X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará

REDES NEURAIS ARTIFICIAIS APLICADAS A PROBLEMAS DE CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO

Ricardo Cerri, Rodrigo C. Barros e André Carlos P. L. F. de Carvalho
Departamento de Ciências de Computação - Universidade de São Paulo - São Carlos - ICMC/USP
{cerri,rcbarros,andre}@icmc.usp.br

Resumo – Em problemas de classificação hierárquica multirrótulo, cada exemplo pode ser classificado em duas ou mais classes simultaneamente, diferentemente de problemas de classificação convencionais. Adicionalmente, as classes envolvidas nesses problemas são estruturadas em uma hierarquia, que pode ser uma árvore ou um grafo acíclico direcionado (DAG). Dessa forma, um exemplo pode ser atribuído a dois ou mais caminhos de uma estrutura hierárquica formada por centenas e até milhares de classes, dificultando muito o problema de classificação. Muitos métodos têm sido propostos para solucionar tais problemas, alguns deles utilizando um único classificador para lidar com todas as classes simultaneamente (métodos globais), e outros utilizando vários classificadores para decompor o problema original em vários subproblemas (métodos locais). Este trabalho propõe um método local para classificação hierárquica multirrótulo utilizando redes neurais artificiais. O método é chamado HMC-LMLP (*Hierarchical Multi-label Classification with Local Multi-Layer Perceptron*), e utiliza uma rede *Multi-Layer Perceptron* (MLP) associada a cada nível da hierarquia. As predições feitas em um nível são então utilizadas como entrada para outra MLP responsável pelas predições no próximo nível. São utilizados dois algoritmos para o treinamento das MLPs, o algoritmo *Back-propagation* e o algoritmo *Resilient back-propagation*. Adicionalmente, além da medida de erro convencional, uma medida de erro específica para problemas multirrótulo é utilizada para o treinamento das redes. O método é comparado com outros dois métodos locais considerados estado da arte para problemas de classificação hierárquica multirrótulo, utilizando conjuntos de dados relacionados à predição de funções de proteínas. De acordo com os resultados experimentais, o método proposto obteve resultados preditivos competitivos, o que sugere as redes neurais artificiais como alternativas promissoras para tratar problemas de classificação hierárquica multirrótulo.

Palavras-chave – Aprendizado de máquina, redes neurais, classificação hierárquica multirrótulo, predição de funções de proteínas.

1 INTRODUÇÃO

Na maioria dos problemas de classificação descritos na literatura, um classificador atribui apenas uma classe a um dado exemplo, e as classes envolvidas no problema não são estruturadas hierarquicamente. Entretanto, em muitos problemas de classificação reais, como por exemplo a predição de funções de proteínas, classes podem ser divididas em subclasses ou agrupadas em superclasses. Nesses casos, as classes formam uma estrutura hierárquica, geralmente uma árvore ou um grafo acíclico direcionado (DAG). Esses problemas são conhecidos na literatura de Aprendizado de Máquina (AM) como problemas de classificação hierárquica, no qual exemplos são atribuídos a classes associadas a nós pertencentes a uma hierarquia.

Duas abordagens principais têm sido utilizadas para tratar problemas de classificação hierárquica, chamadas local (*top-down*) e global (*one-shot*). A abordagem local utiliza algoritmos de classificação convencionais para formar uma árvore de classificadores, que são então utilizados de maneira *top-down* para a classificação de exemplos. Inicialmente, a classe mais genérica de um exemplo é predita. Essa classe é localizada no primeiro nível hierárquico, e é então utilizada para reduzir o conjunto de possíveis classes do exemplo no próximo nível, ou seja, somente as subclasses da classe predita no primeiro nível são utilizadas para o treinamento no segundo nível. Assim, quando um exemplo é atribuído a uma classe não folha da hierarquia, ele é posteriormente classificado em uma subclasse dessa classe. Uma desvantagem do método local é que, conforme a hierarquia é percorrida em direção às folhas, erros de classificação são propagados para os níveis mais profundos, a não ser que algum procedimento seja adotado para evitar esse problema.

De acordo com Silla e Freitas [2], o aspecto mais importante da abordagem local é que a hierarquia de classes é considerada utilizando informações locais de diferentes maneiras. Baseado na maneira com que essas informações são utilizadas, três principais diferentes grupos de métodos locais podem ser definidos: métodos que utilizam um classificador local por nó, um classificador local por nó pai, e um classificador local por nível. No primeiro grupo, um classificador binário é treinado para cada nó da hierarquia de classes, exceto para o nó raiz. O segundo grupo treina um classificador multi-classe para cada nó pai da hierarquia, ou então utiliza alguma técnica decomposicional com classificadores binários, como o *um-contra-todos* [3] ou *Support Vector Machines* (SVM) [4], para fazer a distinção entre suas subclasses. O último grupo treina um classificador multi-classe para cada nível hierárquico, sendo cada classificador responsável pelas predições em seu nível correspondente.

Há também muitos problemas de classificação nos quais os dados são estruturados de maneira mais complexa, pois além das classes serem estruturadas em uma hierarquia, um exemplo pode pertencer a mais de uma classe em um mesmo nível hierárquico. Esses problemas são conhecidos como problemas de classificação hierárquica multirrótulo, e são muito comuns, por exemplo, em

problemas de classificação de funções de genes e proteínas [5–7, 9–12]. Em problemas de classificação hierárquica multirrótulo, um exemplo pode ser atribuído a dois ou mais caminhos de uma hierarquia de classes. Dado um espaço de exemplos $\mathbf{X}$, o objetivo do processo de treinamento é encontrar uma função que mapeie cada exemplo $\mathbf{x}_i$ em um conjunto de classes, respeitando as restrições da estrutura hierárquica, e otimizando algum critério de qualidade.

Um exemplo de problema hierárquico multirrótulo estruturado como uma árvore é ilustrado na Figura 1. Na figura, um exemplo é atribuído a dois caminhos da hierarquia, formados pelas classes 11.02.03.01, 11.02.03.04, 11.06.01 e todas as suas superclasses. Quando uma predição é feita em algum nó interno da hierarquia, uma subárvore é gerada. No caso de um problema hierárquico simples-rótulo, essa subárvore é reduzida a um caminho. A figura mostra uma predição para um exemplo, atribuindo a ele três nós folhas: 11.02.03.01, 11.02.03.04 e 11.06.01. Todas as superclasses dessas classes também são atribuídas ao exemplo, de maneira que a classificação final seja uma subárvore formada por três caminhos da hierarquia.

É importante perceber a diferença entre problemas hierárquicos simples-rótulo e problemas multirrótulo não hierárquicos. Um problema hierárquico simples-rótulo pode ser visto como multirrótulo de maneira trivial, considerando que um caminho de uma hierarquia possui mais de uma classe. Quando o caminho 11/11.06/11.06.01 é atribuído a um exemplo, essa predição significa que o exemplo pertence às classes 11, 11.06 e 11.06.01. Entretanto, na literatura, um problema hierárquico é considerado multirrótulo somente quando classes de mais de um caminho da hierarquia são atribuídos a um exemplo.

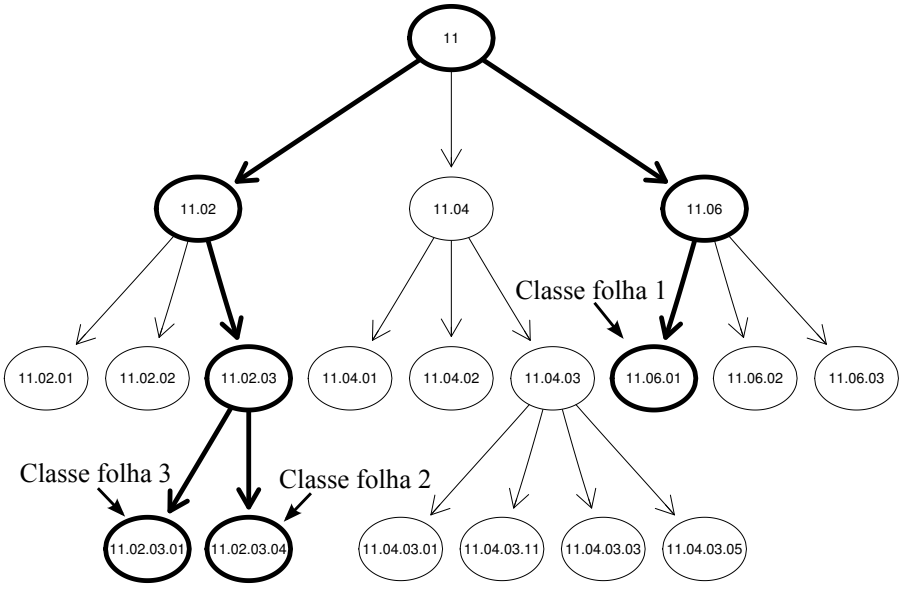


Figura 1: Exemplo de hierarquia estruturada como uma árvore.

Este trabalho propõe um método local para problemas de classificação hierárquica multirrótulo chamado HMC-LMLP (*Hierarchical Multi-label Classification with Local Multi-Layer Perceptron*). O método utiliza uma rede neural *Multi-Layer Perceptron* (MLP) associada a cada nível da hierarquia, sendo cada MLP responsável pelas predições em seu respectivo nível. As predições em um nível são então utilizadas como entrada para a rede neural responsável pelas predições no próximo nível. Para o treinamento das redes neurais, dois algoritmos são utilizados: o algoritmo *Back-propagation* [13] e o algoritmo *Resilient back-propagation* [14]. Outra medida de erro proposta especificamente para problemas multirrótulo [15] também foi utilizada no processo de treinamento das MLPs. O método é comparado com métodos locais de árvore de decisão considerados estado da arte em problemas de classificação hierárquica multirrótulo, utilizando conjuntos de dados de predições de funções de proteínas. Os experimentos sugerem a viabilidade do método proposto considerando que foram obtidos resultados preditivos competitivos.

2 TRABALHOS RELACIONADOS

Vários trabalhos têm sido propostos para tentar solucionar problemas de classificação hierárquica multirrótulo. Esta seção apresenta alguns desses trabalhos, organizados de acordo com a taxonomia proposta em [2], que descreve um algoritmo de classificação de acordo com uma 4-tupla $< \Delta, \Xi, \Omega, \Theta >$, em que: Δ indica que o algoritmo é hierárquico simples-rótulo (SPP) ou hierárquico multirrótulo (MPP); Ξ indica o tipo de predição do algoritmo, se obrigatória em nós folhas (MLNP) ou não obrigatória em nós folhas (NMLNP); Ω: indica a estrutura hierárquica na qual o algoritmo pode ser aplicado - T (árvore) ou D (DAG); e Θ: indica a abordagem utilizada pelo algoritmo na taxonomia proposta, se um classificador local por nó (LCN), por nível (LCL), por nó pai (LCPN) ou global (GC).

Um método global é proposto em [9], baseado do algoritmo indutor de árvores de decisão C4.5 [16], e aplicado à predição de funções de genes. Os autores modificaram a fórmula da entropia do algoritmo C4.5 original, utilizando a soma das entropias de todas as classes e incorporando também informações sobre a hierarquia de classes. A entropia é utilizada para decidir a melhor divisão na árvore de decisão, ou seja, o melhor atributo a ser colocado em nó da árvore. O método é categorizado como $< MPP, NMLNP, T, GC >$.

Em [17], um método baseado na abordagem LCN é proposto. O método utiliza uma hierarquia de classificadores SVM para a predição de funções de proteínas estruturadas de acordo com a hierarquia da *Gene Ontology* (GO) [18]. Os classificadores são treinados para cada classe separadamente, e então as predições são combinadas utilizando um modelo de redes bayesianas [19]. O método é categorizado como $< MPP, NMLNP, D, LCN >$.

Em [20], foi proposta uma combinação (*ensemble*) de classificadores para utilização da hierarquia da GO, estendendo o método proposto em citeBarutcuoglu2006. O *ensemble* é baseado em três diferentes métodos: (i) o treinamento de um classificador SVM para cada nó da GO; (ii) a combinação dos classificadores SVM utilizando redes bayesianas para correção das predições, de acordo com os relacionamentos hierárquicos da GO; e (iii) a indução de um classificador Naive Bayes [21] para cada termo da GO, para combinar as predições feitas pelos classificadores SVM independentes. O método é categorizado como $< MPP, NMLNP, D, LCN >$.

Redes neurais artificiais foram utilizadas como classificadores base em um método chamado *HMC-Label-Powerset* [12]. O método é uma adaptação de um método multirrótulo chamado Label-Powerset [24, 25]. Em cada nível hierárquico, o método HMC-LP combina as classes atribuídas a um exemplo para formar uma nova e única classe, transformando o problema hierárquico multirrótulo original em um problema hierárquico simples-rótulo. O método é categorizado como $< MPP, MLNP, T, LCPN >$.

Em [5], três métodos baseado no conceito de *Predictive Clustering Trees* (PCT) são comparados utilizando conjuntos de dados relacionados à genômica funcional. Os autores utilizaram o método Clus-HMC [26], que induz uma única árvore de decisão, com outros dois métodos chamados Clus-HSC and Clus-SC. O método Clus-SC treina uma árvore de decisão para cada classe, ignorando os relacionamentos hierárquicos, e o método Clus-HSC explora os relacionamentos hierárquicos entre as classes para induzir uma árvore de decisão para cada nó da hierarquia. O método Clus-HMC é categorizado como $< MPP, NMLNP, D, GC >$, enquanto os métodos Clus-HSC e Clus-SC são categorizados como $< MPP, NMLNP, D, LCN >$. O método Clus-HMC também é utilizado como classificador base em um método de combinação utilizado em [27]. Os autores utilizaram a técnica *Bagging* [28] para treinar diferentes classificadores. O novo método, chamado Clus-HMC-ENS, é categorizado como $< MPP, NMLNP, D, GC >$.

3 HMC-LMLP

O método HMC-LMLP (*Hierarchical Multi-label Classification with Local Multi-Layer Perceptron*), categorizado como $< MPP, NMLNP, D, LCL >$, incrementalmente treina uma rede neural MLP para cada nível hierárquico. Primeiramente, uma MLP é treinada para o primeiro nível. Essa rede é formada por uma camada de entrada, uma camada escondida, e uma camada de saída. Após o fim do processo de treinamento, duas novas camadas são adicionadas à primeira MLP para que seja iniciado o treinamento no segundo nível hierárquico. Assim, as saídas da rede responsável pelas predições no primeiro nível são utilizadas como entrada para a camada escondida da rede responsável pelas predições no segundo nível. Esse processo é repetido para todos os níveis hierárquicos. Deve-se lembrar que cada camada de saída possui tantos neurônios quanto for a quantidade de classes de seu nível correspondente, ou seja, cada neurônio é responsável pela predição de uma classe.

A Figura 2 apresenta uma ilustração da arquitetura da rede neural do método HMC-LMLP para uma hierarquia de dois níveis. Como pode ser observado, a rede é completamente conectada. Quando a rede MLP associada a um nível hierárquico específico está sendo treinada, os pesos sinápticos das redes associadas aos níveis anteriores não são ajustados, pois seus ajustes já ocorreram nas fases anteriores do treinamento da rede. Na fase de teste, para classificar um exemplo, um limiar de ativação é aplicado a cada camada de saída correspondente a um nível hierárquico. Os neurônios de saída que tiverem valores maiores ou iguais ao dado limiar são ativados, indicando que suas classes correspondentes estão sendo preditas.

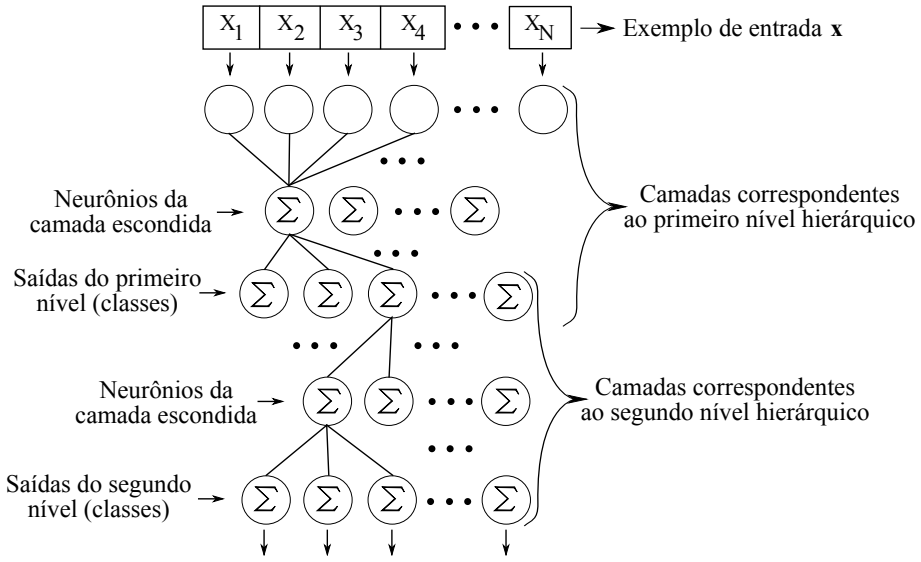


Figura 2: Arquitetura do método HMC-LMLP para uma hierarquia de dois níveis.

É esperado que diferentes valores de limiares resultem em diferentes classes sendo preditas. Como a função de ativação utilizada em cada neurônio é a sigmoidal logística, as saídas dos neurônios têm seus valores entre 0 e 1. Quanto maior for o valor de limiar utilizado, menor é o número de classes preditas. Reciprocamente, quanto menor o valor de limiar utilizado, maior o número de classes preditas.

Depois da classificação final de novos exemplos, uma fase de pós-processamento é utilizada para corrigir inconsistências que podem ter ocorrido durante a classificação, ou seja, quando uma classe é predita sem que sua superclasse seja predita. Essas inconsistência podem ocorrer porque todas as MLPs são treinadas utilizando o mesmo conjunto de exemplos. Em outras palavras, os exemplos utilizados para treinamento em um dado nível não são filtrados de acordo com as classes em que foram classificados no nível anterior. A fase de pós-processamento garante que apenas predições consistentes sejam feitas, por meio da remoção das classes que foram preditas sem que suas superclasses tenham sido preditas.

Qualquer algoritmo de treinamento pode ser utilizado para a indução das MLPs do método HMC-LMLP. Neste trabalho, foram utilizados o algoritmo *Back-propagation* [13] e o algoritmo *Resilient back-propagation* [14]. O último tenta eliminar a influência do tamanho da derivada parcial na atualização dos pesos, considerando apenas o sinal da derivada para indicar a direção da atualização dos pesos sinápticos.

Adicionalmente, foi investigado o desempenho das MLPs com a utilização de duas medidas de erro durante o treinamento das redes: a medida de erro convencional (saída desejada − saída obtida) e uma medida de erro específica para o treinamento de redes neurais em problemas multirrótulo, proposta por Zhang e Zhou [15], dada por:

$$E = \sum_{i=1}^{N} \left[\frac{1}{|C_i||\widehat{C}_i|} \sum_{(l,m)\in C_i \times \widehat{C}_i} \left[\exp(o_m^i - o_l^i)) \right] \right] \tag{1}$$

em que N é o número de exemplos, C_i o conjunto de classes positivas de um exemplo $\mathbf{x}_i$, $\widehat{C}_i$ o complemento de C_i, e o_k é a saída do k-ésimo neurônio, correspondente à classe c_k. O erro (e_j) do neurônio j é definido como:

$$e_j = \begin{cases} \left(\frac{1}{|C_i||\widehat{C}_i|} \sum_{m\in\widehat{C}_i} \exp(o_m - o_j) \right) & \text{se } c_j \in C_i \\ \left(- \frac{1}{|C_i||\widehat{C}_i|} \sum_{l\in C_i} \exp(o_j - o_l) \right) & \text{se } c_j \in \widehat{C}_i \end{cases} \tag{2}$$

A medida de erro multirrótulo considera a correlação entre as diferentes classes de um exemplo. A principal característica da medida apresentada na Equação (1) é que ela é focada na diferença entre as saídas das MLPs para classes que pertencem e para classes que não pertencem a um dado exemplo, ou seja, ela tenta capturar as nuances do problema multirrótulo em questão.

O desenvolvimento do método HMC-LMLP é motivado pelo fato de que as redes neurais podem ser naturalmente consideradas classificadores multirrótulo, já que suas camadas de saída podem predizer duas ou mais classes simultaneamente. Essa característica é particularmente interessante, pois possibilita a utilização de apenas um classificador por nível hierárquico. A maioria dos métodos tenta utilizar um único classificador para fazer a distinção entre todas as classes, empregando mecanismo internos complexos, ou então decompõem os problemas originais em vários subproblemas por meio da utilização de vários classificadores por nível, podendo perder informações importantes durante esse processo [2].

4 METODOLOGIA EXPERIMENTAL

Para a realização dos experimentos, foram utilizados quatro conjuntos de dados, disponíveis gratuitamente[1], referentes às funções protéicas do organismo *Saccharomyces cerevisiae*. Tais conjuntos são relacionados a dados de Bioinformática, como fenótipo e expressão gênica. Eles estão organizados em uma estrutura de árvore, de acordo com o esquema FunCat de classificação, desenvolvido pelo *Munich Information Center for Protein Sequences* (MIPS) [23]. A Tabela 1 mostra as características dos dados de treinamento, validação e de teste utilizados.

Tabela 1: Número de atributos ($|A|$), número de classes ($|C|$), número total de exemplos (Total) e número de exemplos multirrótulo (Multirrótulo) dos quatro conjuntos de dados utilizados.

Conjunto de Dados	$\|A\|$	$\|C\|$	Treinamento		Validação		Teste	
			Total	Multirrótulo	Total	Multirrótulo	Total	Multirrótulo
Cellcycle	77	499	1628	1323	848	673	1281	1059
Church	27	499	1630	1322	844	670	1281	1057
Derisi	63	499	1608	1309	842	671	1275	1055
Eisen	79	461	1058	900	529	441	837	719

O desempenho do método proposto é comparado com o de dois algoritmos locais de árvores de decisão, considerados estado-da-arte para problemas de classificação hierárquica multirrótulo, introduzidos em [5]: Clus-HSC, método que explora os relaci-

[1] `http://www.cs.kuleuven.be/~dtai/clus/hmcdatasets.html`

onamentos hierárquicos para construção de árvores de decisão para cada nó hierárquico; e Clus-SC, método que gera árvores de decisão binárias para cada classe da hierarquia. Tais métodos são baseados no conceito de *Predictive Clustering Trees (PCT)* [30].

Para realização da análise de desempenho, são utilizadas curvas Precisão-Revocação (curvas PR), que refletem a precisão de um classificador em função de sua revocação, e apresentam uma descrição informativa do desempenho de cada método ao lidar com conjuntos de dados altamente assimétricos [31] (justamente o caso de problemas de classificação hierárquica multirrótulo). As medidas de precisão hierárquica (hP) e revocação hierárquica (hR) (Equações 3 e 4), utilizadas para construção das curvas PR, assumem que um exemplo pertence não somente a uma classe, mas também a todas as suas classes ancestrais [32]. Desta forma, dado um exemplo $(\mathbf{x}_i, C_i)$, $\mathbf{x}_i \in \mathbf{X}$, C'_i sendo o conjunto de classes preditas para tal exemplo e C_i o conjunto de classes reais, C'_i e C_i podem ser modificados de forma a conter suas respectivas classes ancestrais: $\widehat{C}'_i = \bigcup_{c_k \in C'_i} Ancestrais(c_k)$ e $\widehat{C}_i = \bigcup_{c_l \in C_i} Ancestrais(c_l)$, onde $Ancestrais(c_k)$ é o conjunto de ancestrais da classe c_k.

$$hP = \frac{\sum_i |\widehat{C}_i \cap \widehat{C}'_i|}{\sum_i |\widehat{C}'_i|} \tag{3}$$

$$hR = \frac{\sum_i |\widehat{C}_i \cap \widehat{C}'_i|}{\sum_i |\widehat{C}_i|} \tag{4}$$

Uma curva PR é obtida por meio da variação de valores do limiar de ativação dos neurônios, que é aplicado aos neurônios das camadas de saída, gerando diferentes valores de hP e hR. As saídas dadas pelos neurônios são representadas por vetores de valores reais, onde cada valor denota o grau de pertinência de um dado exemplo a uma dada classe. Para cada valor de limiar, um ponto da curva PR é obtido, e a curva final é traçada por meio da interpolação destes pontos [31]. As áreas sob tais curvas ($AU(\overline{PRC})$) são aproximadas por meio da soma das áreas trapezoidais entre cada par de pontos, e são utilizadas para avaliar o desempenho dos métodos. Quanto maior o valor de $AU(\overline{PRC})$ de determinado método, melhor é o seu desempenho preditivo.

Para verificar a significância estatística dos resultados, foram aplicados os testes de Friedman e Nemenyi, recomendados para comparações envolvendo diferentes conjuntos de dados e vários classificadores [33]. Assim como feito em [5], 2/3 de cada conjunto de dados foram utilizados para treinamento e validação dos algoritmos, e 1/3 para teste.

O método proposto foi executado com o número de neurônios de cada camada escondida igual a 50% do número de neurônios da camada de entrada correspondente. Como cada MLP é composto por três camadas (de entrada, escondida e de saída), os valores da taxa de treinamento utilizados foram 0.2 e 0.1 para as camadas escondida e de saída, respectivamente. Da mesma forma, os valores utilizados para a constante de momento foram 0.1 e 0.05 para estas mesmas camadas. Nenhuma tentativa de otimização de parâmetros foi realizada. O processo de treinamento durou um máximo de 1000 ciclos e, a cada 10 ciclos, as curvas PR foram calculadas para o conjunto de validação. O modelo que obteve o melhor desempenho no conjunto de validação foi então avaliado nos dados de teste. Para cada conjunto de dados, o método proposto foi executado 10 vezes, cada execução iniciando aleatoriamente os valores dos pesos sinápticos. O valor final de $AU(\overline{PRC})$ foi obtido pela média dos valores das execuções individuais.

Os métodos Clus-HSC e Clus-SC foram executados uma vez cada utilizando seus valores padrão de configuração, como descrito em [5].

5. ANÁLISE EXPERIMENTAL

A Figura 3 apresenta exemplos de curvas PR obtidas por todos os métodos, e a Tabela 2 apresenta suas $AU(\overline{PRC})$, juntamente com o número de vezes em que o desempenho de cada método aparece entre as três melhores $AU(\overline{PRC})$ (parte inferior da tabela). A Tabela 2 também apresenta, para o método HMC-LMLP, os desvios padrões e o número de ciclos necessários para obter tais resultados. Deve ser observado que os gráficos da Figura 3, para o método HMC-LMLP, foram obtidos a partir de uma única execução do algoritmo. Assim, eles são mostrados para exemplificação e não representam as médias das $AU(\overline{PRC})$ mostradas na Tabela 2.

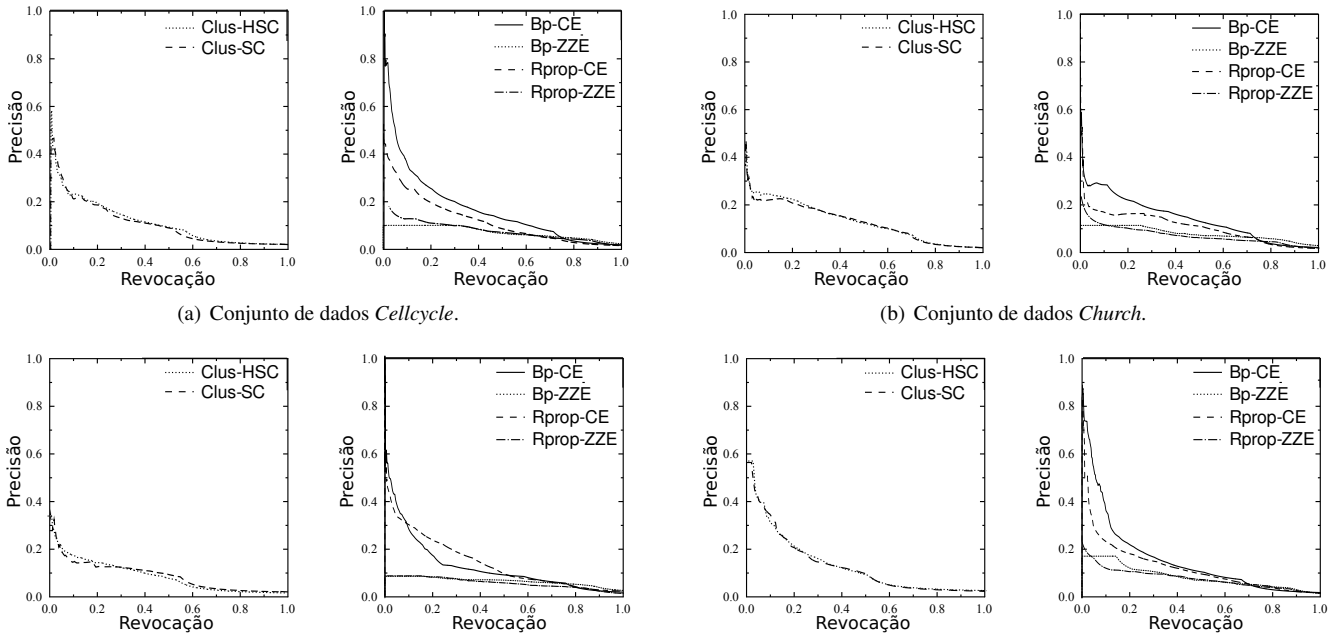
Nos gráficos da Figura 3 e nas $AU(\overline{PRC})$ da Tabela 2, quatro variações do método HMC-LMLP são utilizadas: *Back-propagation* e *Resilient back-propagation* utilizando a medida de erro convencional (Bp-CE e Rprop-CE), e *Back-propagation* e *Resilient back-propagation* utilizando a medida de erro multirrótulo proposta por Zhang e Zhou [15] (Bp-ZZE and Rprop-ZZE).

De acordo com a Tabela 2, os melhores resultados na maioria dos conjuntos de dados foram obtidos pela variação Bp-CE, que obteve a melhor $AU(\overline{PRC})$ três vezes, seguida por Rprop-CE, que obteve a segunda melhor $AU(\overline{PRC})$ três vezes, e pelos métodos Clus-HSc e Clus-SC. De acordo com o teste de Nemenyi, entretanto, resultados estatisticamente significantes foram observados apenas se comparadas as variações Bp-CE com as variações Bp-ZZE e Rprop-ZZE, e também se comparadas as variações Rprop-CE e Rprop-ZZE (parte inferior da Tabela 2).

De acordo com a Tabela 2, os desempenhos do método HMC-LMLP, especialmente quando utilizando a medida de erro convencional, foram competitivos se comparados com os resultados obtidos pelos métodos baseados em PCTs (Clus-HSC e Clus-SC). Esse resultado pode ser considerado motivador, dado que MLPs convencionais treinadas com o algoritmo *Back-propagation* foram utilizadas, e não foi feita nenhuma tentativa de otimizar os valores dos parâmetros das MLPs. De acordo com os resultados, desempenhos competitivos puderam ser obtidos com poucos ciclos de treinamento e relativamente poucos neurônios nas camadas escondidas (50% do número de entradas correspondentes).

O método HMC-LMLP obteve os piores resultados quando utilizado com a medida de erro multirrótulo. Isso pode ter acontecido porque, com essa medida, o número de classes preditas foi muito maior do que quando utilizando a medida de erro convencional. Isso pode ser confirmado pela análise do comportamento das curvas PR obtidas pelas variações Bp-ZZE e Rprop-ZZE (Figure 3). Nessas curvas, os valores de precisão permanecem sempre entre os valores 0.0 e 0.2 conforme os valores de revocação são variados. Nas curvas obtidas pelos outros métodos, os valores de precisão tendem a aumentar conforme os valores de revocação diminuem. Geralmente, baixos valores de precisão são indicativo de predições em níveis mais profundos (mais predições), enquanto altos valores de precisão são indicativo de predições em níveis mais próximos à raiz (menos predições). Os piores resultados obtidos com a utilização da medida de erro multirrótulo não eram esperados, especialmente porque bons resultados foram obtidos quando a medida foi inicialmente utilizada em problemas multirrótulo não hierárquicos [15]. Nos conjuntos de dados utilizados neste trabalho, entretanto, seu uso parece não ser muito adequado, talvez devido à natureza mais difícil do problema de classificação considerado, que possui centenas de classes.

Também é interessante observar nos resultados, que diferentemente das outras variações do método HMC-LMLP, o algoritmo Rprop utilizado com a medida de erro multirrótulo obteve seu melhor desempenho apenas após 980/1000 ciclos de treinamento. Originalmente, a medida de erro multirrótulo foi utilizada com o algoritmo *Back-propagation* em um modo de treinamento on-line. O algoritmo Rprop, entretanto, utiliza um modo de treinamento *batch*, que pode ter influenciado a maneira como a medida de erro captura as características do aprendizado multirrótulo. Por outro lado, o fato das $AU(\overline{PRC})$ obtidas pela variação Rprop-ZZE continuarem aumentando conforme o processo de treinamento continua pode indicar que essa variação é mais robusta a mínimos locais, e experimentos utilizando mais ciclos de treinamento podem levar melhores resultados.



(a) Conjunto de dados *Cellcycle*.

(b) Conjunto de dados *Church*.

(c) Conjunto de dados *Derisi*.

(d) Conjunto de dados *Eisen*.

Figura 3: Exemplos de curvas PR obtidas pelos métodos.

Tabela 2: $AU(\overline{PRC})$ obtidas nos quatro conjuntos de dados. ($\mu \pm \sigma$).

	Bp-CE	Rprop-CE	Bp-ZZE	Rprop-ZZE	Clus-HSC	Clus-SC
Cellcycle	0.14 ± 0.009 (20)	0.13 ± 0.012 (30)	0.08 ± 0.005 (10)	0.07 ± 0.008 (990)	0.11	0.11
Church	0.14 ± 0.002 (10)	0.13 ± 0.010 (40)	0.07 ± 0.008 (10)	0.07 ± 0.004 (1000)	0.13	0.13
Derisi	0.14 ± 0.010 (30)	0.14 ± 0.005 (30)	0.08 ± 0.008 (10)	0.07 ± 0.004 (980)	0.09	0.09
Eisen	0.17 ± 0.007 (60)	0.15 ± 0.014 (70)	0.09 ± 0.006 (10)	0.09 ± 0.004 (1000)	0.13	0.13
Nº Rank 1	3	1	0	0	0	0
Nº Rank 2	1	3	0	0	1	1
Nº Rank 3	0	0	0	0	3	3

Analisando as predições obtidas pelo método HMC-LMLP, pode-se observar que o método tem uma tendência a fazer mais predições nos primeiros níveis hierárquicos. Essa é uma característica dos métodos baseados na abordagem local, pois esses utilizam uma estratégia *top-down* que primeiro classifica um exemplo em classes localizadas no primeiro nível, e depois tenta predizer suas subclasses. Adicionalmente, conforme a hierarquia torna-se mais profunda, os conjuntos de dados tornam-se mais esparsos, tendo poucos exemplos positivos, o que aumenta a dificuldade da classificação. Classes localizadas em níveis mais profundos, no entanto, podem ser preditas com a utilização de valores de limiares adequados nas camadas de saída da rede. Geralmente, a utilização de valores baixos aumenta a revocação, resultando em predições nos níveis mais profundos, enquanto a

utilização de altos valores de limiares aumenta a precisão, resultando em predições nos primeiros níveis hierárquicos.

Uma desvantagem do método HMC-LMLP em comparação com os métodos Clus-HSC e Clus-SC é que, diferentemente desses métodos, as redes neurais não produzem regras de classificação. Entretanto, a investigação de MLPs convencionais aplicadas a problemas hierárquicos multirrótulo parece ser um campo muito interessante e promissor, pois as redes neurais podem ser naturalmente consideradas classificadores multirrótulo, já que podem predizer várias classes simultaneamente. Além disso, redes neurais são considerados classificadores robustos, capazes de encontrar soluções aproximadas para problemas muito complexos, o que o caso de problemas de classificação hierárquica multirrótulo.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou um novo método local para solucionar problemas de classificação hierárquica multirrótulo que utiliza *Multi-Layer Perceptrons* como classificadores base. O método proposto, denominado HMC-LMLP, treina uma rede neural MLP diferente para cada nível hierárquico. As saídas da rede responsável pelas predições em um determinado nível são utilizadas como entrada para a rede associada ao próximo nível, e assim sucessivamente. Dois algoritmos foram utilizados para treinar as MLPs base: Back-propagation [13] e Resilient back-propagation [14]. Além disso, uma nova medida de erro proposta especificamente para problemas multirrótulo [15] foi investigada. Os resultados dos experimentos sugerem que o método HMC-LMLP obtém desempenho preditivo competitivo quando comparado a algoritmos considerados estado-da-arte para problemas de classificação hierárquica multirrótulo [5]. Tais resultados são encorajadores, levando em conta que foram empregadas MLPs convencionais, sem quaisquer modificações para tratar problemas multirrótulo, e que não foram feitas tentativas de otimização dos parâmetros de execução das redes. Por outro lado, os métodos baseados em PCT têm sido investigados e otimizados por mais de uma década [11, 27, 30]. Como trabalhos futuros, propõe-se investigar a utilização de outras abordagens de redes neurais, como as redes *Radial Basis Function* (RBF) [34], como classificadores base do método proposto. Ainda, pretende-se testar o método HMC-LMLP em outros domínios de aplicação como a categorização de textos [35, 36]. Hierarquias estruturadas como DAGs também serão objeto de estudo, exigindo modificações durante a avaliação dos resultados obtidos pelo método.

AGRADECIMENTOS

Gostaríamos de agradecer à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). Também gostaríamos de agradecer à Dra. Celine Vens pelo auxílio com os métodos baseados em PCT.

Referências

[1] A. Freitas e A. C. Carvalho. "A Tutorial on Hierarchical Classification with Applications in Bioinformatics". In *Research and Trends in Data Mining Technologies and Applications*, chapter VII, pp. 175–208. Idea Group, 2007.

[2] C. Silla e A. Freitas. "A survey of hierarchical classification across different application domains". *Data Mining and Knowledge Discovery*, vol. 22, pp. 31–72, 2010.

[3] G. Tsoumakas, I. Katakis e I. P. Vlahavas. "Mining Multi-label Data". In *Data Mining and Knowledge Discovery Handbook*, pp. 667–685. Springer, second edition, 2010.

[4] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1999.

[5] C. Vens, J. Struyf, L. Schietgat, S. Džeroski e H. Blockeel. "Decision trees for hierarchical multi-label classification". *Machine Learning*, vol. 73, pp. 185–214, 2008.

[6] R. Alves, M. Delgado e A. Freitas. "Knowledge discovery with Artificial Immune Systems for hierarchical multi-label classification of protein functions". In *International Conference on Fuzzy Systems*, pp. 2097–2104, 2010.

[7] F. Otero, A. Freitas e C. Johnson. "A hierarchical multi-label classification ant colony algorithm for protein function prediction". *Memetic Computing*, vol. 2, pp. 165–181, 2010.

[8] A. C. P. L. F. Carvalho e A. A. Freitas. "A Tutorial on Multi-label Classification Techniques". In *Foundations of Computational Intelligence*, volume 205, pp. 177–195. Springer, 2009.

[9] A. Clare e R. D. King. "Predicting gene function in Saccharomyces cerevisiae". *Bioinformatics*, vol. 19, pp. 42–49, 2003.

[10] J. Struyf, H. Blockeel e A. Clare. "Hierarchical multi-classification with predictive clustering trees in functional genomics". In *Workshop on Computational Methods in Bioinformatics*, volume 3808 of *LNAI*, pp. 272–283. Springer, 2005.

[11] H. Blockeel, L. Schietgat, J. Struyf, S. Dzeroski e A. Clare. "Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics." In *Knowledge Discovery in Databases*, pp. 18–29, 2006.

[12] R. Cerri e A. C. P. L. F. Carvalho. "Hierarchical Multilabel Classification Using Top-Down Label Combination and Artificial Neural Networks". In *Brazilian Symposium on Artificial Neural Networks*, pp. 253–258, 2010.

[13] D. E. Rumelhart e J. L. McClelland. *Parallel distributed processing: explorations in the microstructure of cognition*, volume 1. MIT Press, Cambridge, MA, 1986.

[14] M. Riedmiller e H. Braun. "A Direct adaptive method for faster backpropagation learning: The RPROP algorithm". In *International Conference on Neural Networks*, pp. 586–591, 1993.

[15] M.-L. Zhang e Z.-H. Zhou. "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization". *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 1338–1351, 2006.

[16] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[17] Z. Barutcuoglu, R. E. Schapire e O. G. Troyanskaya. "Hierarchical multi-label prediction of gene function". *Bioinformatics*, vol. 22, pp. 830–836, 2006.

[18] M. Ashburner *et al.*. "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." *Nature Genetics*, vol. 25, pp. 25–29, 2000.

[19] N. Friedman, D. Geiger e M. Goldszmidt. "Bayesian Network Classifiers". *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[20] Y. Guan, C. Myers, D. Hess, Z. Barutcuoglu, A. Caudy e O. Troyanskaya. "Predicting gene function in a hierarchical context with an ensemble of classifiers". *Genome Biology*, vol. 9, pp. S3, 2008.

[21] P. Langley, W. Iba, and e K. Thompson. "An analysis of Bayesian classifiers". In *National conference on Artificial intelligence*, pp. 223–228, 1992.

[22] G. Valentini. "True Path Rule Hierarchical Ensembles". In *International Workshop on Multiple Classifier Systems*, pp. 232–241, 2009.

[23] H. W. Mewes *et al.*. "MIPS: a database for genomes and protein sequences." *Nucleic Acids Research*, vol. 30, pp. 31–34, 2002.

[24] M. R. Boutell, J. Luo, X. Shen e C. M. Brown. "Learning multi-label scene classification". *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[25] G. Tsoumakas e I. Vlahavas. "Random k-Labelsets: An Ensemble Method for Multilabel Classification". In *European Conference on Machine Learning*, pp. 406–417, Warsaw, Poland, 2007.

[26] H. Blockeel, M. Bruynooghe, S. Dzeroski, J. Ramon e J. Struyf. "Hierarchical multi-classification". In *Workshop on Multi-Relational Data Mining*, pp. 21–35, 2002.

[27] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev e S. Dzeroski. "Predicting gene function using hierarchical multi-label decision tree ensembles". *BMC Bioinformatics*, vol. 11, pp. 2, 2010.

[28] L. Breiman. "Bagging predictors". *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[29] F. Otero, A. Freitas e C. Johnson. "A hierarchical classification ant colony algorithm for predicting gene ontology terms". In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, volume LNCS, pp. 68–79. Springer, 2009.

[30] H. Blockeel, L. De Raedt e J. Ramon. "Top-down induction of clustering trees". In *International Conference on Machine Learning*, pp. 55–63, 1998.

[31] J. Davis e M. Goadrich. "The relationship between Precision-Recall and ROC curves". In *International Conference on Machine Learning*, pp. 233–240, 2006.

[32] S. Kiritchenko, S. Matwin e A. F. Famili. "Hierarchical Text Categorization as a Tool of Associating Genes with Gene Ontology Codes". In *European Workshop on Data Mining and Text Mining in Bioinformatics*, pp. 30–34, 2004.

[33] J. Demšar. "Statistical Comparisons of Classifiers over Multiple Data Sets". *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[34] M. J. D. Powell. *Radial basis functions for multivariable interpolation: a review*, pp. 143–167. Clarendon Press, New York, NY, USA, 1987.

[35] S. Kiritchenko, S. Matwin e A. Famili. "Functional annotation of genes using hierarchical text categorization". In *Proc. of the ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, 2005.

[36] A. Esuli, T. Fagni e F. Sebastiani. "Boosting multi-label hierarchical text categorization". *Inf. Retr.*, vol. 11, no. 4, pp. 287–313, 2008.