem a vantagem de ser independente de limiar de corte. Quando o intuito é obter os rótulos positivos, essa abordagem é chamada *MaxScore*, pois os maiores valores de *score* são selecionados. Caso contrário, a abordagem é denominada *MinScore*, quando os objetos com maior probabilidade de conter rótulos negativos são selecionados. [Esuli e Sebastiani \(2009\)](#) utilizaram a abordagem *MaxScore* como função de avaliação de pares objeto-rótulo. No entanto, essa estratégia não apresentou bons resultados. Desse modo, neste trabalho será avaliada a abordagem *MinScore*, a qual é genericamente chamada como *score* neste trabalho.

Na Tabela 4.2 é apresentado um exemplo ilustrativo desta estratégia.

Tabela 4.2: Exemplo ilustrativo do método de avaliação *score*. Neste caso, os próprios valores brutos de *score* são utilizados como função de avaliação.

	<i>score</i> bruto				<i>evaluate_{score}</i>		
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$
E_1	0,70	0,30	0,31	$\Rightarrow$	0,70	0,30	0,31
E_2	0,35	0,42	0,60		0,35	0,42	0,60
E_3	0,45	0,51	0,80		0,45	0,51	0,80
E_4	0,48	0,52	0,80		0,48	0,52	0,80
E_5	0,20	0,30	0,49		0,20	0,30	0,49

Amostragem por incerteza baseada em *ranking* (*rank*): [\(Singh et al., 2010\)](#).

Essa estratégia funciona como uma abordagem de normalização para os valores obtidos da estratégia *baseada em confiança*. Os valores de *score* retornados pelo classificador base são utilizados para criar uma ordenação (*ranking*) dos objetos para cada rótulo. Um corte que separa os objetos preditos como positivos dos objetos preditos como negativos é definido baseado em um limiar, o qual, em geral, também é considerado igual a 0,5. A função de avaliação desta estratégia retorna como evidência um valor que representa o quão longe um objeto está do limiar da fronteira de decisão. Essa é uma noção intuitiva semelhante à estratégia *conf*. No entanto, no caso da estratégia *rank*, o interesse é apenas identificar a relação de ordem entre os objetos. Desse modo, valores de diferença entre as predições, principalmente em casos mais extremos, são desconsiderados, o que torna a estratégia *rank* menos susceptível a diferentes normalizações de valores e diferentes pesos entre valores de *score* dos diferentes classificadores binários de cada rótulo.

A Equação 4.1 define a função de avaliação desta estratégia.

$$e_{i,j} = |pos_{t,j} + 0,5 - pos_{i,j}| \quad (4.1)$$

onde $pos_{i,j}$ é a posição no *ranking* relativo ao rótulo y_j do objeto E_i e $pos_{t,j}$ é a posição no *ranking* de objetos do rótulo y_j em que ocorre a transição de objetos preditos como positivos para preditos como negativos, ou seja, $pos_{t,j}$ é classificado como positivo e $pos_{t+1,j}$ é classificado como negativo. Desse modo, $pos_{t,j} + 0,5$ representa o ponto de corte entre os exemplos positivos e negativos.

Além disso, o valor $e_{i,j}$ também é escalado linearmente pela proporção entre a distribuição de exemplos positivos e negativos de cada rótulo. Desse modo, o problema de desbalanceamento entre as classes é tratado, o que evita exemplos de classes minoritárias serem mais prováveis de estarem perto do ponto de transição do *ranking* apenas pelo próprio fato de pertecerem à classe minoritária.

Na Tabela 4.3 é apresentado um exemplo ilustrativo de aplicação do método *rank* como função de avaliação.

Tabela 4.3: Exemplo ilustrativo do método de avaliação *rank*. O limiar 0,5 e distribuição uniforme de rótulos (sem desbalanceamento) são considerados.

	score bruto				evaluate _{rank}		
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$
E_1	0,70	0,30	0,31	$\Rightarrow$	0,5	1,5	1,5
E_2	0,35	0,42	0,60		2,5	0,5	0,5
E_3	0,45	0,51	0,80		1,5	0,5	2,5
E_4	0,48	0,52	0,80		0,5	1,5	2,5
E_5	0,20	0,30	0,49		3,5	1,5	0,5

Baseada em desacordo: (Hung e Lin, 2011; Yang et al., 2009).

Diferentemente das outras abordagens, essa estratégia utiliza dois classificadores base com o intuito de identificar a diferença entre suas predições para cada exemplo não rotulado. A ideia intuitiva deste método é consultar o oráculo pelo rótulo do(s) exemplo(s) que tenha(m) o maior desacordo em relação às predições dos classificadores, *i.e.*, o exemplo que poderia ser o mais informativo para resolver esse conflito. Três maneiras de medir o grau de desacordo dos valores preditos foram propostas: *MMR*; *HLR*; e *SHLR*.

MMR utiliza um classificador principal para obter *valores de confiança* da predição dos exemplos e um classificador auxiliar para obter *valores de decisão* (positivo ou negativo apenas) das predições. Esse classificador auxiliar é utilizado como estimativa do rótulo verdadeiro e, dessa

maneira, para determinar o quão conflitante são as predições dos dois classificadores.

HLR considera o grau de conflito de maneira mais rígida usando apenas os valores de decisão de ambos os classificadores para decidir se existe acordo ou desacordo entre cada predição de rótulo de um objeto.

SHLR tenta fazer um balanço entre *MMR* e *HLR* por meio de uma função que define a influência que cada abordagem exerce no *score* final.

Vale observar que ambos os classificadores devem preferencialmente retornar valores de *score* da mesma natureza, *i.e.*, deve-se evitar comparar valores de diferentes tipos como probabilidade e margem, por exemplo.

Durante o desenvolvimento deste trabalho essa estratégia foi avaliada em experimentos preliminares. Porém não apresentou bons resultados apesar do maior esforço computacional exigido pelo método. Desse modo, esse tipo de abordagem foi desconsiderada na continuidade do trabalho.

Propostas para a função de agregação AGGREGATE

Depois de obtidos os valores de evidência para cada par objeto-rótulo, é necessário agregar esses valores para se obter um único valor de evidência para cada objeto. Algumas estratégias propostas e utilizadas pela comunidade para esse fim são descritas a seguir.

Média (*avg*): (Esuli e Sebastiani, 2009; Singh et al., 2010) A estratégia *avg* calcula a média dos valores de evidência sobre todos os pares objeto-rótulo e retorna um único valor para um determinado objeto. Desse modo, o valor de evidência final e_i para o objeto E_i dado seus correspondentes pares objeto-rótulo de evidência $e_{i,j}$ é definido pela Equação 4.2.

$$aggregate_{avg}(\{e_{i,j}\}_{j=1}^q) = e_i = \frac{\sum_{j=1}^q e_{i,j}}{q} \quad (4.2)$$

Mínimo/Máximo (*min/max*): (Brinker, 2006; Esuli e Sebastiani, 2009)

Esta abordagem visa maximizar a quantidade de informação para um dentre os q possíveis rótulos, ou seja, escolher o maior valor de evidência objeto-rótulo para um dado objeto. O intuito é fornecer ao oráculo aqueles exemplos que, caso rotulados, fornecerão mais informação para pelo menos um dos rótulos do problema. Nos casos de *conf*

(*MinConfidence*), *score* (*MinScore*) e *rank*, o menor (*min*) valor de evidência $e_{i,j}$, com $j = 1..q$, é escolhido para representar a evidência final do exemplo E_i . No caso de *MaxScore*, o maior valor é o escolhido.

Na Equação 4.3 é definido o conceito dessa estratégia para o operador *min*.

$$e_i = \text{aggregate}_{\min}(\{e_{i,j}\}_{j=1}^q) = \min(\{e_{i,j}\}_{j=1}^q) \quad (4.3)$$

Round Robin (rr): (Esuli e Sebastiani, 2009)

Esta estratégia tem como intuito eleger vários objetos a serem rotulados pelo oráculo, cada um maximizando a informação para diferentes rótulos. Uma fila de prioridade na rotulação para cada rótulo, ordenada pela função de avaliação, é gerada. Em uma determinada iteração do aprendizado ativo, esse método escolhe um objeto de cada uma das filas de prioridades até que se atinja o número de objetos que se deseja rotular. Desse modo, essa estratégia tem sentido apenas com a rotulação em lote de objetos, pois assim é possível eleger representantes de diferentes rótulos. Como será descrito na Seção 4.2.4, este trabalho utiliza rotulação de um objeto por iteração e, desse modo, essa estratégia não foi utilizada neste trabalho.

As abordagens descritas foram propostas em trabalhos relacionados da comunidade. Neste trabalho propomos uma nova estratégia denominada *dev*, descrita a seguir.

Desvio (dev): A abordagem por nós proposta está baseada nas diferenças (desvios) entres os valores de evidências $e_{i,j}$ de cada objeto. A intuição desta estratégia é que um objeto que contém pequenas diferenças nos valores entre as evidências dos rótulos preditos como positivos e as evidências dos rótulos preditos como negativo apontam incerteza na predição do objeto, o que o torna um candidato potencial para a rotulação pelo oráculo. A Equação 4.4 define a estratégia *dev*.

$$\text{aggregate}_{\text{dev}}(\{e_{i,j}\}_{j=1}^q) = e_i = \text{mediapos}(\{e_{i,j}\}_{j=1}^q) - \text{primeironeg}(\{e_{i,j}\}_{j=1}^q) \quad (4.4)$$

onde a função *mediapos* retorna o valor médio das evidências que são classificadas com rótulos positivos e a função *primeironeg* retorna o valor de evidência do rótulo mais próximo de ser classificado como positivo que é, porém, classificado como negativo.

Desse modo, quanto menor é o valor da função $aggregate_{dev}$, maior prioridade tem o objeto de ser selecionado para rotulação pelo oráculo. Essa estratégia tem maior sentido quando aplicada aos valores de evidência retornados pela função de avaliação $score$. Além disso, essa estratégia é especialmente boa quando utilizada com classificadores que fornecem bons valores de $score$ para ordenar os rótulos de um dado exemplo. Desse modo, este método é chamado neste trabalho como $score_{dev}$.

Nas Tabelas 4.4, 4.5 e 4.6, os exemplos ilustrativos anteriormente apresentados são incrementados com as funções de agregação pertinentes para cada função de avaliação. Desse modo, é possível observar diferentes escolhas de objetos pelos diferentes métodos. Casos de empates são decididos aleatoriamente.

Tabela 4.4: Exemplo ilustrativo do método de avaliação $conf$ com aplicação das funções de agregação avg e min . Os objetos em negrito seriam os selecionados para rotulação pelo oráculo.

	score bruto				evaluate _{conf}				aggregate	
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$		avg	min
E_1	0,70	0,30	0,31		0,20	0,20	0,19		0,20	0,19
E_2	0,35	0,42	0,60		0,15	0,08	0,10		0,11	0,08
E_3	0,45	0,51	0,80	$\Rightarrow$	0,05	0,01	0,30	$\Rightarrow$	0,12	0,01
E_4	0,48	0,52	0,80		0,02	0,02	0,30		0,11	0,02
E_5	0,20	0,30	0,49		0,30	0,20	0,01		0,17	0,01

Tabela 4.5: Exemplo ilustrativo do método de avaliação $score$ com aplicação das funções de agregação avg , min e dev . Para dev , apenas a evidência com maior valor de $score$ foi considerada positivo. Os objetos em negrito seriam os selecionados para rotulação pelo oráculo.

	score bruto				evaluate _{score}				aggregate		
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$		avg	min	dev
E_1	0,70	0,30	0,31		0,70	0,30	0,31		0,44	0,30	0,39
E_2	0,35	0,42	0,60		0,35	0,42	0,60		0,46	0,35	0,18
E_3	0,45	0,51	0,80	$\Rightarrow$	0,45	0,51	0,80	$\Rightarrow$	0,59	0,45	0,29
E_4	0,48	0,52	0,80		0,48	0,52	0,80		0,60	0,48	0,28
E_5	0,20	0,30	0,49		0,20	0,30	0,49		0,33	0,20	0,19

Tabela 4.6: Exemplo ilustrativo do método de avaliação $rank$ com aplicação das funções de agregação avg e min . Os objetos em negrito seriam os selecionados para rotulação pelo oráculo.

	score bruto				evaluate _{rank}				aggregate	
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$		avg	min
E_1	0,70	0,30	0,31		0,5	1,5	1,5		1,17	0,50
E_2	0,35	0,42	0,60		2,5	0,5	0,5		1,17	0,50
E_3	0,45	0,51	0,80	$\Rightarrow$	1,5	0,5	2,5	$\Rightarrow$	1,50	0,50
E_4	0,48	0,52	0,80		0,5	1,5	2,5		1,50	0,50
E_5	0,20	0,30	0,49		3,5	2,5	3,5		1,83	0,50

4.2.4 Protocolo de experimentos

Diversos aspectos do protocolo de experimentos devem ser considerados ao utilizar e avaliar métodos de aprendizado ativo. Da mesma forma que o aprendizado monorrótulo, além das próprias estratégias de aprendizado ativo, a maneira como esses métodos são avaliados também é uma característica fundamental no desenvolvimento de novas estratégias para o contexto multirrótulo. Alguns aspectos importantes a serem considerados são: o tamanho do conjunto de exemplos inicialmente rotulados; o tamanho do lote de exemplos a serem rotulados e incorporados ao modelo; o conjunto de exemplos utilizado como teste para medir a eficácia do modelo criado; a estratégia de validação cruzada (ou *holdout*); quando parar o aprendizado ativo; e as medidas de avaliação utilizadas.

A seguir são descritas os principais aspectos que envolvem a avaliação experimental em aprendizado ativo.

Composição do conjunto inicial de exemplos rotulados

Métodos de aprendizado ativo supervisionado exigem um conjunto inicial de (poucos) exemplos rotulados. A maneira como esse conjunto é composto influencia diretamente na qualidade do aprendizado ativo e, consequentemente, no custo da rotulação. Uma maneira simples de construir esse conjunto inicial é coletar alguns exemplos aleatoriamente para que o oráculo os rotule. Outra maneira, seria utilizar técnicas de agrupamento (*clustering*) para, mesmo sem exemplos rotulados, buscar por exemplos representativos para compor o conjunto inicial.

Nos trabalhos relacionados ao aprendizado ativo multirrótulo, os autores utilizaram a primeira estratégia: coletar aleatoriamente um conjunto pequeno inicial de exemplos rotulados antes de iniciar o processo ativo. No entanto, o tamanho desse conjunto inicial é uma variável importante no processo. Esse conjunto não deve ser grande demais, pois o processo de rotulação ficaria custoso antes mesmo de se iniciar o aprendizado ativo. Por outro lado, o conjunto inicial também não pode ser pequeno demais a ponto de não fornecer evidências suficientes para a escolha de exemplos representativos pelo algoritmo de aprendizado ativo.

Diversos tamanhos de conjunto inicial foram utilizados em trabalhos relacionados. Em (Brinker, 2006), o conjunto inicial foi composto por 10 exemplos coletados aleatoriamente. Em (Yang et al., 2009), foram avaliados conjuntos de tamanhos que variaram de 100 a 500 exemplos. Em (Esuli e Sebastiani, 2009), os exemplos da coleção foram ordenados cronologicamente e os 100 primeiros objetos foram selecionados para compor o conjunto inicial.

No entanto, conjuntos de dados com características diferentes necessitam de diferentes números de exemplos inicialmente rotulados. Qual é o tamanho adequado para um dado problema multirrótulo?

Em (Singh et al., 2010), um exemplo positivo e um negativo por rótulo foram selecionados para compor o conjunto inicial. Como normalmente se tem a informação do conjunto de rótulos L antes de se iniciar a rotulação de exemplos, essa poderia ser uma maneira de adaptar o conjunto inicial de exemplos às características do problema. A vantagem dessa estratégia é a garantia de uma representatividade mínima de exemplos para todos os rótulos. No entanto, essa não é uma estratégia realista ou factível para todos os problemas multirrótulo. A menos que o rotulador/especialista tenha em mente todos esses exemplos, positivos e negativos para cada rótulo, será necessário coletar (possivelmente de maneira aleatória) e examinar diversos exemplos até se obter esse conjunto mínimo representativo. O que o rotulador fará com exemplos examinados de um determinado rótulo que já tenham representantes no conjunto inicial? Descartá-los não é uma boa estratégia, afinal os exemplos já foram examinados e rotulados, pelo menos mentalmente, pelo especialista.

Um critério mais realista para a composição do conjunto inicial, e que não descartaria desnecessariamente exemplos já examinados, poderia ser: *rotular exemplos aleatoriamente até que exista **pelo menos** um exemplo positivo e um negativo para cada rótulo*. Essa estratégia também tem a vantagem de garantir uma representatividade mínima para cada rótulo. A desvantagem é que pode ser necessário coletar e examinar exemplos em demasia se houver um grande desbalanceamento na frequência de ocorrência dos diferentes rótulos. Os rótulos menos frequentes podem necessitar a coleta de muitos exemplos para serem representados, o que pode mitigar ou até eliminar os benefícios do aprendizado ativo já na composição do conjunto inicial de exemplos rotulados.

Neste trabalho propomos um critério que possibilita o controle do custo para composição do conjunto inicial de exemplos rotulados e que, mesmo assim, considera a característica do domínio (quantidade de rótulos do problema). Essa estratégia consiste em rotular aleatoriamente exemplos até que a soma dos rótulos que compõem os multirrótulos de cada exemplo do conjunto D_l de exemplos já rotulado seja maior ou igual a $FRI \times q$, onde FRI é um *Fator de Rotulação Inicial* definido pelo usuário e q a quantidade de rótulos do domínio. A Equação 4.5 define o critério de parada de rotulação dessa estratégia.

$$FRI \times q \geq \sum_i^{|D_l|} |Y_i| \quad (4.5)$$

onde Y_i é o multirrótulo do exemplo E_i pertencente ao conjunto de exemplos já rotulados D_l .

No Algoritmo 5 é apresentado o procedimento para a composição no conjunto inicial de exemplos rotulados utilizando o critério FRI .

```

input :  $q$ : número de possíveis rótulos do domínio;
          $D_u$ : conjunto de exemplos não rotulados;
          $oracle$ : o rotulador;
          $FRI$ : fator de rotulação inicial.
output:  $D_l$ : conjunto inicial de exemplos rotulados.
 $acum \leftarrow 0$ 
repeat
     $E_i \leftarrow randomchoose(D_u)$  //escolha aleatório do exemplo
     $D_u \leftarrow D_u - \{E_i\}$  //atualização do conjunto
     $Y_i \leftarrow oracle(E_i)$  //rotulação do exemplo pelo oráculo
     $D_l \leftarrow D_l + \{(E_i, Y_i)\}$  //adição do exemplo ao conjunto inicial
     $acum \leftarrow acum + |Y_i|$  //atualização do contador de rótulos
until  $FRI \times q \geq acum$  ;
return  $D_l$ 

```

Algoritmo 5: Procedimento para composição do conjunto inicial de exemplos rotulados proposto e utilizado neste trabalho.

Para exemplificar, considere o conjunto de dados *Scene*, o qual contém $q = 6$ possíveis rótulos. Caso $FRI = 2$, seriam rotulados exemplos aleatoriamente até que $FRI \times q = 12$ monorrótulos tenham sido coletados. Desse modo, é possível compor o conjunto inicial de exemplos rotulados de maneira que torne possível uma comparação mais adequada entre conjuntos de dados com diferentes características. Além disso, o usuário pode controlar o fator FRI para aumentar/diminuir a quantidade necessária de exemplos rotulados inicialmente e, assim, fornecer também uma representatividade mínima, pelo menos para a maioria dos rótulos, sem coletar tantos exemplos de forma aleatória a ponto de eliminar os benefícios do aprendizado ativo. Apesar de não ser possível garantir que todos os rótulos terão exemplos iniciais que os representem, nada impede que no decorrer do processo de aprendizado ativo novos exemplos sejam rotulados pelo oráculo com rótulos ainda não presentes no conjunto, já que o oráculo tem conhecimento dos q possíveis rótulos do domínio. Essa estratégia foi utilizada para a realização dos experimentos deste trabalho.

Tamanho do lote de objetos

O tamanho do lote de exemplos a serem rotulados e incorporados ao modelo define quantos exemplos em cada iteração do aprendizado ativo serão consultados. Em (Singh et al., 2010) e (Brinker, 2006), apenas um exemplo por iteração é consultado, rotulado e incorporado ao modelo. Em (Esuli e

[Sebastiani, 2009](#)) e ([Yang et al., 2009](#)) são selecionados 50 exemplos por iteração, mas [Yang et al. \(2009\)](#) também executam experimentos com 20 exemplos selecionados a cada iteração.

O ideal em experimentos com aprendizado ativo é avaliar o tamanho do lote de objetos como uma variável experimental, o que possibilitaria tirar conclusões também em relação a essa característica. No entanto, escolher apenas um objeto a cada iteração pode ser a melhor opção se for levada em consideração o ciclo virtuoso que o aprendizado ativo pode oferecer. A idéia é que cada objeto selecionado pelo aprendizado ativo traga informação para melhorar o modelo anteriormente criado e, assim, também melhorar a escolha do próximo objeto a ser rotulado e incorporado ao modelo. Tomando esse pressuposto em consideração e a necessidade de limitar o escopo dos experimentos, neste trabalho é selecionado um único objeto a cada iteração do aprendizado ativo, ou seja, o tamanho do lote de objetos é igual a um.

Critério de parada da rotulação

Outra decisão importante em aprendizado ativo, tanto na aplicação prática das técnicas quanto no cenário experimental, é determinar quando a aplicação deve parar de rotular novos exemplos. Uma maneira de decidir quando parar poderia ser o ponto em que o custo de adquirir novos exemplos rotulados é maior que o custo dos erros cometidos pelo classificador atual. Outra maneira poderia ser identificar se a acurácia do classificador já atingiu um platô, ou seja, rotular mais exemplos traria pouca ou nenhuma melhora no modelo atual de classificação.

Alguns critérios de paradas foram propostos para o aprendizado ativo monorrótulo ([Vlachos, 2008](#); [Bloodgood e Vijay-Shanker, 2009](#); [Olsson e Tomanek, 2009](#)), todos baseados na ideia de propor uma medida para identificar a estabilidade do aprendizado, os quais poderiam ser adaptados para o cenário multirrótulo. No entanto, como exposto em ([Settles, 2010](#)), o critério real de parada para aplicações práticas é baseado no fator econômico ou outros fatores externos, ou seja, a definição de limiares, configurações do algoritmo de aprendizado ativo e outros aspectos técnicos provavelmente vêm depois da definição dos recursos financeiros.

Neste trabalho, o processo de aprendizado ativo foi executado por 100 iterações para todos os conjuntos de dados. Em aplicações práticas, esse número poderia ser maior ou menor, dependendo das condições e dos recursos disponíveis. No entanto, com intuito de simplificar o processo de avaliação dos experimentos, e considerando que o número inicial de exemplos rotulados, proposto neste trabalho, já leva em conta as características do conjunto de dados, esse critério estático de parada da rotulação foi ado-

tado.

Conjunto de teste: *independente* ou *remanescente*

Existem basicamente duas abordagens a se considerar na escolha do conjunto de teste no qual é calculada a eficácia do classificador gerado pelo algoritmo de aprendizado. A primeira abordagem é considerar um conjunto de teste independente do utilizado pelo algoritmo de aprendizado ativo — Figura 4.3#1). Essa abordagem foi utilizada em (Esuli e Sebastiani, 2009) e (Brinker, 2006)³. A outra abordagem é utilizar o restante dos exemplos no conjunto de consulta para avaliar o classificador gerado — Figura 4.3#2). Essa abordagem foi utilizada em (Singh et al., 2010) e (Yang et al., 2009). É importante observar que a medida de qualidade do modelo aferida utilizando essa segunda abordagem é adequada somente para exemplos remanescentes que podem ter uma distribuição diferente daquela do conjunto de dados, mas não necessariamente para novos exemplos não rotulados.

Figura 4.3: Protocolo de experimentos para avaliação do classificador utilizando um conjunto *independente* e o conjunto de exemplos de consulta *remanescentes*.

Avaliação do aprendizado ativo

Assim como no aprendizado ativo monorrótulo, a curva de aprendizado é a abordagem padrão utilizada pela comunidade para aferir a qualidade dos métodos de aprendizado ativo multirrótulo. A partir dela também podem ser extraídos indicadores da qualidade do aprendizado ativo, como o valor de eficácia final (FV)⁴ do classificador após todas as iterações do aprendizado ativo tenham sido realizadas, utilizado em (Singh et al., 2010; Yang et al., 2009; Esuli e Sebastiani, 2009), e a área sob a curva de aprendizado (AULC).

³Esse trabalho não descreveu explicitamente a abordagem do conjunto de teste utilizada. No entanto, um conjunto independente de teste foi aparentemente utilizado.

⁴Final Value em Inglês.

4.3. Avaliação experimental

Alguns dos algoritmos de aprendizado ativo descritos na Seção 4.2.3, página 74, foram considerados para execução de experimentos. Esses algoritmos foram implementados utilizando o *framework* *Mulan*⁵ (Tsoumakas et al., 2011) e foram avaliados em treze conjuntos de dados multirrótulo de diferentes domínios, descritos na Seção 2.6, página 41. A seguir são descritos em maiores detalhes os experimentos realizados neste trabalho relacionados ao aprendizado ativo multirrótulo.

4.3.1 Configuração dos experimentos

Como mencionado anteriormente, os algoritmos de aprendizado ativo multirrótulo são instanciados com duas funções:

1. Função de avaliação de objeto-rótulo (*evaluate*)
2. Função de agregação dos valores de evidência de objeto-rótulo (*aggregate*)

Para o primeiro componente foram consideradas as seguintes alternativas:

- Amostragem por incerteza (*conf*)
- Amostragem por relevância (*score*)
- Amostragem por incerteza baseada em *ranking* (*rank*)

Para a função de agregação foram consideradas as seguintes estratégias:

- Média (*avg*)
- Desvio (*dev*)

Como mencionado, experimentos preliminares indicam que as abordagens *min/max* apresentam resultados bastante instáveis. A estratégia *avg* apresenta maior estabilidade e, mesmo quando é superada por *min/max*, a diferença entre as abordagens é baixa. Desse modo, decidiu-se limitar o escopo desconsiderando a abordagem *min/max*.

Além disso, deve-se também considerar que a abordagem *dev* é adequada apenas para *score* como função de avaliação. Desse modo, além da

⁵As implementações dos algoritmos de aprendizado ativo encontra-se disponíveis em <http://www.labc.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/active-learning.zip>

abordagem *baseline (random)*, na Tabela 4.7 são apresentadas as combinações das funções *evaluate* e *aggregate* consideradas para os algoritmos de aprendizado ativo multirrótulo utilizados neste trabalho.

Função <i>evaluate</i>	Função <i>aggregate</i>
<i>conf</i>	<i>avg</i>
<i>rank</i>	<i>avg</i>
<i>score</i>	<i>avg</i>
<i>score</i>	<i>dev</i>

Tabela 4.7: Algoritmos de aprendizado ativo utilizados considerando diferentes funções *evaluate* e *aggregate*.

Quanto aos algoritmos base de aprendizado multirrótulo, foram consideradas duas alternativas: uma indutiva e outra transdutiva. O algoritmo indutivo é utilizado para experimentos que utilizam o protocolo *independente* de teste. Esse algoritmo também poderia ser utilizado no protocolo *remanescente*, no entanto, devido à natureza desse protocolo indicar um problema de inferência transdutiva, um algoritmo de classificação multirrótulo com inferência transdutiva foi utilizado. Desse modo, os seguintes algoritmos de aprendizado supervisionado multirrótulo foram considerados:

- Protocolo *independente*: *BR(SVM)*
- Protocolo *remanescente*: *RCut(LPBHN)*

O algoritmo base monorrótulo *Support Vector Machine (SVM)* foi utilizado, especificamente a implementação chamada *LIBLinear*⁶ que é composta por *kernel* linear. A diferença básica dessa implementação em relação a outras implementações de *SVM* linear é uma otimização em termos de tempo de execução obtida para conjunto de dados com atributos esparsos. Essa característica é desejável em experimentos envolvendo aprendizado ativo devido ao custo de tempo envolvido para treinar um grande número de modelos: um modelo para cada rótulo do problema para cada iteração do aprendizado ativo. Além disso, esse algoritmo foi configurado de maneira a retornar valores normalizados de estimativas de probabilidade como confiança de predição dos exemplos. Os outros parâmetros do algoritmo foram definidos conforme o padrão da implementação *LIBLinear*. Essa implementação foi combinada com o método *BR* para formar o algoritmo de aprendizado multirrótulo de inferência indutiva.

O algoritmo base *LPBHN*, proposto por Rossi et al. (2013), foi utilizado para realização da classificação transdutiva. Esse algoritmo é baseado em grafos e também apresenta otimizações em termos de tempo de execução

⁶<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

para conjunto de dados esparsos e tem apresentado bons resultados para diversos problemas monorrótulo multiclasse. Para utilizá-lo como algoritmo multirrótulo, o método *RCut* baseado em *ranking* foi escolhido. Esse método transforma o problema multirrótulo em um problema multiclasse. Desse modo, o classificador base retorna um *ranking* de rótulos para um novo exemplo a ser predito. Os t primeiros rótulos do *ranking* são preditos para esse novo exemplo, onde t é o número inteiro mais próximo da cardinalidade (número médio de rótulos por exemplo) do conjunto de exemplos de treinamento.

Essa estratégia de classificação multirrótulo baseada em *ranking* foi escolhida devido à característica do algoritmo-base *LPBHN*, o qual fornece boas ordenações dos rótulos preditos para um dado objeto. No entanto, os valores de confiança retornados pelo algoritmo não são adequados para realizar a ordenação (*ranking*) de objeto para um dado rótulo. Desse modo, as funções de avaliação *conf* e *rank* não são aplicáveis para esse algoritmo, pois elas têm como pressuposto a existência de uma boa ordenação dos objetos dado um rótulo para funcionarem adequadamente.

Na Tabela 4.8 são apresentadas as configurações dos experimentos de aprendizado ativo executados e apresentados neste trabalho.

Tabela 4.8: Configurações de aprendizado ativo utilizadas nos experimentos.

	Protocolo de experimento	
	<i>independente</i> BR(SVM)	<i>remanescente</i> RCut(LPBHN)
<i>random</i>	✓	✓
<i>conf avg</i>	✓	—
<i>rank avg</i>	✓	—
<i>score avg</i>	✓	✓
<i>score dev</i>	✓	✓

Todos os experimentos foram executados utilizando a estratégia de validação cruzada *10-fold cross-validation*. No cenário transdutivo, no qual o protocolo *remanescente* é utilizado, as partições independentes de teste de cada uma das dez iterações são descartadas, já que os exemplos remanescentes do conjunto de consulta são utilizados para avaliar o modelo criado, conforme descrito na Figura 4.3#2), página 87. Além disso, vale observar que a validação cruzada *10-fold cross-validation* já implica em utilizar dez diferentes amostragens aleatórias para o método *random*.

4.3.2 Resultados e discussão

A seguir são apresentados os resultados dos experimentos realizados utilizando os protocolos *independente* e *remanescente*. Inicialmente, é apresentado o gráfico da curva de aprendizado para um conjunto de dados con-

siderado na avaliação experimental. Posteriormente, os resultados para todos os conjuntos de dados são apresentados resumidamente utilizando as medidas *FV* e *AULC*. Como mencionado, o método *random* se refere ao *baseline* (aprendizado passivo) e o método *score dev* se refere à proposta deste trabalho. Os outros métodos são propostas de trabalhos relacionados.

Protocolo de experimentos com conjunto *independente* de teste

Na Figura 4.4 é apresentada a curva de aprendizado para os algoritmos considerados neste trabalho utilizando o conjunto de dados *bibtex*. As curvas de aprendizado para todos os conjuntos de dados são apresentadas no Apêndice A.

Figura 4.4: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *bibtex*.

Observa-se que o método de aprendizado ativo *score dev* é constantemente superior a maneira passiva de rotulação (*random*) para todas os quatro cenários considerados. O método *score avg* também é superior ao *random* em todos os cenários, exceto para $FRI = 1$ e *Micro-F* como medida de avaliação, cenário no qual o método apresentou o pior resultado entre todos os outros métodos. Os outros métodos de aprendizado ativo não foram capazes de superar constantemente o método *random* em nenhum cenário.

Na Tabela 4.9, os resultados dos experimentos de aprendizado ativo são apresentados de maneira sumarizada para todos conjunto de dados, utili-

zando os valores *FV* de eficácia na última iteração do aprendizado ativo para cada estratégia. Além disso, na última linha de cada sub-tabela (*melhor ou igual random*) é apresentada uma comparação dos métodos de aprendizado ativo relativa ao método *baseline* (*random*).

Tabela 4.9: Valores *FV* de eficácia na última iteração da curva de aprendizado para cada método de aprendizado ativo com o protocolo *independente*. Os valores entre parênteses se referem à posição no *ranking* do método.

FRI=1										
	Micro-F					Macro-F				
	<i>random</i>	<i>conf avg</i>	<i>rank avg</i>	<i>score avg</i>	<i>score dev</i>	<i>random</i>	<i>conf avg</i>	<i>rank avg</i>	<i>score avg</i>	<i>score dev</i>
#1 <i>bibtex</i>	.15 (2.5)	.10 (5.0)	.15 (2.5)	.13 (4.0)	.18 (1.0)	.02 (3.0)	.02 (3.0)	.02 (3.0)	.02 (3.0)	.02 (3.0)
#2 <i>cal500</i>	.34 (4.5)	.34 (4.5)	.36 (2.0)	.36 (2.0)	.36 (2.0)	.20 (3.0)	.19 (5.0)	.21 (1.0)	.20 (3.0)	.20 (3.0)
#3 <i>corel16k</i>	.03 (4.0)	.02 (5.0)	.04 (2.0)	.04 (2.0)	.04 (2.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)
#4 <i>corel5k</i>	.04 (3.0)	.03 (5.0)	.04 (3.0)	.05 (1.0)	.04 (3.0)	.31 (3.0)	.31 (3.0)	.31 (3.0)	.31 (3.0)	.31 (3.0)
#5 <i>emotions</i>	.58 (4.0)	.60 (3.0)	.61 (2.0)	.57 (5.0)	.64 (1.0)	.54 (4.0)	.56 (3.0)	.58 (2.0)	.53 (5.0)	.61 (1.0)
#6 <i>enron</i>	.47 (4.0)	.40 (5.0)	.48 (3.0)	.52 (1.0)	.49 (2.0)	.26 (3.0)	.25 (4.5)	.27 (1.5)	.27 (1.5)	.25 (4.5)
#7 <i>llog</i>	.03 (2.0)	.03 (2.0)	.02 (4.0)	.01 (5.0)	.03 (2.0)	.39 (3.0)	.39 (3.0)	.39 (3.0)	.39 (3.0)	.39 (3.0)
#8 <i>medical</i>	.55 (5.0)	.57 (4.0)	.60 (2.5)	.61 (1.0)	.60 (2.5)	.56 (5.0)	.58 (2.5)	.58 (2.5)	.60 (1.0)	.57 (4.0)
#9 <i>ohsumed</i>	.12 (4.0)	.11 (5.0)	.15 (2.5)	.24 (1.0)	.15 (2.5)	.04 (3.5)	.03 (5.0)	.05 (2.0)	.08 (1.0)	.04 (3.5)
#10 <i>scene</i>	.55 (4.0)	.43 (5.0)	.58 (1.0)	.57 (2.5)	.57 (2.5)	.54 (4.0)	.39 (5.0)	.58 (1.5)	.57 (3.0)	.58 (1.5)
#11 <i>slashdot</i>	.09 (4.0)	.23 (1.0)	.12 (3.0)	.06 (5.0)	.16 (2.0)	.18 (4.5)	.21 (1.0)	.19 (3.0)	.18 (4.5)	.20 (2.0)
#12 <i>tmc2007</i>	.51 (3.0)	.47 (5.0)	.50 (4.0)	.56 (1.5)	.56 (1.5)	.19 (2.5)	.13 (5.0)	.18 (4.0)	.31 (1.0)	.19 (2.5)
#13 <i>yeast</i>	.58 (3.0)	.56 (5.0)	.60 (1.0)	.58 (3.0)	.58 (3.0)	.32 (3.0)	.28 (5.0)	.33 (2.0)	.34 (1.0)	.30 (4.0)
<i>ranking médio</i>	3.6	4.2	2.5	2.6	2.1	3.4	3.7	2.4	2.5	2.9
<i>melhor/igual random</i>	—	38%	85%	69%	100%	—	54%	92%	92%	85%

FRI=5										
	Micro-F					Macro-F				
	<i>random</i>	<i>conf avg</i>	<i>rank avg</i>	<i>score avg</i>	<i>score dev</i>	<i>random</i>	<i>conf avg</i>	<i>rank avg</i>	<i>score avg</i>	<i>score dev</i>
#1 <i>bibtex</i>	.19 (3.5)	.20 (2.0)	.19 (3.5)	.18 (5.0)	.21 (1.0)	.03 (4.0)	.04 (1.5)	.03 (4.0)	.03 (4.0)	.04 (1.5)
#2 <i>cal500</i>	.34 (4.0)	.33 (5.0)	.35 (3.0)	.36 (1.5)	.36 (1.5)	.19 (4.5)	.19 (4.5)	.20 (2.0)	.20 (2.0)	.20 (2.0)
#3 <i>corel16k</i>	.06 (3.0)	.05 (5.0)	.06 (3.0)	.07 (1.0)	.06 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)
#4 <i>corel5k</i>	.08 (3.5)	.08 (3.5)	.08 (3.5)	.09 (1.0)	.08 (3.5)	.32 (3.0)	.32 (3.0)	.32 (3.0)	.32 (3.0)	.32 (3.0)
#5 <i>emotions</i>	.59 (4.0)	.62 (2.0)	.58 (5.0)	.60 (3.0)	.64 (1.0)	.54 (5.0)	.60 (2.0)	.55 (4.0)	.57 (3.0)	.62 (1.0)
#6 <i>enron</i>	.49 (4.0)	.48 (5.0)	.51 (2.5)	.53 (1.0)	.51 (2.5)	.28 (3.0)	.28 (3.0)	.28 (3.0)	.29 (1.0)	.27 (5.0)
#7 <i>llog</i>	.06 (3.5)	.07 (2.0)	.06 (3.5)	.04 (5.0)	.09 (1.0)	.39 (3.5)	.39 (3.5)	.39 (3.5)	.39 (3.5)	.40 (1.0)
#8 <i>medical</i>	.68 (5.0)	.70 (3.0)	.69 (4.0)	.72 (1.5)	.72 (1.5)	.63 (4.0)	.64 (2.0)	.63 (4.0)	.65 (1.0)	.63 (4.0)
#9 <i>ohsumed</i>	.19 (2.0)	.15 (4.5)	.15 (4.5)	.23 (1.0)	.18 (3.0)	.06 (2.5)	.05 (4.5)	.05 (4.5)	.08 (1.0)	.06 (2.5)
#10 <i>scene</i>	.57 (4.0)	.49 (5.0)	.60 (2.0)	.61 (1.0)	.58 (3.0)	.56 (4.0)	.47 (5.0)	.61 (1.5)	.61 (1.5)	.59 (3.0)
#11 <i>slashdot</i>	.19 (4.0)	.26 (2.0)	.23 (3.0)	.12 (5.0)	.28 (1.0)	.22 (4.0)	.23 (2.0)	.23 (2.0)	.20 (5.0)	.23 (2.0)
#12 <i>tmc2007</i>	.54 (3.5)	.53 (5.0)	.56 (1.5)	.56 (1.5)	.54 (3.5)	.22 (4.0)	.19 (5.0)	.24 (2.0)	.28 (1.0)	.23 (3.0)
#13 <i>yeast</i>	.59 (4.0)	.59 (4.0)	.60 (1.5)	.60 (1.5)	.59 (4.0)	.31 (4.5)	.32 (3.0)	.34 (1.5)	.34 (1.5)	.31 (4.5)
<i>ranking médio</i>	3.7	3.7	3.1	2.2	2.3	3.8	3.2	2.9	2.3	2.7
<i>melhor/igual random</i>	—	54%	85%	77%	92%	—	77%	92%	92%	92%

Como pode ser observado, diversos métodos de aprendizado ativo apresentaram valores de *FV* iguais ou superiores aos apresentados pelo método *random*. O método *conf avg*, no entanto, não apresentou um *ranking médio* superior ao *random* para $FRI = 1$ em ambas as medidas de avaliação utilizadas (*Micro/Macro-F*) e para $FRI = 5$ na medida de avaliação *Micro-F*. Os outros métodos de aprendizado ativo apresentaram valores de *ranking médio* superiores ao *random* em todos os casos.

Uma característica importante na avaliação de um método de aprendi-

zado ativo é o fato de ser constantemente superior, ou pelo menos igual, ao método *random*. Na prática, esse é um aspecto fundamental do método, pois diferentemente da avaliação de algoritmos de aprendizado padrão, não se tem os dados rotulados de antemão, já que objetivo do aprendizado ativo é exatamente obter bons exemplos rotulados. Assim, não é possível realizar de antemão uma previsão da eficácia de vários métodos de aprendizado ativo da maneira como pode ser realizado para escolher o algoritmo de aprendizado padrão. Desse modo, para medir a estabilidade do método de aprendizado ativo, na última linha da Tabela 4.9 é apresentado o valor *melhor ou igual random*. Esse valor se refere à porcentagem de conjunto de dados em que o método ativo foi superior ou igual ao método *random*.

Pode-se observar que o método *score dev* apresentou os melhores valores de estabilidade para a medida de avaliação *Micro-F*, na qual para 100% e 92% dos conjuntos de dados o método foi superior ou igual ao *random* para $FRI = 1$ e $FRI = 5$, respectivamente. O método *score dev*, juntamente com *rank avg* e *conf avg*, também apresentou o melhor valor de estabilidade para *Macro-F* e $FRI = 5$, com 92% dos casos melhor ou igual ao *random*. No cenário com *Macro-F* e $FRI = 1$, *rank avg* e *score avg* apresentaram os melhores valores de estabilidade com 92%, seguidos pelo método *score dev* com 85%.

Resumidamente, o método de aprendizado ativo *conf avg* não apresentou os melhores valores de estabilidade para nenhum cenário; o método *rank avg* apresentou o melhor valor de estabilidade em 2/4 dos cenários; o método *score avg* também apresentou o melhor valor de estabilidade para 2/4 dos cenários; e o método *score dev*, proposto neste trabalho, apresentou o melhor resultado para 3/4 dos casos, ou seja, apresentou o melhor índice de estabilidade.

Na Figura 4.5 é apresentado graficamente o *ranking* médio, em relação à medida *FV*, de cada método com teste de Friedman e pós-teste de *Nemenyi* com nível de significância de 95% para identificação de diferenças estatisticamente significativas entre os métodos (Demšar, 2006).

Embora existam diferenças consistentes entre os métodos de aprendizado ativo e o *baseline random*, nenhum método ativo apresentou melhora com diferença estatisticamente significativa em relação ao método *random*. A única diferença observada é em relação aos métodos *score dev* e *conf avg* para $FRI = 1$ e *Micro-F1*, o que indica novamente a dificuldade de *conf avg* em realizar aprendizado ativo.

Os resultados referente à medida *AULC* são apresentandos na Tabela 4.10 de maneira análoga aos resultados de *FV* apresentados na Tabela 4.9.

Pode ser observado que os métodos *rank avg*, *score avg* e *score dev* apre-

Figura 4.5: Ranking de Friedman com pós-teste de Nemenyi. Classificador $BR(SVM)$. Medida de avaliação FV . Protocolo de experimento *independente*.

sentaram melhores *ranking* médios que *random* para todos os quatro cenários de variação de FRI e medida de avaliação, enquanto que *conf avg* apresenta pior *ranking* médio que *random* para a medida de avaliação *Micro-F* com $FRI = 1$ e $FRI = 5$.

Quanto à estabilidade (*melhor ou igual random*), um comportamento semelhante ao observado para a medida FV também foi constatado para $AULC$. Para $FRI = 1$, o método *score dev* foi o único a ser melhor ou igual ao método *random* em todos os conjuntos de dados, tanto para a medida *Micro-F* quanto para *Macro-F*. Para o cenário com $FRI = 5$ e *Micro-F*, o método *score dev* apresentou a melhor estabilidade (85%). O método *score avg* foi superior para o cenário com $FRI = 5$ e *Macro-F*, no qual foi melhor ou igual ao método *random* em 92% dos conjuntos de dados. Os outros três métodos apresentaram estabilidade de 85%.

Na Figura 4.6 é apresentado graficamente o *ranking* médio, em relação à medida $AULC$, de cada método. Novamente, nenhum método ativo apresentou melhora com diferença estatisticamente significativa em relação ao método *random*. As diferenças significativas para a medida $AULC$ foram encontradas somente entre *score dev* e *conf avg* no cenário $FRI = 1/Micro-F$ e entre *score avg* e *conf avg* no cenário $FRI = 1/Macro-F$.

Protocolo de experimentos *remanescente*

Os experimentos conduzidos utilizando o protocolo *remanescente* simulam aplicações que têm como objetivo rotular exemplos no contexto de inferência transdutiva, ou seja, no contexto no qual já se sabe de antemão os objetos para os quais se quer prever os rótulos. Métodos de inferência indutiva, como o método $BR(SVM)$, também poderiam ser utilizados nesse

Tabela 4.10: Valores de *AULC* para cada método de aprendizado ativo com o protocolo *independente*. Os valores entre parênteses se referem à posição no *ranking* do método.

FRI=1										
	Micro-F					Macro-F				
	<i>random</i>	<i>conf avg</i>	<i>rank avg</i>	<i>score avg</i>	<i>score dev</i>	<i>random</i>	<i>conf avg</i>	<i>rank avg</i>	<i>score avg</i>	<i>score dev</i>
#1 <i>bibtex</i>	.14 (2.5)	.12 (4.5)	.14 (2.5)	.12 (4.5)	.17 (1.0)	.02 (3.0)	.02 (3.0)	.02 (3.0)	.02 (3.0)	.02 (3.0)
#2 <i>cal500</i>	.35 (5.0)	.36 (3.0)	.36 (3.0)	.37 (1.0)	.36 (3.0)	.20 (4.5)	.20 (4.5)	.21 (2.0)	.21 (2.0)	.21 (2.0)
#3 <i>corel16k</i>	.03 (2.5)	.02 (5.0)	.03 (2.5)	.03 (2.5)	.03 (2.5)	.01 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)
#4 <i>corel5k</i>	.03 (4.0)	.03 (4.0)	.04 (1.5)	.04 (1.5)	.03 (4.0)	.31 (3.0)	.31 (3.0)	.31 (3.0)	.31 (3.0)	.31 (3.0)
#5 <i>emotions</i>	.54 (3.5)	.54 (3.5)	.57 (1.5)	.52 (5.0)	.57 (1.5)	.49 (3.0)	.47 (4.5)	.52 (1.5)	.47 (4.5)	.52 (1.5)
#6 <i>enron</i>	.43 (4.0)	.32 (5.0)	.45 (2.0)	.49 (1.0)	.44 (3.0)	.25 (3.0)	.23 (5.0)	.25 (3.0)	.26 (1.0)	.25 (3.0)
#7 <i>llog</i>	.03 (3.0)	.04 (1.5)	.02 (4.5)	.02 (4.5)	.04 (1.5)	.39 (3.0)	.39 (3.0)	.39 (3.0)	.39 (3.0)	.39 (3.0)
#8 <i>medical</i>	.47 (4.0)	.45 (5.0)	.51 (2.5)	.53 (1.0)	.51 (2.5)	.54 (4.5)	.54 (4.5)	.56 (2.0)	.57 (1.0)	.55 (3.0)
#9 <i>ohsumed</i>	.09 (4.0)	.07 (5.0)	.10 (3.0)	.17 (1.0)	.14 (2.0)	.03 (4.0)	.02 (5.0)	.04 (2.5)	.05 (1.0)	.04 (2.5)
#10 <i>scene</i>	.46 (3.5)	.38 (5.0)	.48 (2.0)	.49 (1.0)	.46 (3.5)	.44 (3.5)	.34 (5.0)	.47 (1.0)	.46 (2.0)	.44 (3.5)
#11 <i>slashdot</i>	.06 (4.0)	.18 (1.0)	.07 (3.0)	.05 (5.0)	.13 (2.0)	.18 (4.0)	.20 (1.0)	.18 (4.0)	.18 (4.0)	.19 (2.0)
#12 <i>tmc2007</i>	.46 (3.0)	.42 (5.0)	.43 (4.0)	.50 (1.5)	.50 (1.5)	.15 (2.5)	.11 (5.0)	.13 (4.0)	.23 (1.0)	.15 (2.5)
#13 <i>yeast</i>	.56 (3.5)	.53 (5.0)	.58 (1.0)	.56 (3.5)	.57 (2.0)	.29 (3.5)	.26 (5.0)	.30 (2.0)	.31 (1.0)	.29 (3.5)
<i>ranking médio</i>	3.4	4.0	2.6	2.3	2.7	3.3	3.2	3.0	2.5	3.0
<i>melhor/igual random</i>	—	38%	85%	69%	100%	—	54%	92%	92%	100%

FRI=5										
	Micro-F					Macro-F				
	<i>random</i>	<i>conf avg</i>	<i>rank avg</i>	<i>score avg</i>	<i>score dev</i>	<i>random</i>	<i>conf avg</i>	<i>rank avg</i>	<i>score avg</i>	<i>score dev</i>
#1 <i>bibtex</i>	.19 (3.0)	.19 (3.0)	.19 (3.0)	.18 (5.0)	.20 (1.0)	.03 (3.5)	.04 (1.0)	.03 (3.5)	.03 (3.5)	.03 (3.5)
#2 <i>cal500</i>	.34 (4.5)	.34 (4.5)	.35 (2.5)	.36 (1.0)	.35 (2.5)	.19 (4.5)	.19 (4.5)	.20 (2.0)	.20 (2.0)	.20 (2.0)
#3 <i>corel16k</i>	.06 (2.5)	.05 (5.0)	.06 (2.5)	.06 (2.5)	.06 (2.5)	.01 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)	.01 (3.0)
#4 <i>corel5k</i>	.08 (3.0)	.08 (3.0)	.08 (3.0)	.08 (3.0)	.08 (3.0)	.32 (3.0)	.32 (3.0)	.32 (3.0)	.32 (3.0)	.32 (3.0)
#5 <i>emotions</i>	.56 (3.5)	.57 (1.5)	.55 (5.0)	.56 (3.5)	.57 (1.5)	.52 (3.5)	.54 (2.0)	.51 (5.0)	.52 (3.5)	.55 (1.0)
#6 <i>enron</i>	.48 (4.5)	.48 (4.5)	.49 (2.5)	.51 (1.0)	.49 (2.5)	.27 (3.5)	.27 (3.5)	.27 (3.5)	.28 (1.0)	.27 (3.5)
#7 <i>llog</i>	.06 (3.0)	.06 (3.0)	.06 (3.0)	.05 (5.0)	.07 (1.0)	.39 (3.0)	.39 (3.0)	.39 (3.0)	.39 (3.0)	.39 (3.0)
#8 <i>medical</i>	.67 (4.5)	.68 (2.5)	.67 (4.5)	.69 (1.0)	.68 (2.5)	.62 (3.5)	.62 (3.5)	.62 (3.5)	.63 (1.0)	.62 (3.5)
#9 <i>ohsumed</i>	.15 (3.0)	.12 (5.0)	.13 (4.0)	.18 (1.0)	.16 (2.0)	.05 (3.5)	.05 (3.5)	.05 (3.5)	.06 (1.0)	.05 (3.5)
#10 <i>scene</i>	.53 (3.0)	.46 (5.0)	.55 (1.5)	.55 (1.5)	.52 (4.0)	.52 (3.0)	.45 (5.0)	.55 (1.0)	.53 (2.0)	.51 (4.0)
#11 <i>slashdot</i>	.15 (4.0)	.22 (1.5)	.18 (3.0)	.11 (5.0)	.22 (1.5)	.21 (3.5)	.22 (1.5)	.21 (3.5)	.20 (5.0)	.22 (1.5)
#12 <i>tmc2007</i>	.52 (2.0)	.51 (4.0)	.51 (4.0)	.53 (1.0)	.51 (4.0)	.20 (2.0)	.17 (5.0)	.19 (3.5)	.22 (1.0)	.19 (3.5)
#13 <i>yeast</i>	.58 (4.5)	.59 (2.0)	.59 (2.0)	.59 (2.0)	.58 (4.5)	.31 (3.5)	.31 (3.5)	.32 (1.0)	.31 (3.5)	.31 (3.5)
<i>ranking médio</i>	3.6	4.0	2.5	2.5	2.3	3.5	3.4	3.1	2.5	2.5
<i>melhor/igual random</i>	—	69%	77%	77%	85%	—	85%	85%	92%	85%

contexto. No entanto, a inferência indutiva tem como intuito resolver um problema mais geral ao necessário nesse caso.

A seguir são descritos os experimentos de aprendizado ativo para o protocolo *remanescente*, os quais foram realizados utilizando o algoritmo de inferência transdutiva para aprendizado multirrótulo *RCut(LPBHN)*.

A medida de avaliação *Ranking-Loss* foi utilizada para aferir a qualidade das predições realizadas. A explicação para a utilização dessa medida está relacionada com o paradigma de classificação baseada em *ranking* no qual o método *RCut(LPBHN)* está baseado. Métodos desse paradigma necessitam de uma estratégia de corte no *ranking* de rótulos para se obter a predição final. Não é o objetivo deste trabalho avaliar a qualidade da estratégia de corte no *ranking*. Desse modo, foi escolhida a medida *Ranking-Loss*, a qual

Figura 4.6: Ranking de Friedman com pós-teste de Nemenyi. Classificador $BR(SVM)$. Medida de avaliação $AULC$. Protocolo de experimento *independente*.

avalia a qualidade do *ranking* dos rótulos retornado pelo algoritmo. Vale lembrar que a técnica *RCut* é a estratégia mais simples de realizar o corte no *ranking*. Ela seleciona os t rótulos de melhor posição no *ranking*, onde t é o número inteiro mais próximo da cardinalidade do conjunto de treinamento. Deve ser observado que devido ao fato do número de exemplos rotulados ser reduzido, a estimativa da cardinalidade utilizando o conjunto de treinamento é pouco precisa.

Na Figura 4.7 é apresentada a curva de aprendizado para o conjunto de dados *corel5k*. As figuras referentes às curvas de aprendizado dos outros conjuntos de dados são apresentadas no Apêndice A.

Figura 4.7: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *corel5k*.

Observa-se que o método *score dev* é constantemente superior tanto ao método *random* quanto ao outro método ativo *score avg* para os dois cenários.

rios considerados. O método *score avg* não foi constantemente superior ao método passivo, principalmente para $FRI = 5$.

Nas Tabelas 4.11 e 4.12 são apresentados os resultados para o protocolo *remanescente* para todos os conjuntos de dados referentes às medidas de avaliação do aprendizado ativo *FV* e *AULC*⁷, respectivamente.

Tabela 4.11: Valores *FV* de eficácia na última iteração da curva de aprendizado para cada método de aprendizado ativo com o protocolo *remanescente*. Os valores entre parenteses se referem à posição no *ranking* do método.

	Ranking-Loss					
	FRI=1			FRI=5		
	<i>random</i>	<i>score avg</i>	<i>score dev</i>	<i>random</i>	<i>score avg</i>	<i>score dev</i>
<i>bibtex</i>	0.10(2.0)	0.16(3.0)	0.09(1.0)	0.02(1.5)	0.02(1.5)	0.03(3.0)
<i>cal500</i>	0.17(3.0)	0.15(1.5)	0.15(1.5)	0.17(3.0)	0.16(1.5)	0.16(1.5)
<i>corel16k</i>	0.14(2.5)	0.14(2.5)	0.13(1.0)	0.10(2.5)	0.10(2.5)	0.09(1.0)
<i>corel5k</i>	0.12(2.0)	0.13(3.0)	0.11(1.0)	0.07(1.5)	0.08(3.0)	0.07(1.5)
<i>emotions</i>	0.06(2.5)	0.06(2.5)	0.03(1.0)	0.06(2.5)	0.06(2.5)	0.01(1.0)
<i>enron</i>	0.15(3.0)	0.13(1.5)	0.13(1.5)	0.14(3.0)	0.12(1.0)	0.13(2.0)
<i>llog</i>	0.09(2.0)	0.09(2.0)	0.09(2.0)	0.06(3.0)	0.05(1.5)	0.05(1.5)
<i>medical</i>	0.05(2.0)	0.06(3.0)	0.04(1.0)	0.04(2.0)	0.04(2.0)	0.04(2.0)
<i>ohsumed</i>	0.15(3.0)	0.12(1.5)	0.12(1.5)	0.13(3.0)	0.10(1.5)	0.10(1.5)
<i>scene</i>	0.03(1.0)	0.16(2.5)	0.16(2.5)	0.02(2.0)	0.03(3.0)	0.01(1.0)
<i>slashdot</i>	0.05(2.0)	0.09(3.0)	0.04(1.0)	0.04(1.5)	0.05(3.0)	0.04(1.5)
<i>tmc2007</i>	0.14(3.0)	0.11(1.5)	0.11(1.5)	0.14(3.0)	0.12(1.5)	0.12(1.5)
<i>yeast</i>	0.06(1.0)	0.07(2.5)	0.07(2.5)	0.07(2.0)	0.06(1.0)	0.08(3.0)
<i>ranking médio</i>	2.2	2.3	1.5	2.3	2.0	1.7
<i>melhor/igual random</i>	—	54%	85%	—	77%	85%

Tabela 4.12: Valores de *AULC* para cada método de aprendizado ativo com o protocolo *remanescente*. Os valores entre parenteses se referem à posição no *ranking* do método.

	Ranking-Loss					
	FRI=1			FRI=5		
	<i>random</i>	<i>score avg</i>	<i>score dev</i>	<i>random</i>	<i>score avg</i>	<i>score dev</i>
<i>bibtex</i>	1.73 (2.0)	1.95 (3.0)	1.70 (1.0)	0.28 (1.5)	0.28 (1.5)	0.29 (3.0)
<i>cal500</i>	1.90 (3.0)	1.81 (2.0)	1.80 (1.0)	1.75 (3.0)	1.69 (1.5)	1.69 (1.5)
<i>corel16k</i>	1.86 (2.0)	1.89 (3.0)	1.78 (1.0)	1.04 (3.0)	1.03 (2.0)	0.98 (1.0)
<i>corel5k</i>	1.40 (3.0)	1.37 (2.0)	1.31 (1.0)	0.76 (2.0)	0.79 (3.0)	0.73 (1.0)
<i>emotions</i>	1.05 (1.0)	1.32 (3.0)	1.28 (2.0)	0.72 (2.0)	0.83 (3.0)	0.39 (1.0)
<i>enron</i>	1.59 (2.5)	1.59 (2.5)	1.51 (1.0)	1.44 (3.0)	1.30 (1.0)	1.32 (2.0)
<i>llog</i>	1.18 (3.0)	1.17 (2.0)	1.09 (1.0)	0.64 (3.0)	0.61 (2.0)	0.59 (1.0)
<i>medical</i>	0.98 (3.0)	0.96 (2.0)	0.87 (1.0)	0.48 (3.0)	0.44 (2.0)	0.42 (1.0)
<i>ohsumed</i>	2.11 (3.0)	1.77 (2.0)	1.75 (1.0)	1.49 (3.0)	1.31 (1.0)	1.32 (2.0)
<i>scene</i>	0.87 (1.0)	1.94 (3.0)	1.92 (2.0)	0.39 (2.0)	0.40 (3.0)	0.28 (1.0)
<i>slashdot</i>	0.86 (2.0)	1.20 (3.0)	0.59 (1.0)	0.48 (2.0)	0.50 (3.0)	0.46 (1.0)
<i>tmc2007</i>	1.67 (3.0)	1.41 (2.0)	1.39 (1.0)	1.52 (3.0)	1.31 (2.0)	1.29 (1.0)
<i>yeast</i>	0.84 (3.0)	0.82 (2.0)	0.72 (1.0)	0.80 (1.0)	0.83 (2.0)	0.89 (3.0)
<i>ranking médio</i>	2.4	2.4	1.2	2.4	2.1	1.5
<i>melhor/igual random</i>	—	62%	85%	—	62%	85%

O método *score dev* apresentou os melhores *ranking* médios e a melhor estabilidade (melhor ou igual a *random*) para todos os cenários (85%). Para

⁷Os valores de *AULC* para a medida *Ranking-Loss* foram multiplicados por 10 para considerar a terceira casa decimal na comparação.

o protocolo *remanescente*, o método *score avg* apresentou estabilidade inferior à apresentada por ele utilizando o protocolo *independente*.

Na Figura 4.8 é apresentado o *ranking* médio de cada método de maneira gráfica juntamente com o teste de significância estatística realizado. O método *score dev* apresentou os melhores valores de *ranking* médio para todos os cenários, com diferença estatisticamente significativa para o método *random* quando considerada a medida *AULC*. Também é possível observar diferença estatisticamente significativa de *score dev* para o método *score avg* quando considerada a medida *AULC* e $FRI = 1$.

Figura 4.8: Ranking de Friedman com pós-teste de Nemenyi. Classificador *RCut(LPBHN)*. Medidas de avaliação *AULC* e *FV*. Protocolo de experimento *remanescente*.

4.4. Considerações finais

Neste capítulo foram apresentados conceitos básicos de aprendizado ativo com foco no contexto multirrotulo. Além disso, algumas contribuições para essa área de pesquisa foram apresentadas, das quais se destacam as seguintes:

Análise dos custos de rotulação manual. Quando é fornecido ao especialista um lote de objetos a serem rotulados, duas abordagens podem ser utilizadas para a realização da tarefa: *object-wise* e *label-wise*. Neste capítulo foi apresentada uma análise dos custos incidentes sobre essas abordagens de rotulação manual. Sugere-se a abordagem *object-wise* como a mais adequada para a maioria dos casos, exceto para a rotulação de imagens, na qual a abordagem *label-wise* apresenta menor custo de rotulação manual.

Análise de rotulação completa e parcial de objeto. Como extensão da análise das abordagens de rotulação, outra possibilidade é a rotulação parcial de exemplos. Apesar de parecer válido o esforço para elaboração de algoritmos de aprendizado ativo multirrótulo que considerem exemplos parcialmente rotulados, essa abordagem só faria sentido quando se deseja trabalhar com imagens, pois o custo de entendimento de objetos de outros tipos de dados poderia ser proibitivo. Desse modo, sugere-se utilizar a rotulação completa de objetos sempre que possível.

Composição do conjunto inicial de exemplos rotulados. O algoritmo de aprendizado ativo necessita de um conjunto inicial de exemplos rotulados para realizar sua tarefa. A estratégia de composição desse conjunto influencia na qualidade das consultas realizadas pelo algoritmo de aprendizado ativo e, conseqüentemente, na avaliação experimental desse tipo de método. Neste capítulo foi introduzida uma abordagem para composição de conjunto inicial de exemplos rotulados utilizando um Fator de Rotulação Inicial. Consideramos essa abordagem mais adequada que as outras utilizadas até então na literatura, pois ela considerada a única característica do problema multirrótulo disponível anteriormente ao processo de rotulação: a quantidade de rótulos. Assim, sua utilização torna a comparação dos resultados de diferentes conjuntos mais adequada. Além disso, a abordagem permite o controle sobre a quantidade de exemplos a serem rotulados, o que é importante para não tornar custoso o processo de rotulação antes mesmo de se iniciar o aprendizado ativo.

Algoritmo transdutivo para protocolo *remanescente*. Neste trabalho foi considerado um algoritmo transdutivo para a realização de aprendizado ativo utilizando o protocolo de experimentos *remanescente*. Mesmo considerando essa a melhor abordagem nesse cenário experimental, não é de nosso conhecimento outro trabalho que a tenha considerada.

Método de aprendizado ativo multirrótulo. O método *score dev* para aprendizado ativo multirrótulo foi proposto neste capítulo. Esse método apresenta resultados consistentemente superiores, ou no mínimo iguais, ao método *random* para a grande maioria dos conjuntos de dados utilizados. Isso representa a possibilidade de redução dos custos do processo de rotulação no contexto multirrótulo. Além disso, o método apresentou resultados competitivos aos outros métodos de aprendizado ativo tradicionais da literatura, principalmente no protocolo *remanescente*, na qual uma melhora em relação ao *random* com diferença estatisticamente significativa foi constatada. Desse modo, sugere-se

sua utilização quando se deseja reduzir os custos do processo de rotulação de exemplos multirrótulo.

Conclusões

NESTE capítulo são apresentadas as conclusões decorrentes deste trabalho de doutorado. Os objetivos do trabalho são retomados e relacionados com as contribuições realizadas durante seu desenvolvimento. Posteriormente, as limitações e os trabalhos futuros são apresentados.

O objetivo geral deste trabalho é contribuir para o desenvolvimento da área de aprendizado multirrótulo. Especificamente, dentre os desafios abordados neste trabalho, destacam-se as contribuições apresentadas em diferentes capítulos e relacionadas aos seguintes desafios:

Capítulo 2: Identificar uma referência de aprendizado mínimo para o aprendizado multirrótulo;

Capítulo 3: Identificar e explorar a dependência de rótulos no aprendizado multirrótulo;

Capítulo 4: Reduzir o custo do processo de rotulação manual de exemplos multirrotulados.

5.1. Classificadores *baseline* – Capítulo 2

No Capítulo 2 foram apresentados os conceitos básicos referente ao aprendizado multirrótulo. Mais especificamente, foram apresentadas algumas

definições, abordagens para a avaliação do aprendizado e para caracterização dos problemas multirrótulo, os principais métodos de aprendizado multirrótulo, algumas aplicações que utilizam essa abordagem, bem como os conjuntos de dados utilizados no trabalho.

Algumas contribuições deste trabalho foram apresentadas nesse capítulo, dentre elas se destacam os métodos baseados em grafo de aprendizado multirrótulo (Cherman et al., 2013a) e os classificadores *baseline* (Metz et al., 2012; Spolaôr et al., 2013a; Metz et al., 2013).

Classificadores considerados *baseline* são importantes referências para identificar a eficácia de novas propostas. O erro majoritário, por exemplo, é uma medida em aprendizado monorrótulo fornecida por um classificador *baseline* que sempre prediz a classe majoritária do conjunto de dados, ou seja, que não considera os atributos descritores para fornecer sua predição. Novas propostas de algoritmos de aprendizado podem utilizar o erro majoritário como uma referência mínima de aprendizado a ser alcançada. Apesar de não ser obrigatório ser superior ao *baseline*, casos em que um classificador não consegue superá-lo devem ser explicados e, ao menos, apresentar resultados superiores em aspectos diferentes daqueles estritamente relacionados ao *baseline*.

Nesse sentido, as contribuições relacionadas aos classificadores *baseline* específicos para o aprendizado multirrótulo são descritas as seguir, referenciando as perguntas inicialmente levantadas quanto ao desafio de pesquisa correspondente.

5.1.1 Contribuições

Com intuito de contribuir para a identificação de referência *baseline* de aprendizado multirrótulo, as seguintes perguntas nortearam este trabalho:

- ***Qual abordagem baseline é adequada para o aprendizado multirrótulo?***

Classificadores *baseline* específicos foram propostos e avaliados neste trabalho para as medidas de avaliação *Hamming-Loss*, *Subset-Accuracy* e *F-Measure*. No entanto, o aprendizado multirrótulo pode ser avaliado pela perspectiva de diversas medidas de avaliação. Enquanto um *baseline* específico maximiza/minimiza uma medida de avaliação, as outras medidas podem sofrer deterioração. Esse tipo de abordagem é adequado em casos em que se sabe qual medida de avaliação o método multirrótulo maximiza/minimiza. Para outros casos, um *baseline* geral, chamado $Geral_B$, foi proposto.

- **Como os resultados reportados na literatura se situam em relação aos resultados obtidos por um baseline?**

Por meio de uma revisão sistemática da literatura, os resultados do $Geral_B$ em dez conjuntos de dados frequentemente utilizados pela comunidade foram comparados aos resultados reportados na literatura por 64 trabalhos nesses conjuntos de dados. De um total de 5342 valores de medidas coletadas, constatou-se que 12,8% não foram superiores aos valores das respectivas medidas obtidas pelo *baseline* $Geral_B$. No caso extremo, para o conjunto de dados *corel5k*, 43% dos resultados reportados foram piores ou iguais aos resultados obtidos pelo $Geral_B$. Na outra ponta, apenas 0,6% dos resultados do conjunto de dados *emotions* não foram melhores que os apresentados pelo *baseline*.

5.1.2 Limitações e trabalhos futuros

Nos casos em que se sabe antecipadamente qual é a medida de avaliação que se deseja maximizar/minimizar, um classificador *baseline* específico poderia ser utilizado de fato como referência mínima de aprendizado, ou seja, nessa situação, pode-se considerar que qualquer classificador pior que o *baseline* específico não é capaz de aprender por meio dos atributos descritores os conceitos relacionados aos rótulos. No entanto, esse não é o caso do $Geral_B$, o qual deve ser utilizado com parcimônia. Resultados inferiores ao $Geral_B$ não indicam necessariamente que o algoritmo de aprendizado avaliado não foi capaz de realizar sua tarefa adequadamente. Essa situação é um indicativo de alerta, ou seja, merece maior atenção e explicação dos pesquisadores quanto a esses resultados anômalos obtidos pelo algoritmo.

Como trabalhos futuros, pretende-se ampliar a revisão sistemática e disponibilizar os resultados de maneira estruturada para que se possa responder a diversas perguntas, tais como: *Quais trabalhos na literatura utilizaram conjuntos de dados com cardinalidade maior que A e um número de multirrótulos distintos maior que B? Quais trabalhos conseguiram resultados pelo menos 10% superiores aos apresentados pelo BR em relação à medida F-Measure?*

5.2. Dependência de Rótulos – Capítulo 3

A identificação e a exploração da dependência de rótulos é um desafio importante que tem despertado interesse de vários pesquisadores da área de aprendizado multirrótulo. Há um consenso na comunidade de que os métodos de aprendizado multirrótulo podem produzir melhores modelos

preditivos quando a dependência de rótulos é considerada.

5.2.1 Contribuições

No Capítulo 3, foram apresentados as abordagens *UBC* e *BR+* para lidar com esses desafios e contribuir com a resposta das seguintes perguntas:

- ***Qual é a influência da dependência de rótulos no aprendizado multirrótulo? Quais são os limites da consideração da dependência de rótulos?***

A resposta dessa pergunta passa por duas etapas: 1) mensurar a presença de dependência de rótulos no conjunto de dados; e 2) relacionar as medidas coletadas com métricas de qualidade do aprendizado.

Diversas medidas podem mensurar a presença de dependência de rótulos em conjunto de dados multirrótulo, como medidas de correlação e co-ocorrência. No entanto, não é trivial a tarefa de relacionar a informação retornada por essas medidas com o benefício em termos de eficácia preditiva no aprendizado multirrótulo. O método *Upper Bound Classifier (UBC)*, proposto neste trabalho, realiza essas duas etapas (Cherman et al., 2012). O *UBC* gera um modelo preditivo multirrótulo que identifica os padrões de dependência de rótulos por meio do algoritmo base de aprendizado. Posteriormente, é realizada uma simulação do cenário em que toda a informação de dependência de rótulo está disponível no momento da predição multirrótulo pelo modelo criado. Dessa maneira, o *UBC* consegue medir a influência da dependência de rótulos em termos de medidas de avaliação de aprendizado multirrótulo. As medidas retornadas pela avaliação do *UBC* indicam o limite da consideração da dependência de rótulos, pelo menos na visão do algoritmo base utilizado. Além disso, em conjunto com o método *BR*, o *UBC* pode indicar a margem disponível em um conjunto de dados para a exploração da dependência de rótulos. Pesquisadores podem utilizar o *UBC* para entender a dependência de rótulos de domínios de aplicações específicos e, assim, desenvolver métodos para melhorar a qualidade preditiva nesses domínios. Por exemplo, o *UBC* pode indicar que não existe margem para a exploração da dependência de rótulos em um conjunto de dados e que o esforço do pesquisador nesse domínio deve ser focado no desenvolvimento de métodos que considerem outros aspectos do aprendizado. Além disso, o modelo gerado pelo *UBC* pode ser efetivamente utilizado para a própria exploração da dependência de rótulos, como descrito a seguir.

- **Como explorar essa característica para melhorar a eficácia preditiva dos classificadores multirrótulo?**

O modelo criado pelo *UBC* pode ser utilizado para explorar a dependência de rótulos. Para tanto, o método de aprendizado multirrótulo *BR+* foi proposto neste trabalho (Cherman et al., 2012, 2010b). O *BR+* utiliza previsões de outros classificadores multirrótulo como entrada no modelo criado pelo *UBC* para explorar a dependência de rótulos nos conjuntos de dados. Esse método apresentou resultados superiores a outras abordagens tradicionais, como o *CC*: um método que tem se destacado na comunidade e é considerado estado da arte nesse quesito. Além disso, uma característica importante do *BR+* identificada neste trabalho é a sua capacidade de melhorar as previsões a ele fornecidas. Desse modo, o *BR+* pode usufruir de resultados ainda melhores quando novas abordagens propostas fornecerem a ele melhores previsões iniciais.

5.2.2 Limitações e trabalhos futuros

O poder de identificação da dependência de rótulos do *UBC* está limitado à visão do algoritmo base e à representatividade do conjunto de dados utilizados. Essa é a mesma limitação apresentada pelos algoritmos de aprendizado de máquina supervisionado e é devida ao próprio *bias* do método. O método *BR+*, por sua vez, também apresenta essas limitações. Além disso, esse método também apresenta uma maior complexidade computacional devido à dependência das previsões de outro classificador multirrótulo, apesar de que essa maior complexidade pode ser justificada em diversas aplicações que necessitam ganhos em qualidade preditiva.

Como trabalhos futuros, pretende-se avaliar o *BR+* utilizando uma gama maior de medidas de avaliação para identificar outros cenários em que esse método é capaz de melhorar a qualidade preditiva das previsões iniciais a ele fornecidas. Além disso, devido a essa característica, pode ser interessante retro-alimentar o *BR+* com suas próprias previsões e avaliar a capacidade de mesmo assim melhorá-las. Outro aspecto que poderia contribuir para melhor identificação da dependência de rótulos, tanto para *UBC* quanto para o *BR+*, seria a utilização de um comitê de classificadores como base. No aspecto de escalabilidade, pretende-se implementar os métodos *UBC* e *BR+* em uma plataforma de paralelização. Dessa forma será possível aplicá-los a uma gama maior de problemas.

5.3. Aprendizado Ativo Multirrótulo – Capítulo 4

A obtenção de um conjunto de exemplos rotulado representativo é um processo tipicamente manual e custoso. Esse custo é ainda mais acentuado em aprendizado multirrótulo pela necessidade de mapear múltiplos rótulos aos exemplos.

5.3.1 Contribuições

No Capítulo 4, as propostas relacionadas a esse desafio foram apresentadas (Cherman et al., 2013b), entre elas se destacam as seguintes contribuições às perguntas inicialmente levantadas:

- ***Como o anotador/rotulador humano deve realizar sua tarefa para minimizar o esforço necessário para concluí-la?***

A abordagem *object-wise* é a mais adequada para realizar essa tarefa para maioria dos problemas. Nessa abordagem, o foco de rotulação é dado para cada objeto até identificar a presença/ausência de todos os rótulos do domínio. A única exceção se dá para problemas que contêm objetos que são mais fáceis de compreender que os próprios conceitos dos rótulos. Esse pode ser o caso da anotação de imagens, pois dados desse tipo são geralmente de rápido entendimento por humanos. Além disso, a rotulação de cada objeto poderia ser realizada parcialmente no aprendizado multirrótulo. No entanto, sugere-se que essa estratégia seja evitada sempre que possível, pois o custo de entendimento de um objeto parcialmente rotulado seria incorrido novamente caso seja necessário rotulá-lo por completo.

- ***Métodos de aprendizado ativo multirrótulo permitem reduzir a quantidade de exemplos a serem rotulados sem perder qualidade no modelo preditivo?***

A abordagem geralmente utilizada no processo de rotulação consiste em selecionar aleatoriamente os objetos a serem processados pelo rotulador. Selecionar objetos representativos a serem rotulados utilizando métodos de aprendizado ativo pode reduzir a quantidade necessária de exemplos para se obter um bom modelo preditivo no aprendizado multirrótulo. Alguns métodos de aprendizado ativo multirrótulo avaliados neste trabalho conseguiram realizar essa tarefa com êxito para diversos casos. No entanto, sugere-se especificamente a utilização do método *score dev*, proposto neste trabalho, por ter obtido resultados superiores à abordagem aleatória de maneira mais consistente.

Neste trabalho, um algoritmo de inferência transdutiva foi utilizado como algoritmo base na simulação de problemas em que o conjunto de exemplos a ser automaticamente rotulado é conhecido de antemão. Essa abordagem é recomendada para melhorar a eficácia do aprendizado ativo nesses casos, pois é possível fornecer mais informação ao algoritmo. Não é de nosso conhecimento outro trabalho em aprendizado ativo multirrótulo que tenha avaliado essa abordagem.

- **Como avaliar os métodos de aprendizado ativo no cenário multirrótulo?**

Com as devidas adaptações, a maioria dos métodos de avaliação experimental de aprendizado ativo monorrótulo pode ser aplicado no contexto multirrótulo. No entanto, alguns aspectos necessitam de soluções específicas. A composição do conjunto inicialmente rotulado, aquele utilizado antes de se iniciar o processo de aprendizado ativo propriamente dito, é um exemplo desses aspectos específicos. Para esse caso, sugere-se a utilização da abordagem baseada no *Fator de Rotulação Inicial*, proposta neste trabalho. Essa abordagem permite controle da quantidade máxima de objetos a serem inicialmente rotulados e a comparação entre conjunto de dados dos resultados experimentais.

5.3.2 Limitações e trabalhos futuros

A avaliação de métodos de aprendizado ativo envolve diversas variáveis experimentais, tais como tamanho do conjunto inicial de exemplos rotulados e o critério de parada do processo de rotulação. Neste trabalho foram consideradas duas possibilidades de tamanho de conjunto inicial de exemplos rotulados. No entanto, um estudo mais detalhado sobre essa variável é necessário para identificar padrões em diferentes conjuntos de dados. Além disso, para simplificar a avaliação experimental, também foi considerado um número fixo de iterações no processo de aprendizado ativo. Um estudo sobre essas variáveis é importante para conclusões mais detalhadas sobre o comportamento dos métodos de aprendizado ativo multirrótulo. Desse modo, pretende-se investigar essas variáveis experimentais em trabalhos futuros.

Todas as implementações desenvolvidas neste trabalho estão publicamente disponíveis em <http://www.labic.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/>.

Apêndice - Experimentos com Aprendizado Ativo Multirrótulo

A.1. Protocolo de experimentos utilizando conjunto *independente* de teste

Figura A.1: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *bibtex*.

cal500

Figura A.2: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *cal500*.

corel16k001

Figura A.3: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *corel16k001*.

Figura A.4: Algoritmo de aprendizaje *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *corel5k*.

Figura A.5: Algoritmo de aprendizaje *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *emotions*.

Figura A.6: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *enron*.

Figura A.7: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *llog-f*.

Figura A.8: Algoritmo de aprendizaje *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *medical*.

Figura A.9: Algoritmo de aprendizaje *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *ohsumed-f*.

Figura A.10: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *scene*.

Figura A.11: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *slashdot-f*.

Figura A.12: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *tmc2007-500*.

Figura A.13: Algoritmo de aprendizado *LIBLinear*. Protocolo de experimento *independente*. Conjunto de dados *yeast*.

A.2. Protocolo de experimentos utilizando conjunto *remanescente* de teste

Figura A.14: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *cal500*.

Figura A.15: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *corel16k001*.

Figura A.16: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *corel5k*.

Figura A.17: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *emotions*.

Figura A.18: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *enron*.

Figura A.19: Algoritmo de aprendizaje *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *llog-f*.

Figura A.20: Algoritmo de aprendizaje *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *medical*.

Figura A.21: Algoritmo de aprendizaje *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *ohsumed-f*.

Figura A.22: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *scene*.

Figura A.23: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *slashdot-f*.

Figura A.24: Algoritmo de aprendizado *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *tmc2007-500*.

Figura A.25: Algoritmo de aprendizaje *LPBHN-Rcut*. Protocolo de experimento *remanescente*. Conjunto de dados *yeast*.

Referências Bibliográficas

- Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press, 2ª edição. Citado na página [1](#).
- Blokeel, H., Schietgat, L., Struyf, J., Džeroski, S., e Clare, A. (2006). Decision trees for hierarchical multilabel classification: A case study in functional genomics. *Knowledge Discovery in Databases: PKDD 2006*, páginas 18–29. Citado na página [11](#).
- Bloodgood, M. e Vijay-Shanker, K. (2009). A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, páginas 39–47, Stroudsburg, PA, USA. Association for Computational Linguistics. Citado na página [86](#).
- Boutell, M. R., Luo, J., Shen, X., e Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37:1757–1771. Citado nas páginas [11](#), [40](#), e [45](#).
- Brinker, K. (2006). On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*, Studies in Classification, Data Analysis, and Knowledge Organization, páginas 206–213. Springer Berlin Heidelberg. Citado nas páginas [76](#), [80](#), [83](#), [85](#), e [87](#).
- Brinker, K. e Hüllermeier, E. (2007). Case-based multilabel ranking. In *Proceedings of the 20th International Conference on Artificial Intelligence (IJCAI '07)*, páginas 702–707, Hyderabad - India. Citado na página [19](#).
- Brito, M. R., Chávez, E. L., Quiroz, A. J., e Yukich, J. E. (1997). Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, 35(1):33–42. Citado na página [20](#).

- Carvalho, A. D. e Freitas, A. (2009). *A tutorial on multi-label classification techniques*, volume 5 of *Studies in Computational Intelligence* 205, páginas 177–195. Springer. Citado na página 2.
- Chekina, L., Gutfreund, D., Kontorovich, A., Rokach, L., e Shapira, B. (2013). Exploiting label dependencies for improved sample complexity. *Machine Learning*, 91(1):1–42. Citado na página 44.
- Cherman, E. A., Metz, J., e Monard, M. C. (2010a). Explorando dependência entre rótulos no classificador multirrótulo binary relevance. In *Proceedings of III Workshop on Computational Intelligence in Joint Conference - SBIA/SBRN/JRI 2010*, páginas 1–6, São Bernardo do Campo, Brasil. Publicado em CD-ROM. Citado nas páginas 7 e 59.
- Cherman, E. A., Metz, J., e Monard, M. C. (2010b). A simple approach to incorporate label dependency in multi-label classification. In *Advances in Soft Computing*, volume 6438 of *Lecture Notes in Computer Science*, páginas 33–43. Springer. Citado nas páginas 7, 59, e 105.
- Cherman, E. A., Metz, J., e Monard, M. C. (2012). Incorporating label dependency into the binary relevance framework for multi-label classification. *Expert Systems with Applications*, 39(2):1647–1655. Citado nas páginas 7, 44, 45, 59, 104, e 105.
- Cherman, E. A., Monard, M. C., e Metz, J. (2010c). Métodos multirrótulo independentes de algoritmo: um estudo de caso. In *Anais do XXXVI Conferência Latino-americana de Informática*, páginas 1–14, Asuncion, Paraguay. Publicado em CD-ROM. Citado na página 7.
- Cherman, E. A., Monard, M. C., e Metz, J. (2011). Multi-label problem transformation methods: a case study. *CLEI Electronic Journal*, 14(1):1–10. Citado na página 7.
- Cherman, E. A., Spolaôr, N., Valverde-Rebaza, J. C., e Monard, M. C. (2013a). Algoritmos de aprendizado baseado em grafos para classificação multirrótulo. In *Anais do X Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, páginas 1–12, Fortaleza - Brasil. Citado nas páginas 7, 19, 20, 21, e 102.
- Cherman, E. A., Tsoumakas, G., e Monard, M. C. (2013b). Active learning approaches for multi-label data. *A ser submetido*. Citado nas páginas 7 e 106.

- Clare, A. e King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In *PKDD '01: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, páginas 42–53, London, UK. Springer-Verlag. Citado nas páginas [11](#), [19](#), e [45](#).
- Crammer, K., Singer, Y., K, J., Hofmann, T., Poggio, T., e Shawe-taylor, J. (2003). A family of additive online algorithms for category ranking. *Journal of Machine Learning Research*, 3:1025–1058. Citado nas páginas [19](#) e [22](#).
- Dembczynski, K., Waegeman, W., Cheng, W., e Hüllermeier, E. (2010). On label dependence in multi-label classification. In *Proceedings of 2nd International Workshop on Learning from Multi-Label Data (MLD'10) in International Conference of Machine Learning (ICML)*, páginas 5–12, Haifa - Israel. Citado na página [44](#).
- Dembczyński, K., Waegeman, W., Cheng, W., e Hüllermeier, E. (2012). On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45. Citado nas páginas [45](#) e [46](#).
- Demšar, J. (2006). Statistical comparison of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30. Citado nas páginas [60](#) e [93](#).
- Elisseeff, A. e Weston, J. (2001). A kernel method for multi-labelled classification. In *In Advances in Neural Information Processing Systems*, páginas 681–687. MIT Press. Citado nas páginas [40](#) e [45](#).
- Esuli, A. e Sebastiani, F. (2009). Active learning strategies for multi-label text classification. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, ECIR '09*, páginas 102–113, Berlin, Heidelberg. Springer-Verlag. Citado nas páginas [72](#), [76](#), [77](#), [78](#), [80](#), [81](#), [83](#), [85](#), e [87](#).
- Flach, P. (2012). *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press. Citado nas páginas [1](#), [26](#), e [31](#).
- Freund, Y. e Schapire, R. E. (1995). A decision-theoretic generalization of on-Line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139. Citado na página [19](#).
- Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., e Brinker, K. (2008). Multi-label classification via calibrated label ranking. *Mach. Learn.*, 73(2):133–153. Citado na página [45](#).

- Godbole, S. e Sarawagi, S. (2004). Discriminative Methods for Multi-labeled Classification. *Advances in Knowledge Discovery and Data Mining*, páginas 22–30. Citado nas páginas [12](#), [17](#), e [45](#).
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, 2ª edição. Citado na página [22](#).
- Hung, C.-W. e Lin, H.-T. (2011). Multi-label active learning with auxiliary learner. *Journal of Machine Learning Research - Proceedings Track*, páginas 315–332. Citado na página [79](#).
- Li, T. e Ogihara, M. (2003). Detecting emotion in music. In *Proceedings of the International Symposium on Music Information Retrieval*, páginas 239–240, Washington D.C., USA. Citado na página [11](#).
- Li, T., Zhang, C., e Zhu, S. (2006). Empirical studies on multi-label classification. *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, páginas 86–92. Citado na página [11](#).
- Liu, H., Zhang, S., Zhao, J., Zhao, X., e Mo, Y. (2010). A new classification algorithm using mutual nearest neighbors. In *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*, páginas 52–57. Citado na página [21](#).
- Luo, X. e Heywood, Z. A. N. (2005). Evaluation of two systems on multi-class multi-label document classification. In *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems*, páginas 161–169. Citado na página [19](#).
- Metz, J., Abreu, L. F., Cherman, E. A., e Monard, M. C. (2012). On the estimation of predictive evaluation measure baselines for multi-label learning. In *Advances in Artificial Intelligence (IBERAMIA 2012)*, volume 7637 of *Lecture Notes in Computer Science*, páginas 189–198. Springer. Citado nas páginas [6](#), [32](#), [34](#), [35](#), e [102](#).
- Metz, J., Spolaôr, N., Cherman, E. A., e Monard, M. C. (2013). The learning power of multi-label algorithms: analyzing literature results in contrast to a simple multi-label baseline classifier. *Submetido à Expert Systems with Applications*, páginas 1–25. Citado nas páginas [6](#), [36](#), [59](#), e [102](#).
- Minsky, M. L. e Papert, S. A. (1969). *Perceptrons*. MITP. Citado na página [22](#).
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Education (ISE Editions). Citado nas páginas [1](#) e [2](#).

- Nowak, S., Nagel, K., e Liebetrau, J. (2011). The CLEF 2011 photo annotation and concept-based retrieval tasks. In *CLEF (Notebook Papers/Labs/Workshop)*, páginas 1–25. Citado na página [72](#).
- Olsson, F. e Tomanek, K. (2009). An intrinsic stopping criterion for committee-based active learning. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, páginas 138–146, Stroudsburg, PA, USA. Association for Computational Linguistics. Citado na página [86](#).
- Qi, G. J., Hua, X. S., Rui, Y., Tang, J., Mei, T., e Zhang, H. J. (2007). Correlative multi-label video annotation. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, páginas 17–26, New York, NY, USA. ACM. Citado nas páginas [11](#) e [45](#).
- Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., e Zhang, H.-J. (2009). Two-Dimensional Multilabel Active Learning with an Efficient Online Adaptation Model for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1880–1897. Citado na página [73](#).
- Quinlan, J. R. (1993). *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers, CA. Citado na página [19](#).
- Read, J. (2010). *Scalable Multi-label Classification*. Tese de Doutorado, Department of Computer Science of The University of Waikato, Hamilton, New Zealand. Citado na página [5](#).
- Read, J., Pfahringer, B., e Holmes, G. (2008). Multi-label classification using ensembles of pruned sets. In *ICDM*, páginas 995–1000. IEEE Computer Society. Citado nas páginas [12](#), [15](#), e [45](#).
- Read, J., Pfahringer, B., Holmes, G., e Frank, E. (2009). Classifier chains for multi-label classification. In Buntine, W. L., Grobelnik, M., Mladenic, D., e Shawe-Taylor, J., editors, *ECML/PKDD (2)*, volume 5782 of *Lecture Notes in Computer Science*, páginas 254–269. Springer. Citado nas páginas [12](#), [16](#), [17](#), [18](#), e [59](#).
- Read, J., Pfahringer, B., Holmes, G., e Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, 85:333–359. Citado na página [59](#).
- Reis, D. M., Cherman, E. A., Spolaôr, N., e Monard, M. C. (2012). Extensões do algoritmo de aprendizado de máquina multirrótulo BRkNN. In *Encontro Nacional de Inteligência Artificial (ENIA 2012)*, páginas 1–12. Citado na página [8](#).

- Rezende, S. O. (2003). *Sistemas Inteligentes: Fundamentos e Aplicações*. Editora Manole, Barueri, SP, Brasil. Citado na página 48.
- Rossi, R. G., de Andrade Lopes, A., e Rezende, S. O. (2013). A Parameter-Free Label Propagation Algorithm Using Bipartite Heterogeneous Networks for Text Classification. In *Proceedings of Symposium on Applied Computing (ACM SAC'2014)*. Citado na página 89.
- Schapire, R. E. e Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. In *Machine Learning*, páginas 135–168. Citado nas páginas 18 e 19.
- Settles, B. (2010). Active learning literature survey. Relatório Técnico 1648, University of Wisconsin–Madison. Citado nas páginas 69, 70, 71, e 86.
- Singh, M., Brew, A., Greene, D., e Cunningham, P. (2010). Score Normalization and Aggregation for Active Learning in Multi-label Classification. Relatório técnico, University College Dublin. Citado nas páginas 76, 78, 80, 84, 85, e 87.
- Spolaôr, N., Cherman, E. A., Metz, J., e Monard, M. C. (2013a). A systematic review on experimental multi-label learning. Relatório técnico, Instituto de Ciências Matemáticas e de Computação – ICMC/USP. Nº 392. Citado nas páginas 6, 35, e 102.
- Spolaôr, N., Cherman, E. A., Monard, M. C., e Lee, H. D. (2012). Filter approach feature selection methods to support multi-label learning based on ReliefF and information gain. In *Advances in Artificial Intelligence - SBIA 2012*, volume 7589 of *Lecture Notes in Artificial Intelligence*, páginas 72–81. Springer. Citado na página 8.
- Spolaôr, N., Cherman, E. A., Monard, M. C., e Lee, H. D. (2013b). A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic Notes in Theoretical Computer Science*, 292:135–151. Citado na página 8.
- Spolaôr, N., Cherman, E. A., Monard, M. C., e Lee, H. D. (2013c). ReliefF for multi-label feature selection. In *Brazilian Conference on Intelligent Systems (BRACIS 2013)*, páginas 1–6, Fortaleza – Brasil. Citado na página 8.
- Spyromitros, E., Tsoumakas, G., e Vlahavas, I. (2008). An empirical study of lazy multilabel classification algorithms. *Artificial Intelligence: Theories, Models and Applications*, páginas 401–406. Citado nas páginas 19 e 20.

- Tang, L., Rajan, S., e Narayanan, V. K. (2009). Large scale multi-label classification via metalabeler. In *Proceedings of the 18th international conference on World wide web, WWW '09*, páginas 211–220, New York, NY, USA. ACM. Citado nas páginas [23](#) e [25](#).
- Thabtah, F. A., Cowling, P., e Peng, Y. (2004). MMAC: A new multi-class, multi-label associative classification approach. *Fourth IEEE International Conference on Data Mining (ICDM'04)*, páginas 217–224. Citado na página [19](#).
- Tomás, J. T., Spolaôr, N., Cherman, E. A., e Monard, M. C. (2013a). A framework to generate synthetic multi-label datasets. In *Conferencia Latino-americana de Informática (CLEI 2013)*, páginas 1–11, Vargas – Venezuela. Citado na página [8](#).
- Tomás, J. T., Spolaôr, N., Cherman, E. A., e Monard, M. C. (2013b). Projeto e implementação de um framework para geração de conjunto de dados sintéticos para aprendizado multirrótulo. Relatório técnico, Instituto de Ciências Matemáticas e de Computação – ICMC/USP. Nº 391. Citado na página [8](#).
- Trohidis, K., Tsoumakas, G., Kalliris, G., e Vlahavas, I. (2008). Multilabel classification of music into emotions. In *ISMIR '08: 9th International Conference on Music Information Retrieval*, páginas 325–330. Citado nas páginas [11](#) e [19](#).
- Tsoumakas, G. (2013). Personal communication. Citado na página [37](#).
- Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I., e Vlahavas, I. (2009a). Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proc. ECML/PKDD 2009 Workshop on Learning from Multi-Label Data (MLD'09)*, páginas 101–116. Citado nas páginas [12](#) e [18](#).
- Tsoumakas, G. e Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehouse and Mining*, 3(3):1–13. Citado nas páginas [27](#) e [37](#).
- Tsoumakas, G., Katakis, I., e Vlahavas, I. (2008). Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In *Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, páginas 1–15. Citado nas páginas [19](#), [22](#), e [23](#).

- Tsoumakas, G., Katakis, I., e Vlahavas, I. (2009b). Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*, páginas 1–19. Citado nas páginas 2, 11, 12, e 31.
- Tsoumakas, G., Spyromitros, E., Vilcek, J., e Vlahavas, I. (2011). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414. Citado nas páginas 22 e 88.
- Tsoumakas, G. e Vlahavas, I. (2007). Random k-labelsets: An ensemble method for multilabel classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4701:406–417. Citado nas páginas 12, 14, e 45.
- Ueda, N. e Saito, K. (2003). Parametric mixture models for multi-labeled text. *Advances in Neural Information Processing Systems 15*. Cambridge: MIT Press. Citado nas páginas 11 e 45.
- Vens, C., Struyf, J., Schietgat, L., Džeroski, S., e Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214. Citado na página 11.
- Vlachos, A. (2008). A stopping criterion for active learning. *Comput. Speech Lang.*, 22(3):295–312. Citado na página 86.
- Worring, M., Snoek, C., de Rooij, O., Nguyen, G., e Smeulders, A. (2007). The mediamill semantic video search engine. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, páginas IV–1213–IV–1216. Citado na página 40.
- Yang, B., Sun, J.-T., Wang, T., e Chen, Z. (2009). Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, páginas 917–926, New York, NY, USA. ACM. Citado nas páginas 79, 83, 86, e 87.
- Yang, Y. (2001). A study on thresholding strategies for text categorization. In *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, páginas 137–145. ACM Press. Citado nas páginas 23 e 25.
- Zhang, M.-L. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. on Knowl. and Data Eng.*, 18(10):1338–1351. Citado nas páginas 11, 19, 22, 38, e 45.

- Zhang, M.-L. e Zhang, K. (2010). Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, páginas 999–1008, New York, NY, USA. ACM. Citado na página [44](#).
- Zhang, M.-L. e Zhou, Z. (2007a). Multi-Label learning by instance differentiation. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, páginas 669–674, Vancouver - Canadá. AAAI Press. Citado na página [11](#).
- Zhang, M.-L. e Zhou, Z.-H. (2005). A k-nearest neighbor based algorithm for multi-label classification. *IEEE International Conference on Granular Computing*, 2:718–721. Citado nas páginas [19](#) e [20](#).
- Zhang, M.-L. e Zhou, Z.-H. (2007b). ML-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038 – 2048. Citado na página [45](#).
- Zhang, M.-L. e Zhou, Z.-H. (2013). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 99(Pre-Prints):1. Citado nas páginas [26](#) e [27](#).
- Zhu, S., Ji, X., Xu, W., e Gong, Y. (2005). Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, páginas 274–281, New York, NY, USA. ACM. Citado na página [45](#).