Proposta de um método de combinação de classificadores para construção de classificadores multirrótulo

Kassio Novaes Calembo[1,2] Flavia Cristina Bernardini[3]
Carlos Bazilio Martins[4]

Instituto de Ciência e Tecnologia – Polo Universitário de Rio das Ostras
Universidade Federal Fluminense – (UFF)
Rio das Ostras – RJ, Brasil

Resumo

Aprender conceitos e padrões a partir de dados é um objetivo em aprendizado de máquina. Os classificadores construídos pelos atuais algoritmos de aprendizado rotulam os exemplos com apenas um rótulo, ou oferecem uma probabilidade de o exemplo pertencer às classes do domínio. Entretanto, existem problemas nos quais um exemplo necessita ser multi-rotulado. Uma solução é construir um classificador para cada rótulo e combinar suas decisões. Um método clássico baseado nessa abordagem é denominado *Binary Relevance* — BR. Neste trabalho, é apresentada uma proposta de extensão desse método, denominada *Ensemble Binary Relevance* — EBR —, na qual são construídos classificadores induzidos por diferentes algoritmos de aprendizado para compor o classificador de cada rótulo. Também são descritos experimentos realizados com ambos os métodos, utilizando conjuntos de dados naturais.Os resultados obtidos foram considerados promissores.

Keywords: Inteligência Computacional, Aprendizado de Máquina Supervisionado, Problemas Multirrótulo, Ensembles para Classificação Multirrótulo.

1 Introdução

Um dos principais objetivos de aprendizado de máquina é aprender conceitos e padrões a partir de dados. Os classificadores construídos pelos atuais algoritmos de aprendizado de máquina rotulam os exemplos com apenas uma classe,

[1] Trabalho realizado com auxílio do programa PIBIC — Universidade Federal Fluminense
[2] Email: kassiocalembo@gmail.com
[3] Email: fcbernardini@puro.uff.br
[4] Email: bazilio@ic.uff.br

e/ou oferecem probabilidades de um exemplo pertencer às classes do domínio em questão. Entretanto, existem problemas nos quais um exemplo é rotulado com mais de uma classe, como por exemplo, rotulamento de textos, vídeos ou imagens, onde é necessário que se utilize métodos específicos para fornecer esse tipo de classificação. Uma maneira de resolver esse tipo de problema é decompor o problema multirrótulo original em múltiplos subproblemas binários, construindo um classificador para cada classe e combinar as saídas dos classificadores criados para obtenção da classificação final. Existem diferentes argumentos para motivar a utilização desta abordagem para a solução de problemas multirrótulo, dentre esses motivos podemos citar alguns: há algoritmos de aprendizado bastante estudados e aceitos pela comunidade de aprendizado de máquina para solução de problemas binários; em geral, os algoritmos não são adequados a problemas com grande número de classes ou apresentam dificuldade em lidar com grandes volumes de dados de treinamento; e o uso de técnicas baseadas nesta abordagem podem reduzir a complexidade computacional, graças à divisão do problema inicial em subproblemas mais simples. Ainda, deve ser observado que, para problemas de aprendizado supervisionado padrão, ou seja, problemas nos quais o atributo classe possui somente um valor associado a cada exemplo, melhores resultados podem ser obtidos utilizando métodos de construção de *ensembles* de classificadores [2].

Neste trabalho propomos um método de combinação de classificadores que explora a diversidade de classificadores induzidos por diferentes algoritmos de aprendizado de máquina. Esperamos obter uma complementaridade entre as predições para cada problema binário decomposto do problema original. O método proposto foi implementado utilizando a biblioteca Mulan [5] [13], que é baseado na ferramenta Weka [14]. Foram utilizados conjuntos de dados naturais, disponibilizados juntamente com a biblioteca Mulan para avaliação do método proposto.

Este artigo está organizado da seguinte forma: Na Seção 2 são descritos conceitos e definições de aprendizado de máquina e *ensembles* de classificadores. Na Seção 3 são descritos conceitos de aprendizagem multirrótulo, incluindo uma descrição de um conjunto de dados artificial construído com o intuito de facilitar a compreensão de problemas multirrótulo, uma descrição de métricas de avaliação de modelos multirrótulo e uma descrição do método BR. Na Seção 4 é descrito o método EBR, proposto neste trabalho. Na Seção 5, são apresentadas algumas considerações em relação à implementação do método EBR. Na Seção 6 são descritos os experimentos realizados, incluindo uma descrição dos conjuntos de dados, assim como são descritos e analisados os resultados obtidos. Por fim, na Seção 6 é realizada a conclusão e são mencionados possíveis trabalhos futuros.

[5] Disponível em http://mulan.sourceforge.net

2 Aprendizado de Máquina Supervisionado e *Ensembles* de Classificadores

No problema padrão de aprendizado supervisionado, a entrada do algoritmo consiste de um conjunto de exemplos S, com N exemplos $T_i, i = 1, ..., N$, escolhidos de um domínio X com uma distribuição D fixa, desconhecida e arbitrária, da forma $\{(\mathbf{x}_1,y_1),...,(\mathbf{x}_N,y_N)\}$ para alguma função desconhecida $y = f(\mathbf{x})$. Os $\mathbf{x}_i$ são tipicamente vetores da forma (x_{i1}, x_{i2}, ..., x_{iM}) com valores discretos ou numéricos. x_{ij} refere-se ao valor do atributo j, denominado $\mathbf{X}_j$, do exemplo T_i.Os valores y_i referem-se ao valor do atributo Y, freqüentemente denominado classe.

Os valores de y são tipicamente pertencentes a um conjunto discreto de classes $C_v, v = 1, ..., N_{Cl}$, *i.e* $y \in \{C_1, ..., C_{N_{Cl}}\}$, quando se trata de *classificação*, ou ao conjunto de números reais em caso de *regressão*. O enfoque deste trabalho é problemas de classificação. Assim, quando for dito que um exemplo pertence à uma determinada classe C_v, isso significa que o exemplo possui C_v como valor de y, ou, ainda, o valor C_v foi associado a y quando o problema é associar uma classe a um exemplo.

Dado um conjunto S de exemplos a um algoritmo de aprendizado, um *classificador* $\mathbf{h}$ será induzido. O classificador consiste da *hipótese* feita sobre a verdadeira (mas desconhecida) função f. Dados novos exemplos $\mathbf{x}$, o classificador, ou hipótese, $\mathbf{h}$ prediz o valor correspondente y.

Métodos de construção de *ensembles* de classificadores consistem em induzir múltiplos classificadores e combinar as saídas de cada um deles. Durante a fase de treino, o conjunto de dados é utilizado para construir todos os classificadores que compõem o ensemble [6]. Na fase de teste, um novo conjunto de dados é fornecido a todos os classificadores induzidos durante a fase de treino, os quais produzem individualmente suas estimativas. As saídas dos classificadores são os dados de entrada para um método de combinação, onde serão combinadas para gerar a decisão consensual final do modelo. Uma das principais motivações em construir *ensembles* de classificadores está embasada no fato de *ensembles* combinarem classificadores que teoricamente possuem erros distintos e, assim, a combinação pode estar mais próxima da função verdadeira e desconhecida f do que cada classificador individual em particular [2].

Para avaliar um classificador $\mathbf{h}$, é necessário coletar informações das decisões tomadas pelo classificador em um conjunto de teste não utilizado na fase de treino desse classificador. A técnica de validação cruzada, ou *k-fold cross-validation*, é uma técnica comumente usada para esse fim. No *k-fold cross-validation*, o conjunto S de dados é dividido aleatoriamente em k partições

[6] O conjunto de dados de treinamento também pode ser utilizado para definir se devem ser induzidos mais classificadores, sendo essa decisão dependente do método de combinação, como é o caso do método *Boosting* [4]

$S_1, ..., S_k$ disjuntas, sendo todas as partições de conjuntos de dados de aproximadamente o mesmo tamanho. Após, são executadas k iterações de indução e teste de um classificador. Na primeira iteração, é induzido o classificador $\mathbf{h}_1$ com os conjuntos de dados $S_2, ..., S_k$, sendo depois testado com o conjunto S_1. Com os resultados obtidos, alguma métrica de avaliação *eval* pode ser usada. Assim, obtém-se a avaliação $eval(\mathbf{h}_1)$. Na segunda iteração, é induzido o classificador $\mathbf{h}_2$ com os conjuntos $S_1, S_3, ..., S_k$, sendo depois testado com o conjunto S_2, obtendo assim $eval(\mathbf{h}_2)$, e assim sucessivamente. Tendo em vista que agora existem k estimadores de *eval* independentes [7] $eval(\mathbf{h}_1), ..., eval(\mathbf{h}_k)$, com eles pode-se estimar a média $mean_{eval}$ e o erro padrão $stdErr_{eval}$ de $eval$, definidos respectivamente pelas Equações 1 e 2, do modelo final. Esse modelo final é construído utilizando todos os exemplos disponíveis.

$$mean_{eval}(\mathbf{h}) = \frac{1}{k}\sum_{k=1}^{k} Err(\mathbf{h}_k) \tag{1}$$

$$stdErr_{eval}(\mathbf{h}) = \frac{1}{\sqrt{k-1}}\sqrt{\frac{1}{k}\sum_{k=1}^{k}(Err(\mathbf{h}_k) - m\_err(\mathbf{h}_k))^2} \tag{2}$$

3 Aprendizado Multirrótulo

A grande maioria dos problemas de classificação apresentados na literatura [8] são problemas de classificação de rótulo único. Neste tipo de problema, cada exemplo é associado a uma única classe pertencente a um conjunto finito de classes. Entretanto, existe um grande número de problemas em que um determinado exemplo pode ser associado a mais de uma classe. Tais problemas recebem o nome de problemas multirrótulo. Um exemplo para essa situação é a classificação de um texto que pertence simultaneamente a mais de uma classe, como medicina e informática, ou pertence à economia, política e saúde. Apesar de problemas de classificação multirrótulo serem comuns, o estudo do processo de indução desses classificadores é relativamente novo na comunidade de aprendizado de máquina, tendo iniciado há pouco mais de uma década [8]. Observa-se que aprendizado multirrótulo é uma área emergente e promissora de pesquisa em aprendizado de máquina. Possíveis aplicações na área de aprendizado multirrótulo são classificação de imagens [10,3,6], textos [8,9], bioinformática, entre outros. Ainda, podem ser utilizados conhecimentos do domínio da aplicação para verificar classes que podem ser contraditórias e, nesses casos, pode ser dada preferência a uma classe em detrimento de outra [1].

[7] Na realidade, são "aproximadamente" independentes
[8] Muitos dos algoritmos propostos de aprendizado supervisionado estão descritos em [14]

Para o aprendizado multirrótulo, o conjunto de dados multirrótulo S é composto por N exemplos da forma $(\mathbf{x}_i, Y_i)$, com $i = 1, ..., N$ e $Y_i \subseteq L$, *i.e.*, Y_i é o conjunto de rótulos do $i$ésimo exemplo. Considerando também que $\mathbf{h}$ é um classificador multirrótulo, $Z_i = \mathbf{h}(\mathbf{x}_i)$ é o conjunto de classes preditas por $\mathbf{h}$ para um dado exemplo $\mathbf{x}_i$. Um conjunto de dados multirrótulo é ilustrado na Tabela 1.

Tabela 1
Conjunto de exemplos multirrótulo no formato atributo-valor

	X_1	X_2	...	X_M	Y
T_1	x_{11}	x_{12}	...	x_{1M}	Y_1
T_2	x_{21}	x_{22}	...	x_{2M}	Y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
T_N	x_{N1}	x_{N2}	...	x_{NM}	Y_N

Para entender em um problema simplificado o comportamento dos conjuntos de dados multirrótulo, construímos um conjunto de dados multirrótulo artificial, com dois atributos e um atributo classe multirrótulo. Os atributos descritores X_1 e X_2 são ambos atributos contínuos, com domínio no intervalo $(0, 14) \in \mathbb{N}$. Já o atributo classe possui como domínio os valores possíveis no conjunto discreto {o,+,-,*}. Foram gerados 132 exemplos. O objetivo desse conjunto de dados artificial é ilustrar regiões de intersecção, que pode tornar difícil o aprendizado do conceito multirrótulo. Na Figura 1 é ilustrado como estão distribuídos os exemplos desse conjunto de dados, onde nos casos em que o exemplo possui mais de um rótulo, esses são colocados um ao lado do outro. Nessa figura, pode ser notado que, na parte superior esquerda do gráfico estão exemplos rotulados com {o}, {+} ou {o,+}, e na parte inferior direta estão exemplos rotulados com {*}, {-} ou {*,-}. Na região central, com fundo em cinza, estão exemplos que foram rotulados com combinações dos 4 (quatro) possíveis rótulos, sendo essa a região mais difícil de realizar aprendizado.

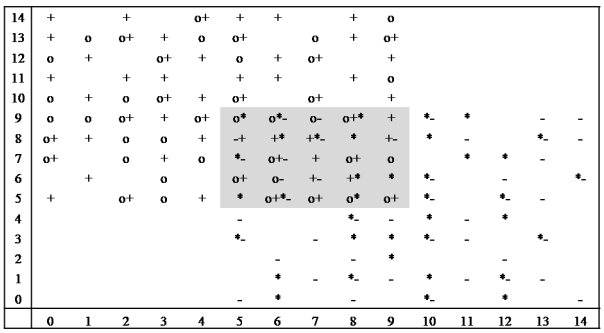

Figura 1. Conjunto de dados artificiais multirrótulo

3.1 Estatísticas dos Conjuntos de Dados Multirrótulo

Em alguns conjuntos de dados, o número de rótulos de cada exemplo é pequeno, se comparado ao número total de rótulos possíveis $|L|$. Esse número pode ser um parâmetro que influencia a performance de diferentes métodos multirrótulo. Para avaliar o quanto essa medida pode impactar, são definidas duas medidas de um conjunto de dados: cardinalidade $Card$ e densidade $Dens$ de rótulo de um conjunto de dados [12]. Cardinalidade de um conjunto de dados S é a média do número de rótulos dos exemplos pertencentes a S, definida pela Equação 3. Já a densidade de um conjunto S é o número médio de rótulos dos exemplos pertencentes a S dividido pelo número de rótulos $|L|$, definida pela Equação 4.

$$Card(S) = \frac{1}{N}\sum_{i=1}^{N}|Y_i| \tag{3}$$

$$Dens(S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{|L|} \tag{4}$$

A cardinalidade de rótulo é independente do número de rótulos $|L|$ no problema de classificação, e é utilizado para quantificar o número de rótulos alternativos que caracterizam os exemplos de um conjunto de dados multirrótulo utilizado para treinamento. Já a densidade de rótulo considera o número de rótulos do domínio $\mathcal{D}$. Dois conjuntos de dados com a mesma cardinalidade de rótulo mas com grande diferença no número de rótulos — e portanto grande diferença na medida de densidade de rótulo — podem não exibir as mesmas propriedades e, assim, causar comportamentos diferentes nos métodos de aprendizado multirrótulo. O número rótulos distintos é também importante para muitos métodos baseados em algoritmos de transformação. Sendo assim, é importante avaliar tais medidas quando se utiliza métodos de aprendizado multirrótulo.

3.2 Medidas de Avaliação

Para avaliar os classificadores multirrótulo, diferentes classes de medidas podem ser utilizadas [12]. Nesse trabalho utilizamos medidas baseadas em exemplos, pois essas medidas avaliam o comportamento dos modelos multirrótulo do ponto de vista de classificação de cada exemplo. As medidas utilizadas são *Hamming Loss* (Ham), precisão ($Prec$), *recall* (Rec), acurácia (Acc), medida F ($FMeasure$) e *subset accuracy* ($SubsetAcc$), definidas pelas Equações 5 a 10, respectivamente. Em relação à medida *Hamming Loss*, deve ser observado que essa medida calcula uma distância de Hamming entre a classificação correta e a classificação predita pela máquina. Na Equação 5, Δ representa a diferença simétrica entre dois conjuntos. Quando Ham é considerada como medida de

avaliação, quanto menor o valor, melhor é a performance do algoritmo, tendo zero como valor ideal. Para as demais medidas, valores maiores indicam melhor performance. Na Equação 10, $I(\text{verdadeiro}) = 1$ e $I(\text{verdadeiro}) = 0$. Deve ser observado que a medida $SubsetAcc$ é uma medida bastante conservadora, pois requer que o conjunto de rótulos preditos seja exatamente igual ao conjunto de rótulos verdadeiros. Pode ser notado, na definição da medida $FMeasure$ — Equação 9 — é uma combinação das métricas $Prec$ e Rec, e portanto avalia um modelo multirrótulo como uma ponderação entre essas métricas.

$$Hamm(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \Delta Z_i|}{|L|} \tag{5}$$

$$Prec(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Yi \cap Zi|}{|Zi|} \tag{6}$$

$$Rec(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Yi \cap Z_i|}{|Y_i|} \tag{7}$$

$$Acc(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{8}$$

$$FMeasure(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \tag{9}$$

$$SubsetAcc(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N} I(Z_i = Y_i) \tag{10}$$

Deve ser observado que cada uma das medidas de avaliação descritas pode ser utilizada como uma métrica *eval* calculada em cada iteração da técnica de *k-fold cross-validation*, para estimar o poder de predição de um classificador multirrótulo construído por métodos de construção de classificadores multirrótulo.

3.3 O Método BR

Uma possível solução para este tipo de problema é a decomposição do problema multirrótulo em vários subproblemas binários. O problema de classificação binário é o mais estudado até hoje e, mesmo sendo o tipo mais simples, é considerado o mais importante, pois com algumas modificações, problemas mais complicados podem ser reduzidos a ele. Neste caso, pode ser construído um comitê de classificadores para oferecer mais de uma classe como saída para um mesmo exemplo [1,12]. Um método clássico que realiza a decomposição em vários problemas binários é denominado *Binary Relevance* — BR. No método BR, é construído um classificador para cada classe com um mesmo algoritmo

de aprendizado de máquina supervisionado. Para simplificação, chamamos um classificador binário componente da solução multirrótulo de classificador base. Para isso, inicialmente o conjunto de dados de treinamento, cujos exemplos de treinamento possuem mais de um rótulo, é transformado em $|L|$ conjuntos de dados S_l, sendo cada conjunto de dados referente a um rótulo l. Dado um algoritmo de aprendizado supervisionado para problemas de classificação de somente um rótulo, é construído um classificador base para cada conjunto de dados S_l. Para classificar um exemplo novo, o exemplo é fornecido a cada um dos classificadores referentes a cada rótulo. Caso o classificador base prediga que o exemplo é positivo, o conjunto de rótulos de saída recebe o rótulo ao qual o classificador se refere. O funcionamento do modelo é ilustrado na Figura 2.

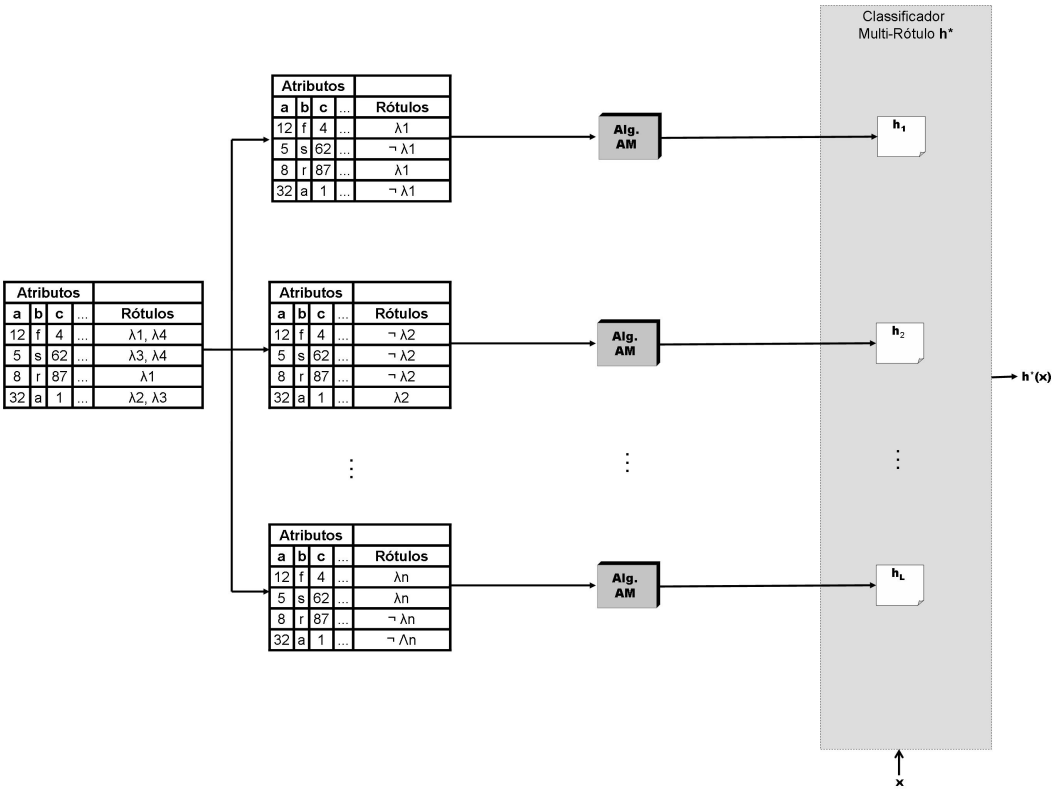

Figura 2. Método *Binary Relevance*

Assim como nos problemas de aprendizado supervisionado tradicionais, também quando se utiliza o método BR é necessário avaliar o desempenho do método com vários algoritmos de aprendizado de máquina para indução dos classificadores binários, a fim de verificar qual o algoritmo base que obtém melhores resultados para um problema em questão. Dessa maneira, a tarefa de avaliar o método BR utilizando diferentes algoritmos pode ser mais complexa que induzir classificadores binários com todos os algoritmos possíveis que se deseja considerar, e combiná-los. Ainda, nosso objetivo é explorar a diversidade dos diferentes algoritmos, para conseguir uma complementaridade entre

as predições, já que classificadores distintos tendem a cometer erros distintos. A seguir, é proposto o método EBR, que combina classificadores de diferentes algoritmos de aprendizado supervisionado.

4 O Método EBR

O método EBR é uma extensão do método BR. A proposta consiste em induzir diferentes classificadores, utilizando diferentes algoritmos de aprendizado de máquina, para cada uma das classes. Para isso, assim como no BR, o conjunto de dados de treino é transformado em $|L|$ conjuntos de dados S_l. No método BR, apenas um algoritmo é utilizado para induzir o classificador para aquele rótulo. No método EBR, são utilizados P algoritmos distintos, que induzem P classificadores distintos para cada conjunto de dados S_l, condizente a cada rótulo l. No BR, são induzidos $|L|$ classificadores; já no EBR, são induzidos $P \times |L|$ classificadores. Dessa maneira, é esperado que os erros cometidos por um classificador sejam compensados pelos outros $P-1$ classificadores, quando se utiliza P classificadores oriundos de P algoritmos distintos.

Após serem induzidos os P classificadores para cada conjunto de dados S_l, quando um novo exemplo é dado para ser classificado, o mesmo é classificado por cada um dos P classificadores de cada rótulo l. Daí, a saída dos classificadores é então combinada para a predição daquele rótulo. O exemplo é fornecido aos P classificadores referentes ao primeiro rótulo. Quando um classificador base classifica um exemplo, o classificador também oferece como saída a probabilidade do exemplo pertencer à essa classe. As probabilidades do exemplo pertencer à classe, fornecidas por cada classificador, são somadas, e uma média simples é realizada. Se a probabilidade do exemplo pertencer àquele rótulo for maior do que a probabilidade de não pertencer, o rótulo é colocado no conjunto de rótulos a ser oferecido como saída para o exemplo. Tal procedimento é repetido para os $l-1$ rótulos restantes. O funcionamento do método EBR é ilustrado na Figura 3.

Há inúmeros trabalhos que exploram a desvantagem do método BR de não considerar a relação entre os rótulos na construção do modelo multirrótulo [5,7,11]. Entretanto, dependendo das características do conjunto de exemplos, o método BR pode obter bons resultados. Assim, o método EBR se beneficia da simplicidade e dos bons resultados obtidos com o método BR, assim como associa o fato de explorar o *bias* de diferentes algoritmos de aprendizado supervisionado.

5 Implementação do Método EBR

Uma ferramenta computacional livre desenvolvida para auxiliar o processo de mineração de dados, incluindo ferramentas implementadas para tarefas

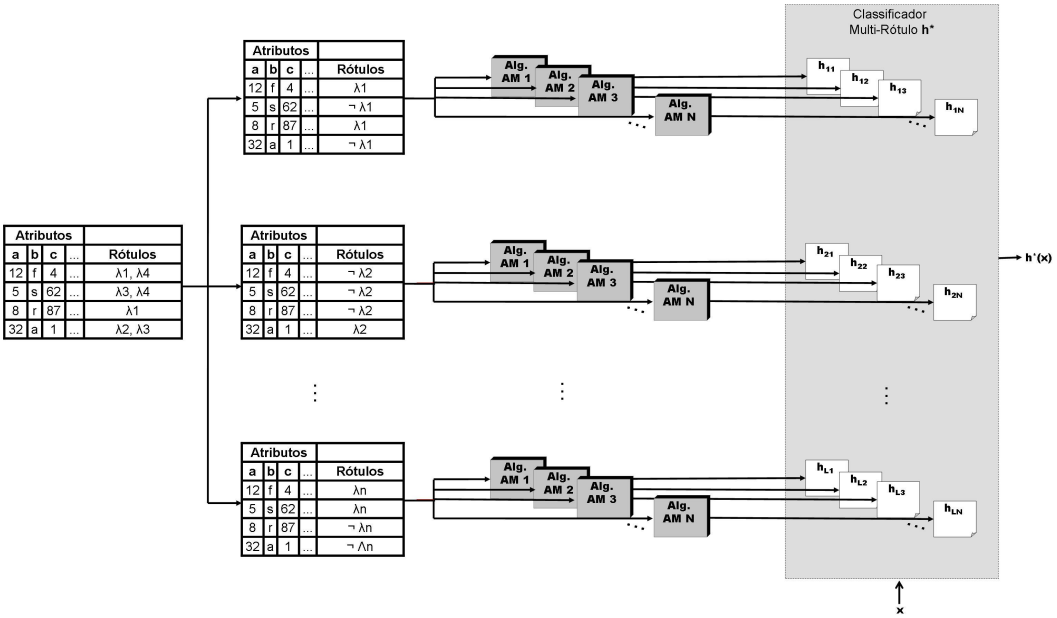

Figura 3. Método EBR

de classificação, aprendizado supervisionado, e outras, é a Weka [14]. Essa ferramenta possui a vantagem de ter sido implementada na linguagem Java, permitindo portabilidade. Na ferramenta Weka, um conjunto de dados só pode ser manipulado se estiver em uma única tabela, na qual cada atributo pode possuir somente um valor único, discreto ou contínuo, conforme a descrição dos atributos, também informada ao Weka juntamente com o conjunto de dados. Entretanto, a ferramenta Weka não atende as necessidades para manipulação de bases de dados multirrótulo, uma vez que os exemplos possuem no atributo classe um conjunto de valores, e não um valor único.

Para atender as necessidade do aprendizado multirrótulo, foi criada a biblioteca Mulan (*Multi-label Learning*) [13]. Essa biblioteca é uma extensão da ferramenta Weka e foi proposta para atender as necessidades dos problemas de aprendizado multirrótulo. O Mulan possui implementações de diversos algoritmos para aprendizado de máquina multirrótulo, além de fazer chamadas a algoritmos de aprendizado implementados no Weka. Para fazer chamadas aos algoritmos implementados no Weka, o Mulan possui implementações de transformações de conjuntos de dados multirrótulo em conjuntos de rótulo único.

Dentre os algoritmos para aprendizado de máquina multirrótulo está implementado o método BR. Para implementar o método BR, descrito na Seção 3.3, o Mulan possui uma implementação de transformação do conjunto de dados multirrótulo em diversos conjuntos de rótulo único específica. A transformação utilizada pelo método BR está implementada na classe BinaryRelevanceTransformation. Tal classe pertence ao pacote mulan.transformations. A biblioteca

Mulan foi escrita em Java, assim como o Weka, porém não existe interface gráfica para utilização do Mulan. O método EBR foi implementado e inserido na biblioteca Mulan.

As medidas de avaliação descritas na Seção 3.2 também foram implementadas no Mulan. Somente foi necessária uma extensão para calcular o desvio padrão de cada uma das medidas, para realização de testes estatísticos. Todas as medidas — $Hamm$, $Prec$, Rec, Acc, $FMeasure$ e $SubsetAcc$ — foram implementadas na classe LabelBasedMeasures, pertencente ao pacote mulan.evaluation. A classe LabelBasedMeasures originalmente calcula a média das medidas durante a execução da técnica *k-fold cross-validation*, e foi modificada para também calcular o erro padrão de cada medida.

6 Experimentos Realizados e Resultados Obtidos

Nesta seção, apresentamos uma descrição dos experimentos realizados e dos resultados obtidos. Os experimentos foram realizados utilizando o método EBR e o método BR. Foram utilizados diferentes algoritmos de aprendizado de máquina para indução dos classificadores binários. Foram utilizadas 7 (sete) bases de dados: Emotions, Genbase, Scene, Yeast, Enron, Medical [9] e DSArtM, sendo este último a base de dados artificial. Na Tabela 2 são descritas as características dessas bases de dados, onde #Exs. é o número de exemplos do conjunto de dados; #At. Disc e #At. Cont. são, respectivamente, o número de atributos discretos e contínuos presentes na base de dados; # Rótulos é o número de rótulos possíveis $|L|$ do conjunto de dados; $Card$ é a medida de cardinalidade de rótulo do conjunto de dados — Eq. 3 — ; e $Dens$ é a medida de densidade de rótulo do conjunto de dados — Eq. 4.

Tabela 2
Características dos conjuntos de dados

Nome	#Exs.	#At. Disc.	#At. Cont	# Rótulos	$Card$	$Dens$
Emotions	593	0	72	6	1.869	0.311
Genbase	662	1186	0	27	1.252	0.046
Scene	2407	0	294	6	1.074	0.179
Yeast	2417	0	103	14	4.237	0.303
Enron	1000	1001	0	53	3.378	0.064
Medical	978	1449	0	45	1.245	0.028
DSArtM	132	0	2	4	1.394	0.348

Inicialmente, foram induzidos classificadores multirrótulo utilizando o método BR para cada um dos conjuntos de dados descritos anteriormente. O método BR foi utilizado em 4 (quatro) cenários de experimento distintos,

[9] http://mulan.sourceforge.net/datasets.html

denominados BR+J48, BR+NB, BR+JRIP e BR+PART. Em cada cenário, foi utilizado um algoritmo distinto de aprendizado de máquina supervisionado de único rótulo, implementado na ferramenta Weka [14]. Os cenários foram (i) BR+J48: algoritmo de aprendizado C4.5 para indução de árvores de decisão, cuja implementação é denominada J48 no Weka; (ii) BR+NB: algoritmo de aprendizado Naive Bayes, que utiliza estatística bayesiana para indução dos classificadores; (iii) BR+JRIP: algoritmo de aprendizado RIPPER para indução de classificadores compostos por regras, cuja implementação é denominada JRIP no Weka, e (iv) BR+PART: algoritmo de aprendizado PART, também utilizado para indução de classificadores compostos por regras de conhecimento. Todos os algoritmos foram utilizados com seus parâmetros *default*. Em [14] podem ser encontradas maiores informações sobre os algoritmos utilizados. Também, foi utilizado o método EBR, sendo que para cada conjunto de dados foram utilizados os mesmos 4 (quatro) algoritmos de aprendizado de máquina.

Os experimentos foram conduzidos utilizando a técnica *K-fold cross-validation*. Para realização dessa técnica, o conjunto de dados é dividido igualmente em K subconjuntos: K-1 subconjuntos são unidos e fornecidos como conjunto de dados de treino para indução do classificador — neste caso, um classificador multirrótulo —, e o subconjunto restante é utilizado para testar o comportamento do classificador induzido na fase de treino. As fases de treino e teste são realizados num total de K iterações. Deve ser observado que, a cada iteração, são utilizados diferentes $K-1$ subconjuntos de treino e, consequentemente, diferentes conjuntos de teste. Nos experimentos realizados, foi utilizado $K = 10$. Em cada iteração, foram calculadas as medidas $Hamm$, $Prec$, Rec, Acc, $FMeasure$ e $SubsetAcc$ definidas pelas Equações 5 a 10, respectivamente. Ao final do processo, foram calculados a média e o desvio padrão de cada uma das medidas, considerando-se as 10 iterações. Nas Tabelas 3 a 8 são exibidos os resultados obtidos — média e erro padrão, entre parênteses — para os experimentos realizados com o método EBR e com o método BR nos 4 (quatro) cenários de experimentos distintos, utilizando as medidas Ham, Acc, $Prec$, Rec, $FMeasure$ e $SubsetAcc$, respectivamente.

Tabela 3
Medida Ham — Média e Desvio Padrão

	EBR	BR+J48	BR+NB	BR+JRIP	BR+PART
Emotions	21,5% (0,8%)	24,7% (0,9%) ▼	25,2% (0,9%) ▼	23,0% (0,9%) ▽	25,6% (1,0%) ▼
Genbase	00,1% (0,0%)	00,1% (0,0%)	03,4% (0,1%) ▼	00,1% (0,0%)	00,1% (0,0%)
Scene	11,1% (0,4%)	13,7% (0,5%) ▼	24,2% (0,9%) ▼	11,9% (0,4%) ▼	11,9% (0,4%) ▽
Yeast	21,1% (0,7%)	24,5% (0,9%) ▼	30,3% (1,1%) ▼	21,5% (0,8%)	22,0% (0,8%)
Enron	05,2% (0,2%)	05,1% (0,2%)	21,8% (0,8%) ▼	05,1% (0,2%)	06,1% (0,2%) ▼
Medical	01,0% (0,0%)	01,0% (0,0%)	02,5% (0,1%) ▼	01,0% (0,0%)	01,1% (0,0%) ▼
DSArtM	27,4% (1,1%)	28,4% (1,0%)	26,5% (1,2%)	27,0% (1,1%)	28,4% (1,0%)

Tabela 4
Medida *Prec* — Média e Desvio Padrão

	EBR	BR+J48	BR+NB	BR+JRIP	BR+PART
Emotions	62,4% (2,3%)	58,1% (2,1%) ▽	57,7% (2,1%) ▼	60,2% (2,2%)	57,6% (2,2%) ▼
Genbase	99,2% (3,5%)	99,3% (3,5%)	33,8% (1,4%) ▼	99,2% (3,5%)	99,3% (3,5%)
Scene	**65,0% (2,3%)**	55,3% (2,0%) ▼	45,9% (1,6%) ▼	57,0% (2,1%) ▼	59,4% (2,1%) ▼
Yeast	60,7% (2,4%)	57,8% (2,2%)	61,1% (1,9%)	56,0% (2,4%) ▽	66,8% (2,4%) ▲
Enron	61,4% (2,2%)	61,8% (2,2%)	22,9% (0,8%) ▼	61,6% (2,2%)	53,8% (1,9%) ▼
Medical	78,1% (2,8%)	77,9% (2,8%)	42,4% (1,6%) ▼	80,4% (2,8%)	76,6% (2,7%)
DSArtM	60,3% (2,3%)	57,9% (2,1%)	52,2% (2,2%) ▼	60,7% (2,3%)	57,9% (2,1%)

Tabela 5
Medida *Rec* — Média e Desvio Padrão

	EBR	BR+J48	BR+NB	BR+JRIP	BR+PART
Emotions	66,4% (2,4%)	59,9% (2,2%) ▼	77,3% (2,8%) ▲	58,7% (2,2%) ▼	61,4% (2,2%) ▼
Genbase	99,0% (3,5%)	99,1% (3,5%) ▼	30,1% (1,3%)	99,0% (3,5%)	99,1% (3,5%)
Scene	75,5% (2,7%)	63,3% (2,3%) ▼	85,8% (3,0%) ▲	62,9% (2,3%) ▼	66,8% (2,4%) ▼
Yeast	60,7% (2,1%)	57,8% (2,0%)	61,1% (2,2%)	56,0% (2,0%) ▼	56,1% (2,0%) ▼
Enron	55,2% (2,0%)	50,3% (1,8%) ▼	71,5% (2,5%) ▲	50,9% (1,8%) ▼	50,4% (1,8%) ▼
Medical	79,5% (2,8%)	80,3% (2,8%)	43,8% (1,6%) ▼	84,9% (3,0%)	79,4% (2,8%)
DSArtM	68,2% (2,7%)	83,3% (3,3%) ▼	72,3% (2,9%)	72,0% (2,8%)	83,3% (3,3%) ▼

Tabela 6
Medida *Acc* — Média e Desvio Padrão

	EBR	BR+J48	BR+NB	BR+JRIP	BR+PART
Emotions	52,5% (1,9%)	46,2% (1,7%) ▼	52,9% (1,9%)	47,3% (1,8%) ▼	45,6% (1,7%) ▼
Genbase	98,5% (3,5%)	98,6% (3,5%)	30,1% (1,3%) ▼	98,5% (3,5%)	98,6% (3,5%)
Scene	**63,4% (2,3%)**	53,5% (1,9%) ▼	45,2% (1,6%) ▼	55,2% (2,0%) ▼	57,8% (2,1%) ▼
Yeast	49,7% (1,8%)	44,0% (1,6%) ▼	42,0% (1,5%) ▼	47,0% (1,7%) ▽	46,7% (1,7%) ▽
Enron	43,7% (1,6%)	41,3% (1,5%) ▽	19,5% (0,7%) ▼	41,2% (1,5%) ▽	37,8% (1,4%) ▼
Medical	74,7% (2,7%)	74,6% (2,6%)	37,1% (1,4%) ▼	77,4% (2,7%)	73,1% (2,6%)
DSArtM	51,4% (2,0%)	53,8% (2,0%)	48,7% (2,1%)	53,0% (2,1%)	53,8% (2,0%)

Tabela 7
Medida $FMeasure$ — Média e Desvio Padrão

	EBR	BR+J48	BR+NB	BR+JRIP	BR+PART
Emotions	64,3% (2,3%)	58,9% (2,1%) ▼	66,0% (2,4%)	59,4% (2,2%) ▼	59,2% (2,2%) ▼
Genbase	99,1% (3,5%)	99,2% (3,5%)	31,8% (1,3%) ▼	99,1% (3,5%)	99,2% (3,5%)
Scene	**69,9% (2,5%)**	59,0% (1,5%) ▼	59,8% (2,1%) ▼	59,8% (2,2%) ▼	62,9% (2,2%) ▼
Yeast	63,9% (2,3%)	59,4% (2,1%) ▼	56,7% (2,0%) ▼	61,6% (2,2%)	60,9% (2,2%)
Enron	58,1% (2,1%)	55,5% (2,0%)	34,7% (1,2%) ▼	55,7% (2,0%)	52,0% (1,9%) ▼
Medical	78,8% (2,8%)	79,1% (2,8%)	43,1% (1,6%) ▼	82,5% (2,9%)	78,0% (2,8%)
DSArtM	63,6% (2,4%)	67,8% (2,5%)	60,3% (2,5%)	65,6% (2,5%)	67,8% (2,5%)

Tabela 8
Medida *SubsetAcc* — Média e Desvio Padrão

	EBR	BR+J48	BR+NB	BR+JRIP	BR+PART
Emotions	**25,8% (1,0%)**	18,4% (0,9%) ▼	20,6% (0,9%) ▼	21,4% (1,2%) ▼	15,7% (0,8%) ▼
Genbase	97,0% (3,4%)	97,1% (3,4%)	27,8% (1,2%) ▼	97,0% (3,4%)	97,1% (3,4%)
Scene	**50,3% (1,8%)**	42,7% (2,1%) ▼	16,9% (0,6%) ▼	45,9% (1,7%) ▼	47,7% (1,7%) ▽
Yeast	**13,0% (0,5%)**	06,8% (0,3%) ▼	09,5% (0,4%) ▼	09,6% (0,4%) ▼	08,9% (0,4%) ▼
Enron	**11,2% (0,5%)**	10,3% (0,4%) ▼	00,1% (0,0%) ▼	09,9% (0,4%) ▼	08,6% (0,4%) ▼
Medical	66,0% (2,4%)	65,5% (2,3%) ▼	26,6% (1,0%)	67,1% (2,4%)	63,2% (2,3%)
DSArtM	26,6% (1,8%)	19,7% (1,4%) ▼	22,1% (1,4%) ▼	27,3% (1,8%)	19,7% (1,4%) ▼

Para avaliar os resultados obtidos, foi realizado o teste t para verificação se o EBR apresenta melhor comportamento que o método BR. Nas Tabelas 3 a 8, ▽ indica que no cenário de experimentação referente utilizando o conjunto de dados referente, o método EBR obteve melhor resultado na medida em questão com 90% de confiança; ▼ indica que no cenário de experimentação referente utilizando o conjunto de dados referente, o método EBR obteve melhor resultado na medida em questão com 95% de confiança; e ▲ indica que no cenário de experimentação referente utilizando o conjunto de dados referente, o método BR utilizando o algoritmo de aprendizado do cenário obteve melhor resultado na medida em questão com 95% de confiança. Resultados nos quais o método EBR obteve resultado superior ao método BR em todos os cenários de experimentação, no conjunto de dados em questão e na medida em questão, com pelo menos 90% de confiança, são mostrados em negrito.

Em relação ao conjunto de dados artificial, é interessante notar que o conjunto de dados é mais simples, porém não houve melhora nos resultados em relação ao método BR. Na realidade, esse resultado não é inesperado, justamente pelo conjunto de dados ser simples e, assim, é esperado que o comportamento dos algoritmos de aprendizado seja semelhante. Somente na medida *SubsetAcc* há melhora significativa em relação a três dos quatro cenários para esse conjunto de dados, mas o resultado do método EBR é bastante próximo ao método BR+JRIP, indicando que o método EBR tende a ser conservador em relação ao melhor cenário do método BR para esse conjunto de dados.

Em relação à base Scene, o método EBR apresentou melhores resultados que o método BR para as medidas *Prec*, *Acc*, $FMeasure$ e *SubsetAcc*. Para essa base o método EBR apresentou os melhores resultados. Segundo a Tabela 2, esse conjunto de dados possui a menor cardinalidade dentre todos os conjuntos de dados. Entretanto, para cardinalidades próximas — Genbase e Medical — essa relação não ocorreu.

Em relação à base Emotions, pode ser observado que o método EBR venceu o método BR em ao menos três dos quatro cenários em todas as medidas. Exceto para a base de dados artificial DSArtM, em todas as medidas para todas as bases, o método EBR vence o método BR em pelo menos um cenário.

O método EBR apresentou pior resultado que o método BR somente para a medida *Rec*, em relação ao cenário BR+NB, para três conjuntos de dados — Emotions, Scene e Enron.

É interessante notar que, para a medida mais conservadora de avaliação *SubsetAcc*, é apresentada uma melhora significativa do método EBR em relação ao BR em 4 (quatro) conjuntos de dados — Emotions, Scene, Yeast e Enron. Esses resultados mostram que o método EBR é indicado para solução de problemas multirrótulo. Outro fato que reforça essa indicação se dá ao fato de não ser possível rejeitar a hipótese nula na grande maioria dos experimentos: O método EBR apresenta resultados iguais à melhor combinação do método BR com algum algoritmo de aprendizado usado como base. A hipótese nula só é rejeitada quando o método EBR é comparado ao método BR no cenário BR+NB com as bases Emotions, Scene e Enron utilizando a medida *Rec*, como mencionado anteriormente. Isso indica que o método EBR é conservador em relação ao método BR, ou seja, oferece no mínimo o melhor resultado que o método BR pode oferecer com os algoritmos de aprendizado utilizados, para a maioria das métricas utilizadas.

7 Conclusões e Trabalhos Futuros

Neste trabalho, propomos um método para construção de classificadores multirrótulo baseado em combinação de classificadores binários, denominado EBR. O método EBR é uma extensão do método BR. O método foi implementado utilizando as bibliotecas Mulan e Weka, na linguagem Java. Foram utilizados 6 (seis) conjuntos de dados naturais e um conjunto de dados artificial para avaliar o método proposto, e 4 (quatro) algoritmos de aprendizado supervisionado distintos para compor os cenários de experimentação.

Os resultados obtidos nos experimentos realizados foram considerados promissores, indicando que a utilização do método EBR pode apresentar melhores resultados segundo a medida mais conservadora de avaliação do comportamento de um método multirrótulo. Ainda, apresenta em geral resultados muito próximos à melhor combinação do método BR com um algoritmo de aprendizado supervisionado. Esse fato pode indicar que o método EBR pode ser utilizado para avaliar como se dá o comportamento do método BR, sem ter que realizar cada cenário de experimentação em particular. Caso se verifique que o método EBR pode satisfazer as necessidades do problema, os outros cenários podem ser executados para refinar a solução. Futuramente, pretendemos explorar o método proposto em mais conjuntos de dados, e com uma maior variedade de algoritmos de aprendizado de máquina, incluindo indução de SVMs, para verificar se esses resultados se mantém.

Uma das desvantagens do método BR está relacionada ao fato de desconsiderar a dependência entre os rótulos, assim como desconsiderar o des-

balanceamento dos dados nos subproblemas de rótulo único gerados. Assim, futuramente também pretendemos explorar tais desvantagens do método BR no método EBR.

Referências

[1] F. Bernardini, A. Garcia, and I. Ferraz. Artificial intelligence based methods to support motor pump multi-failure diagnostic. *Engineering Intelligent Systems*, 17:1–25, 2009.

[2] T. G. Dietterich. *The Handbook of Brain Theory and Neural Networks*, chapter Ensemble learning. MIT Press, 2nd edition, 2002.

[3] A. Dimou, G. Tsoumakas, V. Mezaris, I. Kompatsiaris, and I. Vlahavas. An empirical study of multi-label learning methods for video annotation. In *7th International Workshop on Content-Based Multimedia Indexing, IEEE*, pages 19–24, 2009.

[4] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learninng and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[5] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *PAKDD — LNCS*, volume 3056, pages 22–30. Springer, 2004.

[6] G. Nasierding, G. Tsoumakas, and A. Kouzani. Clustering based multi-label classification for image annotation and retrieval. In *2009 IEEE International Conference on Systems, Man, and Cybernetics, IEEE*, 2009.

[7] J. Read, B. Pfahringer, and G. Holmes. Multi-label classification using ensembles of pruned sets. In *Int. Conf. Data Mining. IEEE Computer Society*, pages 995–1000, 2008.

[8] R. E. Schapire and Y. Singer. *Boostexter: a boosting-based system for text categorization. Machine Learning*, chapter 2-3, pages 135–168. 2000.

[9] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, pages 1–47, 2002.

[10] X. Shen, M. Boutell, J. Luo, and C. Brown. Multi-label machine learning and its application to semantic scene classification. In *Proceedings of the 2004 International Symposium on Electronic Imaging*, pages 18–22. 2004.

[11] G. Tsoumakas, A. Dimou, E. Spyromitros, V. Mezaris, I. Kompatsiaris, and I. Vlahavas. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proc. 1st International Workshop on Learning from Multi-Label Data (MLD'09)*, pages 101–116, 2009.

[12] G. Tsoumakas, I. Katakis, and I. Vlahavas. *Data Mining and Knowledge Discovery Handbook*, chapter Mining Multi-label Data. Springer, 2nd edition, 2010.

[13] G. Tsoumakas, J. Vilcek, E. Spyromitros, and I. Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 2010. (Accepted for publication conditioned on minor revisions).

[14] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.