Proposta de um novo método para classificação multirrótulo baseado em seleção aleatória e *bagging*

Patrícia Pachiega da Gama[1], Flavia C. Bernardini[2], Bianca Zadrozny[1,3]

[1] Instituto de Computação (IC)
Universidade Federal Fluminense (UFF) – Niterói, RJ

[2]Laboratório de Inovação no Desenvolvimento de Sistemas (LabIDeS)
Departamento de Computação – Instituto de Ciência e Tecnologia
Universidade Federal Fluminense (UFF) – Rio das Ostras, RJ

[3]IBM Research Brasil
Rio de Janeiro, RJ – Brasil

pgama@ic.uff.br, fcbernardini@puro.uff.br, biancaz@br.ibm.com

Abstract. *In many real world prediction problems, a classifier should be able to assign not only one but many labels to an example, e.g. prediction of machine failures, musical genre classification, etc. For this kind of problem, multi-label classification methods are needed. One approach frequently used to learn multi-label predictors divides the problem into one or more multi-class classification problems, and combines the models constructed for each sub-problem to classify new instances with multiple labels. Although there are many multi-label learning methods, there is still a need to explore methods that can lead to improvement in prediction power. In this work, we propose a new method, called RL (Random Label), based on dataset transformation and combination of classifiers. Six real-world datasets were used to evaluate our method, which was compared to three existing methods. Results were considered promising.*

Resumo. *Em muitos problemas do mundo real, um classificador deve ser capaz de atribuir não somente um mas vários rótulos a um exemplo, como por exemplo predição de falhas em equipamentos, predição de gêneros musicais, etc. Para este tipo de problema, é necessário utilizar um método para aprendizado multirrótulo. Uma abordagem utilizada para o aprendizado multirrótulo é transformar o problema multirrótulo em vários problemas de classificação de rótulo único. Há vários métodos propostos na literatura baseados nessa abordagem, entretanto ainda há espaço para explorar novos métodos com possibilidade de melhora no poder de predição. Neste trabalho, propomos o método RL (Random Label) baseado na abordagem de transformação do problema multirrótulo em problemas multiclasse, e na abordagem de combinação dos classificadores para novas predições. Foram utilizados seis conjuntos de dados naturais para avaliação do método, que foi comparado a três métodos propostos na literatura. Os resultados obtidos foram considerados promissores.*

1. Introdução

Um dos objetivos do aprendizado de máquina é aprender conceitos e padrões a partir de exemplos. Os classificadores construídos pelos algoritmos de aprendizado supervisionado

clássicos rotulam novos exemplos com apenas uma classe, e/ou oferecem probabilidades de um exemplo pertencer às classes do domínio em questão. Entretanto, existem problemas nos quais um exemplo é inerentemente rotulado com mais de uma classe, como por exemplo, rotulamento de textos, vídeos, imagens, músicas, diagnósticos de falhas em equipamentos, etc, 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 um ou vários subproblemas de aprendizado multiclasse. Há vários métodos propostos na literatura baseados nessa abordagem [Shen et al. 2004, Sebastiani 2002, Dimou et al. 2009, Bernardini et al. 2009, Calembo et al. 2011], aplicados a domínios diversos.

Neste trabalho propomos um método de combinação de classificadores que divide o problema de aprendizado multirrótulo em diversos sub-problemas multiclasse. Os múltiplos classificadores construídos são combinados utilizando o método de combinação de classificadores *bagging* [Breiman 1996]. A vantagem desse método está na simplicidade do processo de transformação do problema, associado à complementaridade dos classificadores construídos e combinados. No *bagging* original, múltiplos classificadores são obtidos através de seleção aleatória do conjunto de dados original, criando conjuntos de dados diferentes que são utilizados para aprender múltiplos classificadores. Outras propostas para obter classificadores variados a partir de um mesmo conjunto de dados original existem na literatura, como a seleção aleatória de atributos [Bryll et al. 2003]. A nossa proposta neste trabalho é baseada na seleção aleatória de rótulos, que ao mesmo tempo permite transformar o problema multirrótulo em um problema multiclasse e permite variação no conjunto de dados utilizado para aprender cada classificador. O método proposto foi implementado utilizando a biblioteca Mulan[1][Tsoumakas et al. 2010b], que é baseado na ferramenta Weka [Witten and Frank 2005]. 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, definições e três métodos de aprendizado multirrótulo propostos na literatura que são os mais próximos do algoritmo por nós proposto. Na Seção 3, é descrito o método RL, proposto neste trabalho. Na Seção 4 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 5 é realizada a conclusão e são mencionados possíveis trabalhos futuros.

2. Aprendizado de Máquina Supervisionado Multiclasse e Multirrótulo

Muitos problemas de classificação do mundo real são relativos à associação de apenas um rótulo a um exemplo, denominados problemas de rótulo único. No aprendizado supervisionado de rótulo único, 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 $\mathcal{D}$ fixa, desconhecida e arbitrária, da forma $(\mathbf{x}_i, y_i)$, com $i = 1, ..., N$, para alguma função desconhecida $y = f(x)$. Os x_i são tipicamente vetores da forma $(x_{i1}, ..., x_{iM})$, com valores discretos ou contínuos, onde x_{ij} refere-se ao valor do atributo j, denominado X_j, do exemplo T_i. Quando se trata de problemas de classificação, os valores y_i são tipica-

[1]Disponível em `http://mulan.sourceforge.net`

mente pertencentes a um conjunto discreto de classes L, *i.e* $y \in L = \{l_1, ..., l_{|L|}\}$. Ainda, referem-se ao valor do atributo Y, o qual é frequentemente denominado atributo classe. Quando $|L| > 2$, o problema é denominado multiclasse. Uma descrição de diversos algoritmos de aprendizado de rótulo único pode ser encontrada em [Witten and Frank 2005].

Já no aprendizado de modelos para problemas multirrótulo, que podem estar relacionados a diversos domínios, tais como classificação de imagens, textos, proteínas, genoma, diagnóstico de falhas de equipamentos, dentre outros [Shen et al. 2004, Sebastiani 2002, Dimou et al. 2009, Bernardini et al. 2009], a entrada do algoritmo para aprendizado de um modelo multirrótulo consiste de um conjunto de exemplos S, com N exemplos $T_i, i = 1, ..., N$, escolhidos de um domínio X com uma distribuição $\mathcal{D}$ fixa, desconhecida e arbitrária, da forma $(\mathbf{x}_i, Y_i)$, com $i = 1, ..., N$. L é o conjunto de rótulos possíveis do domínio $\mathcal{D}$, e $Y_i \subseteq L$, *i.e.*, Y_i é o conjunto de rótulos do $i$ésimo exemplo. A saída de um algoritmo de aprendizado supervisionado de modelos multirrótulos é um classificador $\mathbf{h}$, que classifica um exemplo $\mathbf{x}_i$ com o conjunto $Z_i = \mathbf{h}(\mathbf{x}_i)$, o qual é o conjunto de classes preditas por $\mathbf{h}$ para o exemplo $\mathbf{x}_i$.

2.1. Características e 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. Há duas medidas para avaliar características de um conjunto de dados: cardinalidade $Card$ e densidade $Dens$ [Tsoumakas et al. 2010a]. A cardinalidade de um conjunto de dados S é a média do número de rótulos dos exemplos pertencentes a S — $Card(S) = \frac{1}{N}\sum_{i=1}^{N}|Y_i|$; a densidade de um conjunto S é o número médio de rótulos dos exemplos pertencentes a S dividido pelo número de rótulos $|L|$ — $Dens(S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{|L|}$. 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 de rótulos distintos é também importante para muitos métodos baseados em algoritmos de transformação. Sendo assim, é importante observar tais medidas quando se utiliza métodos de aprendizado multirrótulo.

2.2. Medidas de Avaliação

Para avaliar os classificadores multirrótulo, são utilizadas 3 (três) classes de medidas — baseadas em exemplos, baseadas em rótulos e medidas baseadas em ranqueamento [Tsoumakas et al. 2010a]. As medidas baseadas em exemplos são: *Hamming Loss* (Ham), acurácia (Acc), F1 ($F1$) e *Subset Accuracy* ($SubAcc$), definidas pelas Eqs. 1 a 4[2]. Deve ser observado que a medida $SubAcc$ é bastante conservadora, pois requer que o conjunto de rótulos preditos seja exatamente igual ao conjunto de rótulos verdadeiros. Já as medidas baseadas em rótulos utilizadas foram a F e a AUC (*Area Under ROC*) micro e macro. Considerando que ambas são medidas de avaliação binárias do tipo $B(t_p, t_n, f_p, f_n)$, que é calculada baseada no número de falsos positivos (f_p), falsos negativos (f_n), verdadeiros positivos (t_p) e verdadeiros negativos (t_n), e considerando t_{p_l}, t_{n_l}, f_{p_l} e f_{n_l} como sendo os verdadeiros positivos, verdadeiros negativos, falsos positivos

[2]Na Eq. 1, Δ representa a diferença simétrica entre dois conjuntos. Já na Eq. 4, I(verdadeiro) = 1 e I(falso) = 0.

e falsos negativos para o rótulo l, as versões micro e macro dessas medidas são dadas pelas Eqs. 5 e 6, respectivamente. Por fim, as medidas baseadas em ranqueamento utilizadas são *One-Error* ($1Err$) e *Ranking Loss* ($RankLoss$), definidas respectivamente pelas Eqs. 7 e 8[3]. Cada uma das medidas de avaliação descritas pode ser utilizada como uma métrica calculada em cada partição gerada pela técnica de *k-fold cross-validation* para estimar o poder de predição de um classificador multirrótulo.

$$Hamm(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \Delta Z_i|}{|L|} \quad (1) \qquad Acc(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (2)$$

$$F1(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \quad (3) \qquad SubAcc(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N} I(Z_i = Y_i) \quad (4)$$

$$B_{micro}(\mathbf{h}, S) = \frac{1}{|L|}\sum_{i=1}^{|L|} B(t_{p_i}, t_{n_i}, f_{p_i}, f_{n_i}) \quad (5)$$

$$B_{macro}(\mathbf{h}, S) = \frac{1}{|L|} B(\sum_{i=1}^{|L|} t_{p_i}, \sum_{i=1}^{|L|} t_{n_i}, \sum_{i=1}^{|L|} f_{p_i}, \sum_{i=1}^{|L|} f_{n_i}) \quad (6)$$

$$1Err(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\delta(\arg\min_{l \in L} r_i(l)) \quad \text{onde } \delta(l) = \begin{cases} 0, & \text{se } l \notin Y_i \\ 1, & \text{caso contrário.} \end{cases} \quad (7)$$

$$RankLoss(\mathbf{h}, S) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i||\overline{Y}_i|}|A(i)| \quad \text{onde } A(i) = \{(l_a, l_b) : r_i(l_a) > r_i(l_b), (l_a, l_b) \in Y_i \times \overline{Y}_i\} \quad (8)$$

2.3. Descrição de Alguns Métodos de Classificação Multirrótulo

Algumas abordagens para classificação de problemas multirrótulo atuam na transformação do problema em subproblemas binários, como é o caso do método BR, ou ainda transformam o problema multirrótulo em um único problema multiclasse, como é o caso dos métodos LP e SA [Tsoumakas et al. 2010a]. Tais métodos são descritos a seguir. Esses métodos foram utilizados para comparação do nosso método por serem os métodos que mais se aproximam do método RL, proposto neste trabalho.

2.3.1. Método BR — Binary Relevance

Uma possível solução para este tipo de problema é a decomposição do problema multirrótulo em vários subproblemas binários. Um método popular que realiza a decomposição

[3]Nessas medidas, $r_i(l)$ é o ranking predito para o rótulo l referente ao exemplo $\mathbf{x}_i$, e $\overline{Y}_i$ é o conjunto complementar de Y_i em relação ao conjunto L.

em vários problemas binários é denominado *Binary Relevance* — BR —, que foi usado em [Shen et al. 2004]. No método BR, é construído um classificador para cada classe com um mesmo algoritmo de aprendizado de máquina supervisionado. 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 $l_i, i = 1, ..., |L|$. Dado um algoritmo de aprendizado para problemas de classificação de somente um rótulo, é construído um classificador 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 prediga que o exemplo é positivo, o conjunto de rótulos de saída recebe o rótulo ao qual o classificador se refere.

2.3.2. Método LP — Label Powerset

O método *Label Powerset* — LP, proposto em [Read 2008], é um método simples de transformação do problema multirrótulo em um problema multiclasse. Ele considera cada conjunto de rótulos que existe em um conjunto de treinamento multirrótulo como uma das classes de uma nova tarefa de classificação de rótulo único. Por exemplo, dados três rótulos l_1, l_2 e l_3 e um conjunto de treinamento multirrótulo S, o exemplo $\mathbf{x}_1 \in S$ rotulado com l_1 e l_2 passa a ser rotulado, devido à transformação do método LP, com $l_{1,2}$; o exemplo $\mathbf{x}_2 \in S$ rotulado com l_1, l_3 passa a ser rotulado com $l_{1,3}$; o exemplo $\mathbf{x}_3 \in S$ rotulado com l_1 continua sendo rotulado somente com l_1; e assim sucessivamente. Com esse novo conjunto de dados transformado S', um classificador multiclasse $\mathbf{h}$ é induzido.

Dado um novo exemplo $\mathbf{x}$ a ser rotulado, o classificador $\mathbf{h}$ rotula $\mathbf{x}$ com a classe mais provável, que é um conjunto de rótulos. Supondo que $\mathbf{h}$ pode oferecer, além do rótulo mais provável, uma distribuição de probabilidade para todas as classes possíveis de classificação, então o método LP pode oferecer um ranking dos rótulos originais. Como exemplo, vamos supor que a hipótese $\mathbf{h}$ oferece como distribuição de probabilidade: $l_{1,2} = 0.7$, $l_{2,3} = 0.2$ e $l_1 = 0.1$. Então, a probabilidade de $\mathbf{x}$ ser rotulado com $l_1 = 0.7 \times 1 + 0.2 \times 0 + 0.1 \times 1 = 0.8$, a probabilidade de ser rotulado com $l_2 = 0.7 \times 1 + 0.2 \times 1 + 0.1 \times 0 = 0.9$, e a probabilidade de ser rotulado com $l_3 = 0.7 \times 0 + 0.2 \times 1 + 0.1 \times 0 = 0.2$. A partir daí, é possível definir um limiar de probabilidade t para rotular um novo exemplo com um dado rótulo, por exemplo, $t = 0.5$. Neste caso, $\mathbf{x}$ seria rotulado com l_1 e l_2.

2.3.3. Método SA — Seleção Aleatória

Um outro método simples para transformação do problema multirrótulo em um problema multiclasse consiste em substituir o conjunto de rótulos Y_i associado ao exemplo $\mathbf{x}_i$ por um rótulo y_i selecionado aleatoriamente de Y_i. Essa transformação é denominada SA (Seleção Aleatória, ou, do inglês, *select-random*) e é descrita em [Tsoumakas et al. 2010a]. Por exemplo, dados três rótulos l_1, l_2 e l_3 e um conjunto de treinamento multirrótulo S, o exemplo $\mathbf{x}_1 \in S$ rotulado com l_1 e l_2 passa a ser rotulado (aleatoriamente) com l_1; o exemplo $\mathbf{x}_2 \in S$ rotulado com l_1, l_3 passa a ser rotulado (aleatoriamente) com l_2; o exemplo $\mathbf{x}_3 \in S$ rotulado com l_1 continua sendo rotulado somente com l_1; e assim sucessivamente. Com esse novo conjunto de dados transformado S', um classificador multiclasse $\mathbf{h}$ é induzido.

Dado um novo exemplo $\mathbf{x}$ a ser rotulado, o classificador $\mathbf{h}$ rotula $\mathbf{x}$ com a classe mais provável, que é um único rótulo. Supondo que $\mathbf{h}$ pode oferecer, além do rótulo mais provável, uma distribuição de probabilidade para todas as classes possíveis de classificação, então o método SA pode oferecer um ranking dos rótulos originais. Como exemplo, vamos supor que a hipótese $\mathbf{h}$ oferece como distribuição de probabilidade: $l_1 = 0.6$, $l_2 = 0.1$ e $l_3 = 0.3$. A partir daí, é possível definir um limiar de probabilidade t para rotular um novo exemplo com um dado rótulo, por exemplo, $t = 0.2$. Neste caso, $\mathbf{x}$ seria rotulado com l_1 e l_3.

3. O Método RL — Random Label

O método RL, proposto neste trabalho, foi inspirado no método SA de transformação do problema, já que um questionamento quanto ao uso do método SA é a possibilidade de muitos rótulos não serem utilizados na fase de treinamento do classificador multiclasse. Uma maneira de resolver o problema da possibilidade de rótulos não serem selecionados na transformação do conjunto de dados é repetir a transformação do conjunto de dados do SA C vezes e combinar os classificadores induzidos para cada transformação, que é a proposta do nosso método. Um método clássico que realiza combinação de classificadores é o método *Bagging* [Breiman 1996]. O método *Bagging* combina a decisão de classificadores induzidos a partir de subconjuntos de dados amostrados do conjunto S utilizando a técnica *bootstrap*. Aqui, ao invés de utilizar a técnica *bootstrap* para gerar os subconjuntos, vamos utilizar o método SA, gerando subconjuntos com diferentes rótulos escolhidos aleatoriamente.

Mais especificamente, a proposta do método RL é composta de duas fases: a fase de indução dos modelos, ou classificadores multiclasse, e a fase de predição de exemplos.

Indução de modelos multirrótulo: inicialmente, o método RL utiliza como entrada o conjunto de dados $S = \{(\mathbf{x}_i, Y_i), i = 1, ..., N\}$. A seguir, constrói C conjuntos de dados $S_c, c = 1, ..., C$, nos quais são colocados todos os exemplos $\mathbf{x}_i \in S$, porém cada exemplo $\mathbf{x}_i$ é rotulado com algum $y_i \in Y_i$, selecionado aleatoriamente. Deve ser observado que, dessa maneira, a transformação do método SA é repetida C vezes. A seguir, para cada conjunto de dados S_c, é construído um classificador $\mathbf{h}_c$ com algum algoritmo de aprendizado multiclasse de rótulo único.

Predição de exemplos: Os classificadores construídos por algoritmos de aprendizado multiclasse oferecem como saída a classe mais provável, e também uma distribuição de probabilidade ou um grau de pertinência do exemplo a ser classificado para cada rótulo $l \in L$. Dado um novo exemplo $\mathbf{x}$ a ser classificado, calcula-se, para cada rótulo $l \in L$, a média dos graus de pertinência dados por cada classificador $\mathbf{h}_c$ — $\gamma_l = \frac{1}{C} \sum_{c=1}^{C} p(c|\mathbf{x})$. O exemplo $\mathbf{x}$ é então rotulado pelos rótulos cujo γ_l for maior que um valor limiar t dado como parâmetro do método. Deve ser observado que tanto o número de classificadores C quanto o limiar t são parâmetros do método RL.

Para avaliar o método RL, o método foi implementado utilizando a ferramenta Weka e a biblioteca Mulan. O Weka é uma ferramenta livre desenvolvida para auxiliar o processo de mineração de dados, incluindo ferramentas implementadas para tarefas de classificação, aprendizado supervisionado, e outras [Witten and Frank 2005]. Estão presentes nessa ferramenta implementações de métodos de mineração de dados, além de outras ferramentas. Essa ferramenta possui a interessante vantagem de ter sido imple-

mentada na linguagem Java, permitindo portabilidade. A biblioteca Mulan (*Multi-label Learning*) [Tsoumakas et al. 2010b] foi proposta para atender as necessidades dos problemas de aprendizado multirrótulo. A base da biblioteca Mulan é a ferramenta Weka. Essa biblioteca foi utilizada como base para implementação do método RL.

4. Experimentos Realizados e Resultados Obtidos

Nesta seção, apresentamos uma descrição dos experimentos realizados e dos resultados obtidos. Os experimentos foram realizados utilizando os métodos BR, LP, SA e RL. Foram utilizados 3 (três) diferentes algoritmos de aprendizado de máquina para indução dos classificadores multiclasse implementados no Weka: J48 (a implementação do Weka do algoritmo de aprendizado C4.5 para indução de árvores de decisão); NB (o algoritmo de aprendizado Naive Bayes, que utiliza estatística bayesiana para indução dos classificadores); e SMO (um algoritmo para indução de SVM). Foram utilizadas seis bases de dados[4]: Emotions, Genbase, Scene, Yeast, Enron e Medical. Na Tabela 1 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; e $Dens$ é a medida de densidade de rótulo do conjunto de dados.

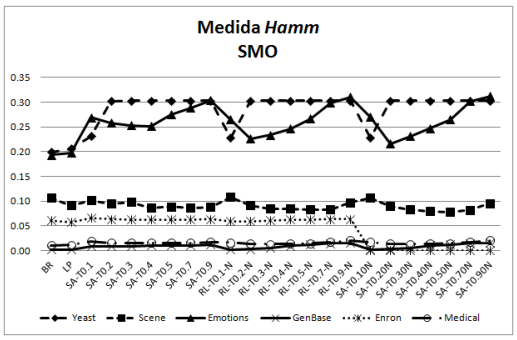
Nome	#Exs.	#At. Disc.	#At. Cont	#Rótulos	$Card$	$Dens$
Emotions	593	0	72	6	1.869	0.311
Genbase	662	1186	0	27	1.252	0.046
Scene	2407	0	294	6	1.074	0.179
Yeast	2417	0	103	14	4.237	0.303
Enron	1000	1001	0	53	3.378	0.064
Medical	978	1449	0	45	1.245	0.028

Tabela 1. Características dos conjuntos de dados

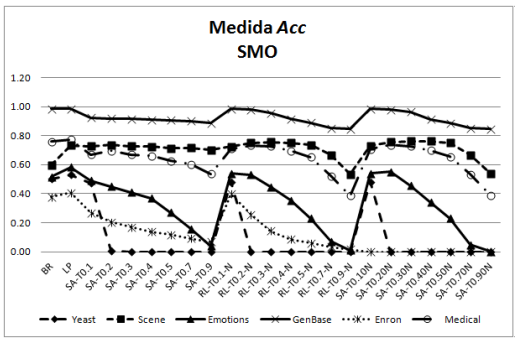
O comportamento do método RL foi avaliado utilizando dois valores distintos de número de classificadores — $C = |L|$ e $C = 10|L|$ —, e 7 (sete) valores distintos de limiar — 0.1, 0.2, 0.3, 0.4, 0.5, 0.7 e 0.9. Todas as medidas de avaliação descritas na Seção 2.2 foram utilizadas. Para avaliar o comportamento de cada algoritmo, foi utilizada a técnica de *k-fold cross-validation*, com $k = 10$, para avaliar os valores das medidas para cada algoritmo. Nas Figuras 1 e 2 são exibidos graficamente os resultados obtidos com todos os métodos utilizando 10-*fold cross-validation*, os quais são BR, LP, SA com variação de limiar $t = 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9$, e RL com todas as combinações de parâmetros, para todas as medidas utilizadas, tendo como algoritmo de aprendizado multiclasse o SMO. Como os resultados obtidos com os outros dois algoritmos (J48 e NB) apresentaram comportamentos semelhantes, por questão de espaço esses resultados não foram exibidos. Podemos observar nesses gráficos que em todos os experimentos com $t = 0.7$ e $t = 0.9$, os resultados foram piores que para os outros valores de t e por isso não foram considerados nos testes de hipóteses descritos a seguir.

Para analisar os resultados e verificar se existe diferença estatisticamente significativa entre os métodos, consideremos diferentes hipóteses nulas para analisar o com-

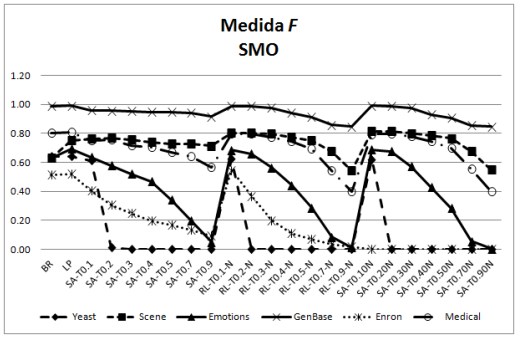
[4]Esses e outros conjuntos de dados estão disponíveis no *site* da biblioteca Mulan — `http://mulan.sourceforge.net/datasets.html`



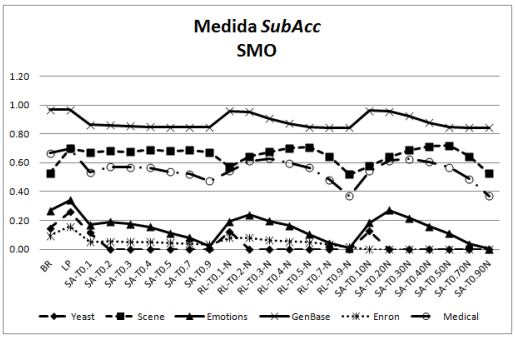
(a) Medida $Hamm$



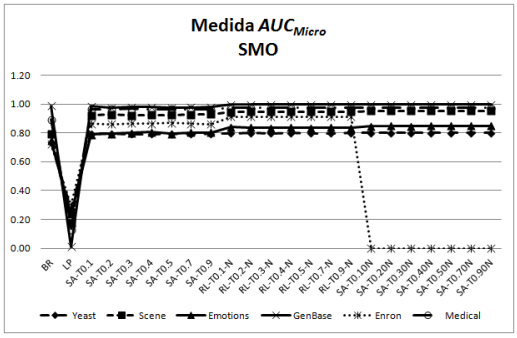
(b) Medida Acc



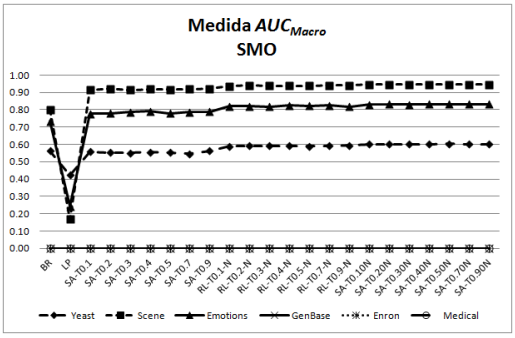
(c) Medida F



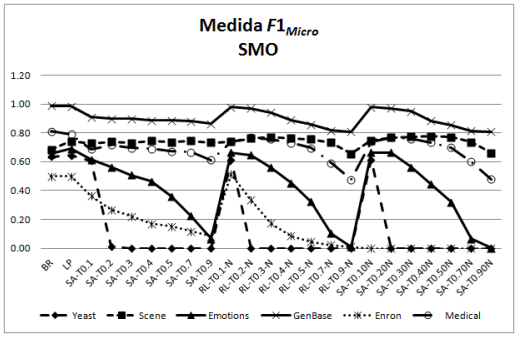
(d) Medida $SubAcc$



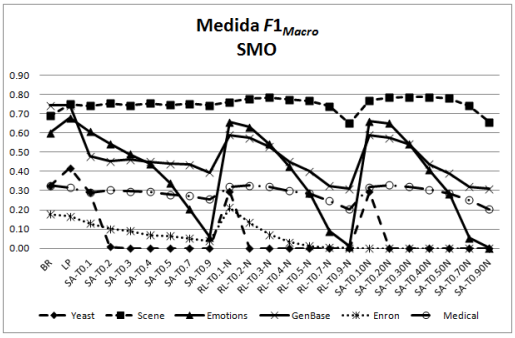
(e) Medida AUC_{Micro}



(f) Medida AUC_{Macro}

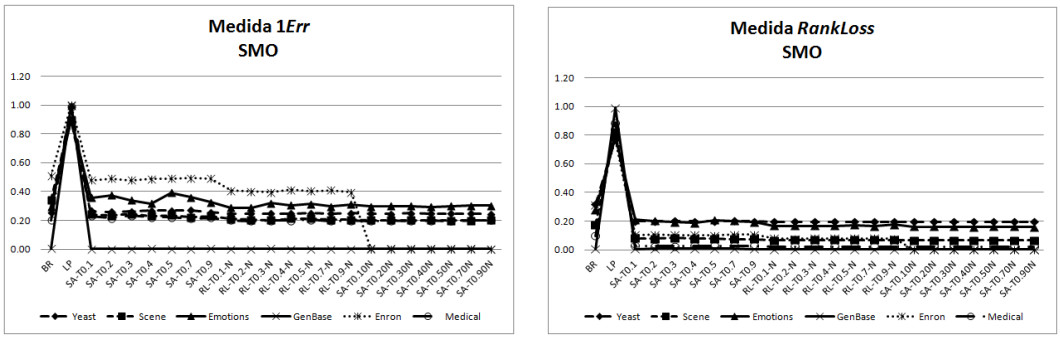


(g) Medida $F1_{Micro}$



(h) Medida $F1_{Macro}$

Figura 1. Resultados Obtidos para todos os Cenários de Experimentação — Medidas Baseadas em Exemplos e Medidas Baseadas em Rótulos



(a) Medida $1Err$ (b) Medida $RankLoss$

Figura 2. Resultados Obtidos para todos os Cenários de Experimentação — Medidas Baseadas em Ranqueamento

portamento das variáveis do método RL em cada hipótese nula, bem como o método RL comparado aos outros métodos. Executamos os testes Wilcoxon[5] quando a hipótese nula é relativa à comparação entre duas possibilidades de valores para a variável aleatória em questão, e Friedman[6] quando a hipótese nula é relativa a mais de dois valores da variável aleatória em questão. Para os testes de hipótese, foram considerados os resultados obtidos utilizando os resultados do 10-*fold cross-validation*. Para executar os testes com os métodos BR, LP e SA, cada execução do método com um algoritmo de aprendizado distinto (J48, NB ou SMO) foi considerada como sendo uma execução independente de cada um dos métodos. Para executar os testes com o método RL, cada combinação do limiar t com o número de classificadores C e com o algoritmo de aprendizado (J48, NB ou SMO) também foi considerada uma execução independente do método RL. Por restrição de espaço, apenas os resultados dos testes de hipótese são reportados aqui. A seguir, descrevemos 4 (quatro) hipóteses nulas que consideramos e testamos.

Hipótese H0: Para $C = |L|$ e $C = 10|L|$ os resultados obtidos com o método RL são comparáveis. Foi realizado o teste Wilcoxon, que rejeitou a hipótese nula com 95% de confiança para as medidas F, $SubAcc$, $1Err$ e $RankLoss$, sendo que RL com $C = |L|$ apresenta melhores resultados para essas medidas e para esses conjuntos de dados. Já para a medida AUC_{Micro}, a hipótese nula é rejeitada, e RL com $C = 10|L|$ apresenta melhores resultados para esses conjuntos de dados com 95% de confiança. Para todas as outras medidas, a hipótese nula não pode ser rejeitada. Como o método RL com $C = 10|L|$ apresentou melhores resultados para somente uma medida em 10 (dez), e o método RL com $C = |L|$ apresentou melhores resultados para 4 (quatro) medidas em 10 (dez), e levando em consideração que o custo computacional do método RL com $C = 10|L|$ é maior que com $C = |L|$, podemos afirmar que o método RL com $C = |L|$ apresenta melhores resultados que o método RL com $C = 10|L|$.

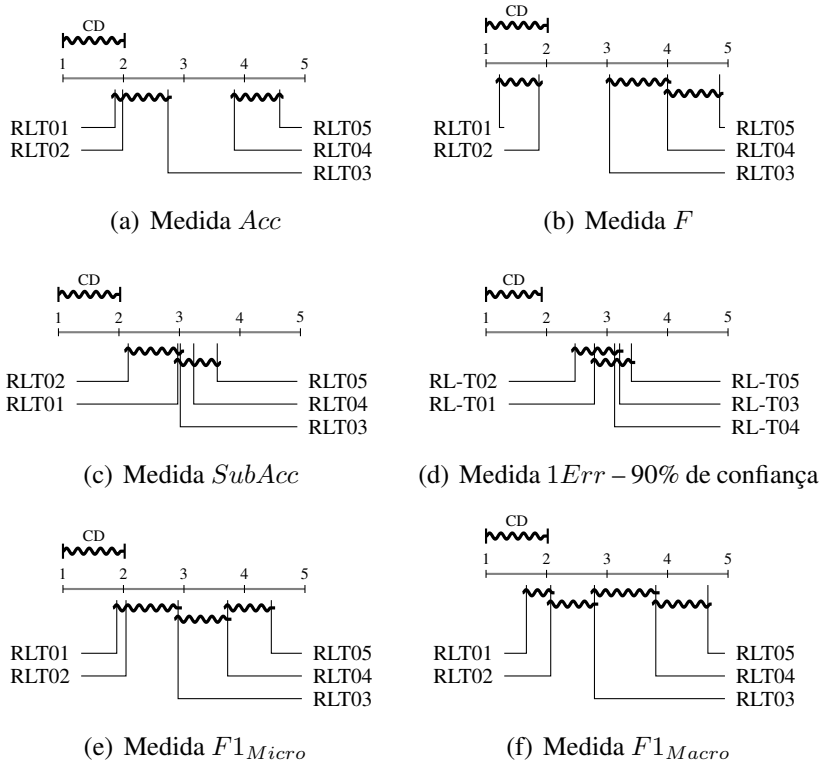
Hipótese H1: Os métodos SA e RL considerando $C = |L|$ são comparáveis. Foi realizado o teste Wilcoxon, que rejeitou a hipótese nula com 95% de confiança segundo as medidas $Hamm$, AUC_{Micro}, $1Err$ e $RankLoss$, e ainda o método RL obteve melhores resultados para as 4 (quatro) medidas. Para todas as outras medidas, a hipótese nula não

[5]O teste Wilcoxon é uma alternativa não paramétrica para o teste t-pareado para comparação de dois algoritmos [Demšar 2006].

[6]O teste Friedman é uma alternativa não paramétrica para o teste ANOVA [Demšar 2006].

foi rejeitada. Como obtivemos melhores resultados em 4 (quatro) das 10 (dez) medidas, podemos afirmar que o método RL obteve melhores resultados que o método SA.

Hipótese H2: RL com $t = 0.1$, $t = 0.2$, $t = 0.3$, $t = 0.4$ e $t = 0.5$ são comparáveis utilizando $C = |L|$. Para facilitar a nomenclatura, o método RL foi renomeado considerando cada um dos valores de t, ou seja, cada combinação de RL com o valor de t passa a se chamar RL-T01, RL-T02, RL-T03, RL-T04 e RL-T05. O teste Friedman foi realizado para essa hipótese, que rejeitou a hipótese nula H2 para as medidas Acc, F, $SubAcc$, $F1_{Mic}$ e $F1_{Mac}$ com 95% de confiança, e para a medida $1Err$ com 90% de confiança. Nas Figuras 3(a) a 3(f) são exibidos os resultados do teste Nemenyi para essas medidas. Podemos observar, nessas figuras, que o método RL para $t = 0.1$ e $t = 0.2$ são os melhores ranquados nas 6 (seis) medidas, porém somente para a medida F há diferença significativa entre os valores de t. Devido a esse indicativo, podemos afirmar que o método RL com $t = 0.1$ e $t = 0.2$ apresenta melhores resultados, e esses valores de t foram selecionados para o teste de hipótese H3, descrito a seguir.



(a) Medida Acc (b) Medida F

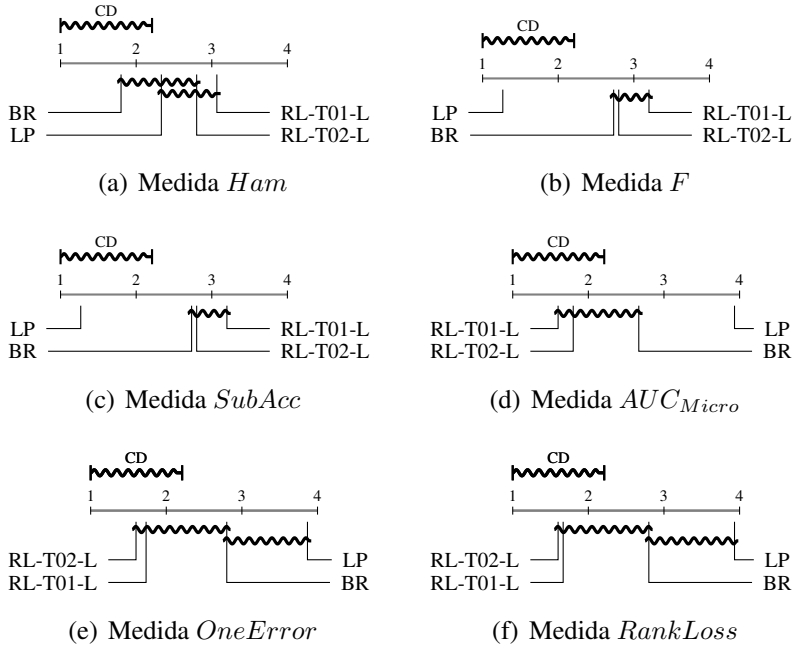
(c) Medida $SubAcc$ (d) Medida $1Err$ – 90% de confiança

(e) Medida $F1_{Micro}$ (f) Medida $F1_{Macro}$

Figura 3. Teste de Hipótese H2

Hipótese H3: Os métodos LP, BR e RL são comparáveis, considerando os resultados do método RL com $C = |L|$ e $t = 0.1$ — RL-T01-L — e $C = |L|$ e $t = 0.2$ — RL-T02-L. O teste Friedman foi executado, que rejeitou a hipótese nula com 95% de confiança para as medidas F, $Hamm$, $SubAcc$, AUC_{Micro}, $1Err$ e $RankLoss$. Nas Figuras 4(a) a 4(f) são exibidos os resultados do teste Nemenyi para essas medidas. Nessas figuras, podemos observar que em 3 (três) das 6 (seis) medidas nas quais a hipótese nula é rejeitada — AUC_{Micro}, $1Err$ e $RankLoss$ —, o método RL com os parâmetros utilizados para comparação ficam melhor ranqueados, porém para duas medidas — F e $SubAcc$ — o método LP apresenta melhores resultados com diferença significativa. As-

sim, podemos afirmar que, dependendo da medida utilizada, o método RL pode apresenatr ou não melhores resultados em relação aos métodos BR e LP.



(a) Medida Ham (b) Medida F

(c) Medida $SubAcc$ (d) Medida AUC_{Micro}

(e) Medida $OneError$ (f) Medida $RankLoss$

Figura 4. Teste de Hipótese H3

Por fim, analisamos se há alguma relação entre a cardinalidade e densidade dos conjuntos de dados com os resultados obtidos. É esperado que os valores na medida $SubAcc$ sejam baixos para conjuntos de dados com alta cardinalidade, pois é mais difícil acertar todo o conjunto de rótulos associado a um exemplo. De fato, para os conjuntos de dados utilizados, os valores de $SubAcc$ para os conjuntos de dados Yeast ($Card = 4.237$), Enron ($Card = 3.378$) e Emotions ($Card = 1.869$) apresentaram os menores valores de $SubAcc$ — Figura 1(d). Analisando os conjuntos de dados com baixa densidade — GenBase e Medical —, os piores resultados para essas bases foram obtidos para a medida $Hamm$, porém nenhuma relação com somente os dois conjuntos de dados em relação às medidas foi observada. A baixa densidade interfere na dispersão dos rótulos entre os exemplos, o que dificulta o aprendizado dos modelos multirrótulo. Mais estudos quanto a essas relações necessitam ser realizadas.

5. Conclusões e Trabalhos Futuros

Neste trabalho, foi proposto um método para construção de classificadores multirrótulo, denominado RL. O método RL é baseado no método SA para problemas multirrótulo e no método *bagging* para combinação de classificadores. O método foi implementado utilizando as bibliotecas Mulan e Weka, na linguagem Java. Foram utilizados 6 (seis) conjuntos de dados naturais para avaliar o método proposto. Pudemos observar nos resultados obtidos dos experimentos realizados que o RL apresenta melhores resultados em relação aos métodos BR, LP e SA para algumas medidas de avaliação de classificadores multirrótulo.

Como trabalho futuro, pretendemos investigar, com maior profundidade, a relação

entre a cardinalidade e a densidade de conjuntos de dados multirrótulo para construção de classificadores multirrótulo, incluindo o método RL.

Agradecimentos

As autoras agradecem ao Prof. Ronaldo Cristiano Prati (UFABC) e ao Prof. Alexandre Plastino (UFF) por suas valiosas observações e contribuições, a Jean Metzz (USP), pelo auxilio para execução dos testes de hipótese, e aos revisores por suas contribuições.

Referências

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

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Bryll, R., Gutierrez-Osuna, R., and Quek, F. (2003). Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, 36:1291–1302.

Calembo, K. N., Bernardini, F. C., and Martins, C. B. (2011). Proposta de um método de combinação de classificadores para construção de classificadores multi-rótulo. In *Conferência Latinoamericana de Informática — CLEI'2011*.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Machine Learning Research*, 7:1–30.

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

Read, J. (2008). A pruned problem transformation method for multi-label classification. In *Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, pages 143–150.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Shen, X., Boutell, M., Luo, J., and Brown, C. (2004). Multi-label machine learning and its application to semantic scene classification. In *Proc. 2004 Int. Symposium on Electronic Imaging – EI 2004*, pages 18–22.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010a). *Data Mining and Knowledge Discovery Handbook*, chapter Mining Multi-label Data. Springer, 2nd edition.

Tsoumakas, G., Vilcek, J., Spyromitros, E., and Vlahavas, I. (2010b). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414.

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