Previsão da quantidade de classes em Classificação Hierárquica Multirrótulo

Thissiany Beatriz Almeida
Universidade Tecnológica Federal do Paraná
Ponta Grossa – PR - Brasil
thissianyalmeida@alunos.utfpr.edu.br

Helyane Bronoski Borges
Universidade Tecnológica Federal do Paraná
Ponta Grossa – PR - Brasil
helyane@utfpr.edu.br

RESUMO
Muitos dos problemas de classificação descritos na literatura de Aprendizagem de Máquina dizem respeito à classificação de dados em que cada exemplo é associado a uma classe pertencente a um conjunto finito de classes, todas em um mesmo nível. No entanto, vários problemas de classificação, são de natureza hierárquica, em que classes podem ser subclasses ou superclasses de outras classes. Em muitos problemas hierárquicos, um ou mais exemplos podem ser associados a mais de uma classe simultaneamente. Esses problemas são conhecidos como problemas de classificação hierárquica multirrótulo. Nesse trabalho, foi utilizada a técnica ML-kNN para a predição de problemas multirrótulos, visando determinar o número de classes que podem ser atribuídas a um exemplo. Através dos experimentos e análise estatística pode-se mostrar que as adaptações realizadas na técnica ML-kNN trouxeram contribuições significativas com relação as medidas de precisão e revocação.

Palavras-chave
Classificação Hierárquica Multirrótulo, ML-kNN, Aprendizagem de Máquina.

ABSTRACT
Many Machine's Learning classification problems are described in the literature associating the classification of data with a class belonging to a finite set of classes, all at a same level. However, many classification problems are by hierarchical nature, in which classes may be subclasses or superclasses of other classes. With many hierarchical problems, one or more examples may be associated with more than one class simultaneously. Those problems are known as multi-label hierarchical classification problems. In this paper, the ML-KNN techniques used to address the prediction of multi-label problems, aiming to determine the number of classes that may be assigned to an example. Through the experiments and statistical analysis can be shown that the adaptations accomplished in the technique ML-kNN brought significant contributions to relationship the measures of precision and revocation.

Keywords
Hierarchical Multi-label Classification, ML-kNN, Learning Machine.

WPCCG'16, Setembro28, 2016, Ponta Grossa, Paraná, Brasil.

1. INTRODUÇÃO
O processo de classificação de dados na Aprendizagem de Máquina (AM) tem como objetivo atribuir uma classe para um novo exemplo a partir de suas características (atributos). Os problemas de classificação podem ser divididos em dois grandes grupos: Classificação Plana e Classificação Hierárquica, sendo diferenciados pelo relacionamento de dependência entre as classes [19]. A tarefa de classificação em AM pode também ser categorizada conforme a quantidade de classes a serem estimadas para um determinado exemplo. Sendo assim, esta categorização pode ser aplicada em problemas tradicionais (unirrótulo) ou problemas multirrótulos.

A motivação inicial para as pesquisas na área de classificação multirrótulo surgiu com a dificuldade causada por ambiguidades em problemas de categorização de textos [17].

A classificação hierárquica multirrótulo é considerada uma área de pesquisa relativamente nova[3][7][10], o que proporciona o interesse de pesquisadores de diferentes áreas. Esse tipo de classificação pode ser utilizado para categorização de textos [1][2], predição de proteínas em dados de bioinformática [3][4], classificação de gêneros musicais e imagens [5], entre outros.

Tem-se nesse trabalho como objeto de estudo os Problemas de Classificação Hierárquica Multirrótulo, onde as classes possuem relação hierárquica umas com as outras, sendo este relacionamento representado em formato de grafos acíclicos direcionados (DAG, do inglês *Directed Acyclic Graph*). Leva-se em consideração a organização hierárquica com o objetivo de aumentar a capacidade preditiva.

Neste trabalho é utilizado o algoritmo ML-kNN [21] para a determinação do número de classes a ser atribuído a um exemplo de teste. Os experimentos foram realizados utilizando dez bases de dados da área genômica funcional, *Gene Ontology*, sendo estas estruturas em DAG. Para a avaliação foram utilizadas variações no algoritmo ML-kNN para os valores de *k* e *threshold*, onde *k* recebe os valores 3, 5 e 7, enquanto o *threshold* varia entre 0.5, 0.7 e 0.8.

2. CONCEITOS FUNDAMENTAIS
2.1 Classificação Hierárquica Multirrótulo
Problemas de classificação hierárquica têm por objetivo a classificação de cada novo dado de entrada em um dos nós folhas fornecendo um conhecimento mais específico e útil [6]. Pode ocorrer, no entanto, do classificador não apresentar uma confiabilidade desejada na classificação em uma das classes do nível mais profundo, sendo mais seguro realizar a classificação nos níveis mais elevados.

A classificação hierárquica multirrótulo tem surgido como uma nova categoria de problemas de classificação, com características tanto dos problemas de classificação multirrótulo, quanto de problemas de classificação hierárquica. Problemas pertencentes a esta nova categoria são denominados de problemas de classificação

hierárquica multirrótulo (HMC, do inglês *Hierarchical Multilabel Classification*).

Em um problema de classificação hierárquica multirrótulo, um exemplo pode pertencer a múltiplas classes ao mesmo tempo e essas classes são organizadas de maneira hierárquica. A hierarquia pode ser representada em formato de árvore ou de um Grafo Acíclico Direcionado (DAG). Dessa forma, um exemplo pertencente a uma classe, automaticamente pertence a todas as suas superclasses [7].

A principal diferença entre a estrutura em árvore e a estrutura DAG é que, na estrutura em árvore, cada nó, exceto o nó-raiz tem somente um nó-pai, enquanto que no DAG cada nó, exceto o nó-raiz, pode ter um ou mais nós – pai [7].

Vários métodos podem ser utilizados no tratamento de tarefas de classificação hierárquica multirrótulo. Na literatura, há vários trabalhos propondo e analisando abordagens e métodos para tratamentos de problemas hierárquicos multirrótulo (HMC) [3][7][10][19], contudo, não há um consenso sobre qual algoritmo utilizar para o tratamento de problemas hierárquicos multirrótulo.

Pode-se dizer também que problemas de classificação hierárquica multirrótulo são mais complexos que os demais problemas de classificação, uma vez que as classes envolvidas no problema, além de estarem estruturados em uma hierarquia, os exemplos podem pertencer a mais de uma classe ao mesmo tempo [7].

2.2 Precisão Hierárquica e Revocação Hierárquica

No trabalho de Kiritchenko et al. (2004), foram propostas duas medidas de avaliação baseadas nas medidas convencionais de precisão e revocação, levando em consideração os relacionamentos hierárquicos entre as classes. Essas medidas foram chamadas de precisão e revocação hierárquicas e levam em consideração classificações nos nós internos e nós-folha.

Cada exemplo pertence não apenas à sua classe, mas também a todos os ancestrais dessa classe na estrutura hierárquica. Dessa maneira, dado um exemplo qualquer (x_i, Y_i), com x pertencente ao conjunto X de exemplos, Y_i o conjunto de classes preditas para o exemplo x_i, e Y_i' o conjunto de classes verdadeiras do exemplo x_i, os conjuntos Y_i e Y_i' podem ser entendidos para conterem suas correspondentes classes ancestrais da seguinte maneira: $\hat{Y} = \bigcup_{y_i \in Y_i} Ancestrais(y_k)$ e $\hat{Y} = \bigcup_{y_i \in Y_i'} Ancestrais(y_i)$.

A precisão e revocação hierárquica (Prec e Rev) são calculadas utilizando as equações abaixo, respectivamente.

$$Prec = \frac{\sum_i |\hat{Y}_i \cap \hat{Y}_i'|}{\sum_i |\hat{Y}_i|}$$

$$Rev = \frac{\sum_i |\hat{Y}_i \cap \hat{Y}_i'|}{\sum_i |\hat{Y}_i'|}$$

Essas medidas contam o número de classes preditas corretamente, juntamente com o número de classes ancestrais dessas classes preditas corretamente, assumindo que exemplos também pertencem aos ancestrais de suas classes corretas [22].

3. METODOLOGIA

3.1 Aplicação das técnicas

Para a realização dos experimentos foram escolhidas 10 bases de dados de funções de proteínas, sendo que estas já estavam normalizadas conforme o trabalho de Borges [19]. Com relação ao algoritmo escolheu-se o classificador ML-kNN [21] que é um dos algoritmos mais tradicionais utilizados em problemas de classificação multirrótulos. Este algoritmo é disponibilizado na ferramenta Mulan [20], sendo que esta trabalha em conjunto com as classes Java do Weka [18], um ambiente conhecido e utilizado pela comunidade de aprendizado de máquina e mineração de dados [18].

O ML-kNN é uma adaptação do algoritmo kNN [12] para o problema multirrótulo. Esse algoritmo determina o conjunto de rótulos do exemplo a ser classificado, baseado na probabilidade máxima a posteriori calculada a partir da frequência de cada rótulo entre os k vizinhos mais próximos, comparado com a frequência em todos os exemplos por meio do cálculo de distância.

O *framework* Mulan utiliza dois formatos de arquivo como entrada, sendo estes o ARFF (*Attribute-Relation File Format*) e o XML (*eXtensible Markup Language*). No arquivo ARFF é possível definir o tipo de dados que estão sendo carregados, e então fornecer seus próprios dados. No arquivo, foi definido cada coluna e o que cada coluna contém, fornecemos cada linha de dados em um formato delimitado por vírgulas. No Mulan cada rótulo vira um atributo classe do tipo {0,1}, onde 1 representa que aquele exemplo é pertencente a classe e 0 a ausência daquela classe. O arquivo XML é responsável por representar a hierarquia/dependência entre os rótulos/classes da base a ser analisada. Este arquivo se faz necessário durante as etapas de classificação e avaliação.

Ao ser analisado o arquivo XML exigido pelo framework Mulan foi verificado que não era possível representar problemas hierárquicos multirrótulo em formato de Grafo Acíclico Direcionado, somente exemplos que tenham a hierarquia estruturada em formato do tipo árvore, que são estruturas mais simples, onde um nó filho tem apenas um nó pai.

As bases escolhidas para este trabalho possuem sua hierarquia estruturada em formato do tipo DAG, sendo esta estrutura mais complexa quando comparada ao do tipo árvore, pois um nó filho pode ter múltiplos pais. Devido a esse fato foi necessário criar uma nova forma de representar a hierarquia dos rótulos substituindo assim, a necessidade de fornecer o arquivo do tipo XML como entrada, e permitindo a utilização de qualquer estrutura hierárquica do tipo DAG.

Tabela 1. Características das bases de dados GO

Bases	Quant Amostras	Quant. Atributos	Quant. Classes	Quant. Max. Níveis
Cellcycle	3751	77	4125	13
Church	3749	27	4125	13
Derisi	3719	63	4119	13
Eisen	2418	79	3573	13
Expr	3773	551	4131	13
Gasch1	3758	173	4125	13
Gasch2	3773	52	4131	13
Pheno	1586	69	3127	13
Seq	3900	478	4133	13
Spo	3697	80	4119	13

O programa desenvolvido lê o arquivo que contém a definição da estrutura hierárquica entre as classes armazenando-a em um grafo. Após isso são criados dois conjuntos de dados, um para a base de treinamento e o outro para a base de teste, e por fim instanciado o classificador passando como parâmetro o valor de k-vizinhos. Lembrando que o valor de k-vizinhos pode influenciar durante o processo de classificação. A construção do modelo de classificação dá-se ao treinar o classificador com a base de dados de treinamento. Para cada instância da base de teste é calculada a distância euclidiana com todas as instâncias da base de treinamento o que

resulta em um vetor. Este vetor varia de 1 até o número de classes e para cada classe é feita uma comparação com o valor de corte *(threshold)*. Caso o valor de semelhança seja igual ou maior ao valor de corte então atribuímos a classe para aquela instância. Após a classificação das classes geradas pela semelhança com os k-vizinhos, o algoritmo faz uma varredura nas classes hierarquicamente ancestrais das preditas e também seta as mesmas como classes pertencentes a instância analisada.

Foi adotada a técnica de Spyromitros [11] antes utilizada apenas para o algoritmo BRkNN. Onde caso nenhuma das classes tenha sido predita para um exemplo, será predita a classe que apresentar o maior valor de semelhança com o exemplo, com o objetivo de viabilizar o cálculo da medida de precisão.

3.2 Avaliação e análise dos resultados

Considerando as peculiaridades inerentes aos problemas de classificação hierárquica multirrótulo, medidas específicas para estes tipos de problemas devem ser utilizadas para avaliar os modelos de classificação gerados para solucioná-los. Tais medidas requerem algumas considerações adicionais, além dos aspectos normalmente considerados na avaliação de modelos convencionais de classificação.

Após as adaptações do algoritmo ML-kNN foram realizados alguns experimentos para comparação do desempenho sendo utilizadas como base as medidas de precisão hierárquica e revocação hierárquica [22], sendo que essas medidas levam em consideração classificações nos nós internos e nós-folha. Para isso, foram escolhidos valores para a variável de corte (*threshold*) de 0.5, 0.7 e 0.8. Considerando o valor da variável de corte constante e variando-se o valor do k-vizinhos obteve-se os resultados como são mostrados nas Tabelas 1, 2 e 3, conforme os valores de *threshold* de 0.5, 0.7 e 0.8, respectivamente.

Tabela 2. Resultados para os experimentos com *threshold* =0.5

***Threshold* = 0.5**						
Bases	**K=3**		**K=5**		**K=7**	
	Prec.	**Rev.**	**Prec.**	**Rev.**	**Prec.**	**Rev.**
Cellcycle	0,736	0,297	0,759	0,289	0,747	0,297
Church	0,769	0,248	0,772	0,247	0,735	0,263
Derisi	0,757	0,257	0,753	0,262	0,761	0,261
Eisen	0,743	0,298	0,741	0,301	0,744	0,296
Expr	0,771	0,293	0,768	0,299	0,756	0,305
Gasch1	0,763	0,291	0,761	0,297	0,761	0,299
Gasch2	0,747	0,287	0,762	0,278	0,771	0,274
Pheno	0,765	0,244	0,733	0,255	0,743	0,248
Seq	0,771	0,286	0,776	0,282	0,780	0,283
Spo	0,741	0,280	0,736	0,285	0,745	0,280

Tabela 3. Resultados para os experimentos com *threshold* =0.7

***Threshold* = 0.7**						
Bases	**K = 3**		**K = 5**		**K = 7**	
	Prec.	**Rev.**	**Prec.**	**Rev.**	**Prec.**	**Rev.**
Cellcycle	0,899	0,201	0,899	0,209	0,910	0,201
Church	0,897	0,188	0,895	0,189	0,897	0,188
Derisi	0,895	0,191	0,892	0,192	0,893	0,194
Eisen	0,855	0,218	0,871	0,214	0,882	0,208
Expr	0,884	0,229	0,887	0,228	0,897	0,221
Gasch1	0,881	0,219	0,881	0,229	0,893	0,219
Gasch2	0,899	0,199	0,895	0,207	0,900	0,204
Pheno	0,894	0,185	0,892	0,186	0,887	0,187
Seq	0,912	0,201	0,906	0,206	0,900	0,212
Spo	0,902	0,194	0,890	0,204	0,889	0,200

Tabela 4. Resultados para os experimentos com *threshold* =0.8

***Threshold* = 0.8**						
Bases	**K = 3**		**K = 5**		**K = 7**	
	Prec.	**Rev.**	**Prec.**	**Rev.**	**Prec.**	**Rev.**
Cellcycle	0,918	0,187	0,925	0,182	0,930	0,183
Church	0,946	0,154	0,947	0,154	0,946	0,154
Derisi	0,943	0,156	0,943	0,155	0,936	0,163
Eisen	0,923	0,166	0,906	0,186	0,910	0,186
Expr	0,924	0,193	0,928	0,195	0,933	0,196
Gasch1	0,920	0,187	0,921	0,195	0,926	0,192
Gasch2	0,919	0,183	0,927	0,181	0,933	0,179
Pheno	0,926	0,160	0,937	0,151	0,930	0,157
Seq	0,928	0,181	0,918	0,194	0,910	0,193
Spo	0,940	0,158	0,926	0,174	0,928	0,171

Para comparação das 9 variações de experimentos realizados adotou-se a utilização do teste estatístico de Wilcoxon. Para realizar esse teste estatístico foram escolhidos os valores obtidos na medida de precisão.

No teste estatístico foi definida que a hipótese nula assume que a diferença de desempenho entre algoritmos não é significativa. Com nível de confiança $\alpha = 0.05$ a hipótese nula não pode ser rejeitada se $-1.96 \leq z \leq 1.96$.

A seguir, os resultados obtidos pelo Teste de Wilcoxon levando em consideração a variação do valor de *k*. Nas Tabelas 5, 6 e 7 são encontrados os valores tabulados do teste para *k* assumindo os valores 3, 5 e 7.

Tabela 5. Resultados do teste com Threshold = 0.5

***Threshold* = 0.5**	
K = 3 com K = 5	Z-Score = 0.153
K = 3 com K = 7	Z-Score = 0.051
K = 5 com K = 7	Z-Score = 0.459

Tabela 6. Resultados do teste com Threshold = 0.7

***Threshold* = 0.7**	
K = 3 com K = 5	Z-Score = 0.948
K = 3 com K = 7	Z-Score = 0.714
K = 5 com K = 7	Z-Score = 1.631

Tabela 7. Resultados do teste com Threshold = 0.8

***Threshold* = 0.8**	
K = 3 com K = 5	Z-Score = 0.153
K = 3 com K = 7	Z-Score = 0.296
K = 5 com K = 7	Z-Score = 0.051

Através dos valores de *z* encontrados nas Tabelas 5, 6 e 7 pode-se concluir que nenhum valor permite que a hipótese nula seja rejeitada, ou seja, não há melhora no desempenho do algoritmo realizando a variação dos valores de *k*.

Nas tabelas 8, 9 e 10 são encontrados os valores tabulados do teste para *threshold* assumindo os valores 0.5. 0.7 e 0.8.

Tabela 8. Resultados do teste com K = 3

K = 3	
Threshold = 0.5 com 0.7	Z-Score = -2.8031
Threshold = 0.5 com 0.8	Z-Score = -2.8031
Threshold = 0.8 com 0.7	Z-Score =-2.8031

Tabela 9. Resultados do teste com K = 5

K = 5	
Threshold = 0.5 com 0.7	Z-Score = -2.8030
Threshold = 0.5 com 0.8	Z-Score = -2.8030
Threshold = 0.8 com 0.7	Z-Score = -2.8030

Tabela 10. Resultados do teste com K = 7

K = 7	
Threshold = 0.5 com 0.7	Z-Score = -2.8032
Threshold = 0.5 com 0.8	Z-Score = -2.8032
Threshold = 0.8 com 0.7	Z-Score = -2.8032

Ao contrário do resultado obtido na variação dos valores de *k*, é possível perceber que neste teste, os valores obtidos de *z*, são superiores ao limite de -1.96, o que significa que a hipótese nula pode ser rejeitada e que há melhoras no desempenho do algoritmo.

4. CONCLUSÃO

Neste trabalho foram apresentados experimentos com as bases de dados estruturadas em formato de Grafo Acíclico Direcionado adaptando-se o algoritmo de classificação hierárquica multirrótulo, o ML-kNN. Dentre essas mudanças, pode-se citar a modificação do arquivo XML utilizado como arquivo de entrada pelo framework Mulan, onde encontra-se implementado o algoritmo ML-kNN. Pode-se ressaltar também o fato da utilização da técnica antes utilizada apenas no algoritmo BR-kNN, se na fase de predição não for atribuída nenhuma classe a instância, atribuísse a essa instância então a classe que possuir o maior valor de confiança.

Com base nos testes estatísticos, o algoritmo ML-kNN tem um desempenho superior levando-se em consideração a medida de precisão quando assume valor de threshold igual a 0.8, esse fato pode ser explicado na comparação entre o valor da confiança dos rótulos com o threshold, pois quanto maior o valor do threshold a tendência é que seja predito um número cada vez menor de classes, entrando na condição onde é atribuída apenas a classe com a maior confiança dentre os rótulos daquela instância.

5. REFERÊNCIAS

[1] Dumais, S. and Chen, H. *Hierarchical classification of web content*. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Athens, Greece, pp. 256-263, 2000.

[2] Sun, A. and Lim, E.-P. *Hierarchical text classification and evaluation*. In Proceedings of the 2001 IEEE International Conference on Data Mining. IEEE Computer Society, pp. 521-528, 2001.

[3] Costa, E. P., Lorena, A. C, Carvalho, A.P.L.F, &Freitas,A. A. (2007). A review of Performance Evaluation Measures for Hierarchical Classifiers. In Proceedings of the AAAI07 – Workshop on Evaluation Methods for Machine Learning II, P.1-6.

[4] Holden, N. and Freitas, A. *A Hierarchical classification of protein function with ensembles of rules and particle swarm optimisation*. Soft Comput. vol. 13, pp. 259-272, 2008.

[5] Barutcuoglu, Z. and DeCoro, C. *Hierarchical shape classification using Bayesian aggregation. In Proceedings of the IEEE International Conference on Shape Modeling and Applications*. Matsushima, Japan, pp. 44-44, 2006.

[6] Carvalho, A. C. P. F.; Freitas, A. A. ***Tutorial on Hierarchical Classification with Applications in Bioinformatics***.v.1.São Paulo: Idea Group, 2007.

[7] Cerri, R., Carvalho, A. C. P. L. F., e Costa, E. P(2008). Classificação hierárquica de proteínas utilizando técnicas de aprendizado de máquina. In *II Workshop on Computational Intelligence*, páginas 1-6, Salvador.

[8] Guyon, I. and Elisseeff, *A. An introduction to feature extraction. In Feature Extraction, Foundations and Applications*. Springer, pp. 1-24, 2006.

[9] Yang, Y. and Pedersen, J. O. *A comparative study on feature selection in text categorization*. In Proceedings of the Fourteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., pp. 412-420, 1997.

[10] Santos, A. M., Canuto, A. M. P. *Investigating the influence of repart in ensemble systems designed by boosting*. In: *IJCNN. Hong Kong*: IEEE, 2008. P. 2907-2914.

[11] Spyromitros, E.; Tsoumakas, G.; Vlahavas, I.*An empirical study of lazymultilabel classification algorithms.In:Hellenic conference on Artificial Intelligence*,p. 401–406, Berlin, Alemanha, 2009.

[12] Aha, D. W., Kibler, D. e Albert, M. K. (1991). *Instance-based learning algorithms. Machine Learning*, 6(1): 37-66.

[13] Quinlan, J.R. C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning). 1. Ed. San Francisco, California: Morgan Kaufmann, 1993. Paperback.

[14] Schapire, R. E.; Singer, Y. Boostexter: A boosting-based system for text categorization. Machine Learning, v. 39, n. 2/3, p. 135-168, 2000.

[15] Hall, M.; Frank, E.; Holmes, G.;Pfahringer, B.; Reutemann,P.; Witten, I. H. 2009. The WEKA data mining software: an update. **SIGKDD Explor. News,** v. 11, p.10-18, 2009.

[16] Borges, Helyane Bronoski; Nievola, J. C. *Multi-Label Hierarchical Classification using a Competitive Neural Network for Protein Function Prediction.* In: 2012 International Joint Conference on Neural Networks (IJCNN 2012), 2012, Brisbane, Austrália. 2012 International Joint Conference on Neural Networks (IJCNN 2012). Piscataway, NJ: IEEE Press, 2012. v. 1. p. 1-8.

[17] Tsoumakas, G., Katakis, I., Vlahavas, I. (2010) "Mining Multi-label Data", Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach (Ed.), Springer, 2nd edition, 2010.

[18] Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern recognition 40(7), 2038–2048 (2007).

[19] Kiritchenko, S.; Matwin, S.; Famili, A. F. *Hierarchical text categorization as a tool of associating genes with gene ontology codes*. In: *Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics*, Pisa, Italia,2004.