A Hybrid Global-Local Approach for Hierarchical Classification

Julio Hernandez and **L. Enrique Sucar** and **Eduardo F. Morales**
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1, Tonantzinlta, Puebla, Mexico
{julio.hernandez.t, esucar, emorales}@inaoep.mx

Abstract

Hierarchical classification is a variant of multidimensional classification where the classes are arranged in a hierarchy and the objective is to predict a class, or set of classes, according to a taxonomy. Different alternatives have been proposed for hierarchical classification, including local and global approaches. Local approaches are prone to suffer the *inconsistency* problem, while the global approaches tend to produce more complex models. In this paper, we propose a hybrid global-local approach inspired on multidimensional classification. It starts by building a local multi-class classifier per each parent node in the hierarchy. In the classification phase all the local classifiers are applied *simultaneously* to each instance resulting in a most probable class for each classifier. A set of *consistent* classes are obtained, according to the hierarchy, based on three novel alternatives. The proposed method was tested on three different hierarchical classification data sets and was compared against state-of-the-art methods, resulting in significantly superior performance to the traditional top-down techniques; with competitive results against more complex top-down classifier selection methods.

Introduction

The traditional classification process consists on assigning a class c, from a finite set C of classes, to a single instance x, represented by a feature vector. A dataset D, for this kind of classification, is composed of n examples: $(x_1, c_1), ..., (x_n, c_n)$. The multidimensional classification[1] process, in contrast with this approach, assigns a subset of classes $J \subseteq C$ to a single instance x. A dataset D for a multidimensional problem is composed of n examples: $(x_1, J_1), ..., (x_n, J_n)$.

Hierarchical classification is a variant of the multidimensional task with the difference that classes are arranged in a hierarchy. This hierarchy can be either a tree or a Direct Acyclic Graph (DAG), where each node corresponds to a class. There are a many fields where hierarchical classification has gain popularity like musical genre classification (Silla-Jr. and Freitas 2009), web content tasks (Dumais and Chen 2000), bioinformatics (Valentini 2009), (Freitas and de Carvalho 2007), (Secker et al. 2010), and computer vision (Barutçuoglu and DeCoro 2006), among others.

Different alternatives have been proposed for hierarchical classification, including local or top-down (Holden and Freitas 2008; Silla-Jr. and Freitas 2009; 2011) and global or big-bang (Wang and Zhou 2001; Blockeel et al. 2006; Vens et al. 2008) approaches. The local or global hierarchical problems can be divided into single label (assign one label per instance) or multi-label (assign more than one label per instance) (Dumais and Chen 2000; Kiritchenko et al. 2006; Vens et al. 2008) problems. Local approaches consist of a series of local classifiers, which are usually applied in a top-down fashion; they suffer the *inconsistency* problem; that is, if a local classifier assigns a wrong class, the error is propagated down the hierarchy. The global approach considers a single classifier (usually for all the leaf classes in the hierarchy) resulting in a more complex model than the local approaches (Silla-Jr. and Freitas 2011).

In this work, we propose an alternative approach for hierarchical classification, which can be thought of as a hybrid (global-local) method, inspired in non-hierarchical multidimensional classification techniques. The main idea is to learn a series of local classifiers and then to combine their results to obtain a set of consistent classes; that is a set of paths from the root to a leaf in the hierarchy. In the training phase, the proposed method learns a multi-class classifier per each parent node in the hierarchy. In the classification phase, in contrast with traditional top-down approaches, all the local classifiers are applied *simultaneously* to each instance, so for each local classifier a probability for each class is estimated. Then, a set of *consistent* classes, according to the hierarchy, is obtained. We propose and compare three alternatives to build the consistent set: (i) order all the classes according to their probabilities and select the first consistent subset based on this order, (ii) multiply the probabilities of all possible trajectories from the root to a leaf node in the hierarchy, and select the trajectory with greatest value, and (iii) as (ii) but using addition of probabilities instead of multiplication.

We evaluated the proposed method with three hierarchical classification data sets of different domains: text, images, and genes; considering two different base classifiers: Naive Bayes and Random Forest. We compared the pro-

[1] Also known as multi-label classification when the classes are all binary.

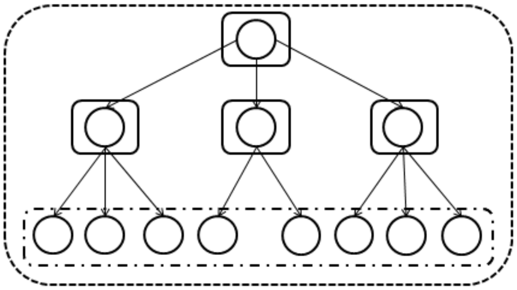

Figure 1: Main types of hierarchical classifiers: *Local classifier*: a multi-label classification algorithm is used per parent node. The circles represent the classes and the solid squares represent multi-label classifiers. *Global classifier*: a classification algorithm (dashed square) that learns a global classification model that takes into account the whole hierarchy. *Flat classifier*: a flat multi-label classification algorithm is used (dash-dot square) to only predict the leaf nodes.

posed approach in terms of standard and hierarchical precision against a traditional hierarchical local classifier, the top-down approach (Koller and Sahami 1997), using the same base classifiers. We also compared it against an improvement of the top-down method that does classifier selection for each node and, additionally, is one of the current top performing techniques in the literature (Secker et al. 2007; 2010). For all the data sets our method has significantly superior performance than the top-down approach with similar efficiency. With respect to the top-down classifier selection method, its performance is similar in two datasets and superior in one, but it is in all cases much more efficient.

Next we present a review of hierarchical classification including the most relevant related work. Then we describe in detail the proposed method and the experimental evaluation. We conclude with a summary and directions for future work.

Hierarchical Classification

According to (Freitas and de Carvalho 2007), hierarchical classification methods differ in three principal criteria. The first criterion is the type of hierarchical structure used, this one can be a tree or a DAG. The second criterion is related to how deep the classification in the hierarchy is performed. One way is to always classify a leaf node, also known as *mandatory leaf-node prediction*; another one is to consider stopping the classification process at any level of the hierarchy, also known as *non-mandatory leaf-node prediction*. The final criterion is related to how the hierarchical structure is explored: Local (also known as Top-Down), Global (also known as Big-Bang), or Flat, see Fig 1.

The most popular form to explore the hierarchical structure is the local or top-down approach. This approach can be performed in a binary or a multi-class way. In the training phase a tree of classifiers is built. If the classification process is binary, there is a binary classifier for each node (class), except for the root node. If the classification process is multi-class, then there is a multi-valued classifier per each parent node, except for the leaf nodes. In the classification phase, a test example is classified in a top-down fashion. The first classifier decides where the example belongs and passes the example to the respective classifier in the hierarchy, this procedure is repeated until the example reaches a leaf node. Typically, every node in the hierarchy uses the same kind of classification algorithm. An important limitation of this type of methods is that if the prediction is incorrect in certain node in the hierarchy the error is propagated to all its descendants; this is known as the *inconsistency* problem.

There has been an increasing interest in developing better hierarchical classifiers, next we review the most relevant recent work.

Related Work

Secker et al. (2007) propose an alternative strategy based on the premise that each local classifier should be adapted to the particular problem it solves. They developed a top-down *classifier selection* technique in which a different classifier is selected at each node in the hierarchy from a set of possible models, based on its performance in a validation set. From this work several extensions have been developed (Holden and Freitas 2008; Silla-Jr. and Freitas 2009; Secker et al. 2010) which incorporate also feature selection for each local classifier. In general, these methods improve the performance of the local approaches that use the same base classifier for all the hierarchy, however there is also a significant increase in the training time.

The following two related works combine the predictions of all the local classifiers as in the proposed method, however they do it in a different way.

In (Dumais and Chen 2000), the authors propose two methods that consider the output of all the local classifiers in the hierarchy. The first one is based on a Boolean rule and the second on a multiplicative rule; both use *Support Vector Machines* as base classifier. They consider a two-level hierarchy and a decision threshold. For the first option (Boolean rule), the predictions in the second level are considered only if the first level is above a threshold value. In the second option, they combine the probabilities from each branch in both levels of the hierarchy by multiplying them, and then select those that are above the predefined threshold. This work is restricted to two-level hierarchies and depends on the selection of thresholds values.

Valentini (2009) developed a hierarchical ensemble based on the *True Path Rule* from the *Gene Ontology*. This rule establishes the following: “if an example belongs to a class, it belongs to all its ancestors, and if it does not belong to a class, it does not belong to its offsprings”. According to this rule, the positive decision of a local classifier influences its ancestors (in a recursive way) and a negative decision turns to negative all it descendants. In this way a consensus probability is given by each local classifier using an ensemble that combines its local prediction with those of its ancestors and descendants in the hierarchy. This work is focused on gene classification and also suffers from the inconsistency problem in particular for negative predictions.

Thus, previous approaches for combining the outputs of several local classifiers are restricted to particular cases, and depend on predefined thresholds or suffer from inconsistencies; we propose a more general approach which does not require thresholds and reduces the inconsistency problem.

A Hybrid Hierarchical Classifier

We assume a tree-structured taxonomy, T, with t nodes, each node represents a class such that it is a subset of its parent node in the tree. There are c non-leaf nodes and l leaf nodes, such that $t = c + l$. Each non-leaf node c_i has ns_i sons, which represent the direct subclasses of class c_i. We assume that there are m attributes for each class, such that the same set of attributes is considered for all the classes.

The proposed method, HHC, includes two phases, training and classification.

Training: Given a data base composed of n data points: $(\mathbf{x}_1, j_1), \ldots, (\mathbf{x}_n, j_n)$ where $\mathbf{x}_i$ are the m attributes and j_i the class according to a taxonomy T:

1. Learn a multi-class classifier for each non-leaf node c_i to classify its ns_i sons.

For learning each local multi-class classifier we consider as examples all the instances that correspond to each of its ns_i sons and their descendants.

Classification: Given an instance **x**:

1. Classify **x** with all the c local classifiers.
2. Combine the results of all the classifiers to produce a set of classes, O, which corresponds to the most probable path in the hierarchy (from the root to a leaf node), following one of the three alternative strategies (described below).

We propose three different ways to combine the results from the local classifiers to make a global prediction: Descending Order of Probabilities (DOP), Multiplication of Probabilities (MP), and Sum of Probabilities (SP).

Descending Order of Probabilities. The classes predicted by all the local classifiers are ordered according to their probability in a descendant way. According to this order, it seeks the first consistent subset of classes (a path from the root to a leaf) and this set is returned as the global prediction, see Fig. 2. The main advantage of this method is that it selects the best way according to the probability of each node, but its main drawback is when the hierarchy is unbalanced since branches with few nodes have a higher probability to be chosen.

Multiplication of Probabilities. This method multiplies the probabilities of the most probable class for all the nodes of every path, from the root to a leaf, in the hierarchy:

$$R_j = \frac{\prod_{i=1}^{n} p_i}{n} \tag{1}$$

where n is the number of nodes of the j branch, p_i is the probabilities of each node in the branch, and R_j is the result for the branch.

This method implies an assumption of conditional independence; it could suffer from numerical problems in very long paths. The global prediction will be the set of classes of the path with the highest product, see Figure 3.

Sum of Probabilities. The probabilities of the most probable class for all the nodes of every path in the hierarchy are added:

$$R_j = \frac{\sum_{i=1}^{n} p_i}{n} \tag{2}$$

The global prediction will be the path with the highest sum.

This method is similar to the product one (if we take the logarithm of a product it is equivalent to the sum); but it is less prone to numerical problems and it is more efficient.

Experiments and Results

We evaluated the proposed method with three hierarchical databases and compared its performance against state of the art top-down classifiers. First we describe the data sets and methods, then we present the experiments and results, to conclude with an analysis.

Datasets

We consider three hierarchical datasets from different domains: *Reuters-21578* (Yang 1999), *FunCat* (Ruepp et al. 2004), and *IAPR-TC12* (Escalante et al. 2010).

Reuters-21578[2] is a popular database for text retrieval (Yang 1999). It has 135 categories and a taxonomy proposed by (Toutanova et al. 2001) divided in four main branches. *FunCat*[3] is a database in the domain of bioinformatics, in particular for protein function prediction (Ruepp et al. 2004). It includes 27 categories, in this work we consider the branch that corresponds to *Cellcycle*. *IAPR-TC12*[4] (Escalante et al. 2010) is a collection of segmented and annotated images with 20,000 images and 99,000 annotated regions. Annotations are based on an object taxonomy divided in 6 main categories. In this work we consider the category *Landscape-Nature*. The main properties of the three datasets are summarized in Table 1. Additionally, for each database, we only take into account the classes with more than 10 examples to guarantee a sufficient number of examples to train the classifier in each fold.

Methods

We compared the proposed HHC against local hierarchical classifiers with a multi-class classifier per parent node, in which the classification is performed in a top-down fashion. Three variants of this scheme were considered. Two use the same local classifier for each parent node in all the hierarchy; one uses as base classifier *Naive Bayes* and the other *Random Forest*. The third scheme uses the classifier selection method proposed in (Secker et al. 2007). For each local classifier it selects empirically the *best* algorithm from the following set: Naive Bayes, Bayes Net, SVM, AdaBoost, 3-KNN, PART, J48 and Random Forest (from Weka (Witten and Frank 2005)). This method has shown superior performance than the top-down approach using the same classifier for each node in the hierarchy.

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578.

[3] http://mips.helmholtz-muenchen.de/proj/funcatDB/

[4] http://ccc.inaoep.mx/∼tia/pmwiki.php?n=Main.Resources

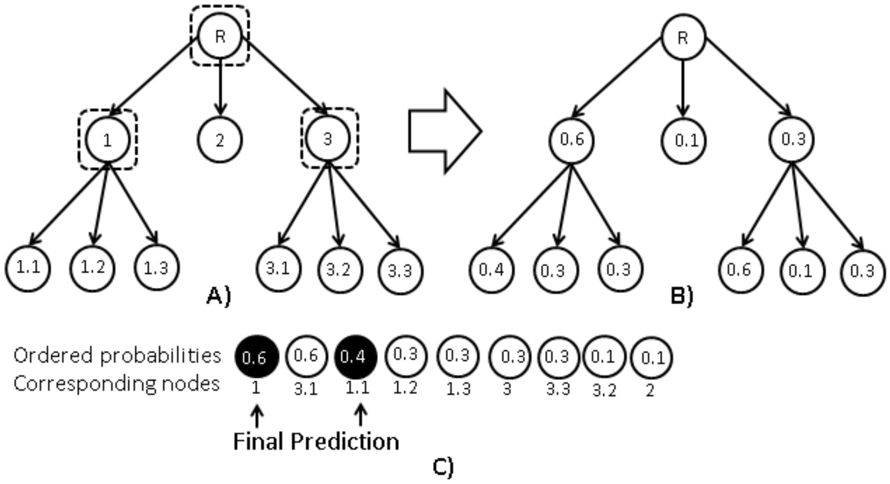

Figure 2: Descending order of probabilities (DOP). A) The class taxonomy, the circles represent the classes and the dashed squares represent multi-label classifiers. B) For each node (except the root), P_i corresponds the predicted probability of each class. C) The classes are sorted in descending order according to their probability, the first subset of consistent classes according to this order are selected (marked).

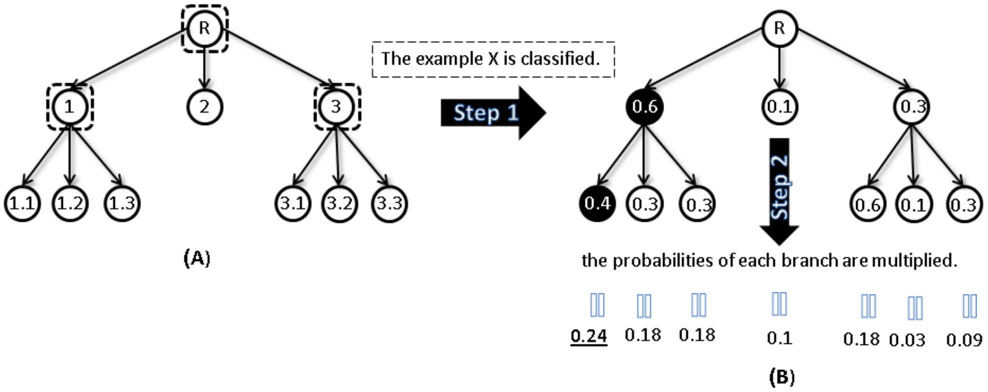

Figure 3: Multiplication of probabilities (MP). (A) The class taxonomy, as in Fig. 2. (B) Each node (except the root) depicts the predicted probability of its class. The probabilities in each trajectory in the tree –for instance $P_1 \times P_{1.1}$– are multiplied, the results are shown below; the trajectory with highest product is selected (underlined).

For HHC, we used the same method for each local classifier. We considered the same two alternatives as the top-down approach: *Naive Bayes* and *Random Forest*. We compared the three options for selecting the output subset of classes: *Descending Order of Probabilities*, *Multiplication of Probabilities*, and *Sum of Probabilities*.

Experiments

We evaluated the different classification schemes in terms of two precision measures: a *standard* precision and the *hierarchical* precision. The standard precision considers a classification correct only if it exactly predicts the class of the test sample.

The hierarchical precision considers that a classifier might be *partially correct*, for instance if it predicts the parent or sibling of the correct class of a sample. It is defined as:

$$hP = \frac{\sum_{i=1}^{i=n} |\hat{C}_i \cap C_i|}{|C_i|} \qquad (3)$$

Where $\hat{C}_i$ is the set of predicted classes for the test sample i and C_i is the actual set of classes for i; the class set includes the more specific class and all its ascendants in the hierarchy; the summation is over all test samples, n.

To perform the experiments we used stratified five-fold cross validation. Tables 2, 3, and 4 summarize the results for the three datasets. In these tables, *DOP* corresponds to *Descending Order of Probabilities*, *MP* to *Multiplication of Probabilities*, and *SP* to *Sum of Probabilities*, considering the two base classifiers (Naive Bayes and Random Forest). For comparison the standard top-down approach is contrasted for the same two base classifiers, as well as the top-down classifier selection method.

Table 1: Characteristics of the databases used in the experiments.

DataBase	Domain	# Classes	# Examples	# Levels	# Attributes	Type of Hierarchy
*FunCat**	Genetics	30	1433	3	77	Tree
*Reuters-21578***	Text	25	6274	2	16145	Tree
*IAPR-TC12****	Image	25	45347	2	23	Tree
* it is considered only the subset *Cellcycle* of the original hierarchy						
** it is considered the subset *R52*						
*** it is considered only the *Landscape* branch of the original hierarchy						

Each cell of the proposed method (DOP, MP, SP) has the symbol "*" if the precision reported in that cell is statistically significantly better than the precision of the corresponding top-down classifier (reported in the last column for the corresponding row). Likewise, each cell has the symbol "†" if the precision reported in that cell is statistically significantly better than the precision of the top-down classifier selection method (reported in the last column of the "classifier selection" row). Statistical significance was measured by the paired two-tailed Student's t-test, using a confidence level of 95%. For each data set, the classifier that obtained the best results in terms of standard and hierarchical precision is shown in bold.

Table 2: Experimental results for the FUNCAT database.

Classifier	DOP	MP	SP	Top-Down
Hierarchical Precision (in %)				
Naive Bayes	28.10	28.78	28.15	28.10
Random Forest	28.73 *	27.72	28.84 *	26.93
Classifier Selection	*N/A*	*N/A*	*N/A*	**31.11**
Standard Precision (in %)				
Naive Bayes	16.35 * †	16.67 †	17.14 †	16.35
Random Forest	15.87 *	17.94 * †	**18.22** * †	13.33
Classifier Selection	*N/A*	*N/A*	*N/A*	14.92

Table 3: Experimental results for the REUTERS database.

Classifier	DOP	MP	SP	Top-Down
Hierarchical Precision (in %)				
Naive Bayes	76.64	76.71	76.71	76.11
Random Forest	84.53	84.79	85.29 *	83.54
Classifier Selection	*N/A*	*N/A*	*N/A*	**89.27**
Standard Precision (in %)				
Naive Bayes	70.01	70.01	70.01	70.01
Random Forest	78.28	79.04 *	78.96	77.32
Classifier Selection	*N/A*	*N/A*	*N/A*	**85.40**

Table 4: Experimental results of the IAPR-TC12 database.

Classifier	DOP	MP	SP	Top-Down
Hierarchical Precision (in %)				
Naive Bayes	50.68 * †	50.84 * †	50.82 * †	37.71
Random Forest	**58.78** * †	55.35 * †	57.90 * †	44.65
Classifier Selection	*N/A*	*N/A*	*N/A*	45.19
Standard Precision (in %)				
Naive Bayes	39.28 *	39.52 *	39.76 *	41.72
Random Forest	47.17 *	47.45 *	46.73 *	47.98
Classifier Selection	*N/A*	*N/A*	*N/A*	**49.38**

From these experiments we can derive the following preliminary conclusions:

- In general the performance is higher with Random Forest than Naive Bayes as base classifier, both for the proposed method and for the standard top-down approach.
- There is no important difference between the three alternative methods for combining the results in the HHC, so any of them can be used as selection method.
- The proposed HHC has in general a better performance than the standard top-down hierarchical classifier in terms of both, standard and hierarchical precision, and in many cases the difference is statistically significant.
- The HHC is competitive with the top-down classifier selection method, using the same technique for all the local classifiers.

Running times

We also evaluated the HHC efficiency in terms of running time, and compare it with the same alternative methods. For this we considered the average training plus classification times of each method in the 5 experiments in the REUTERS domain, as it is the largest dataset (considering # examples × # attributes). The results are summarized in Table 5[5].

From this results we observe, on one hand, that the HHC with the MP and SP alternatives is similar in terms of efficiency to the standard top-down approach, and slightly less efficient with the DOP alternative. On the other hand, the HHC is more efficient than the top-down classifier selection method; between 3.8 and 35 times faster depending mainly on the base classifier.

In summary, the proposed HHC is superior in terms of performance and similar in terms of efficiency to the standard top-down approach, and competitive in terms of predictive performance but more efficient than the classifier selection method.

Conclusions and Future Work

We proposed a novel alternative for hierarchical classification that predicts a set of classes using a hybrid approach. The method learns a multi-class classifier for each parent node in the hierarchy. Contrary to previous approaches, during classification, all the local classifiers are applied *simultaneously* to each instance and the output is a set of classes that

[5] Intel Processor Core I5 at 2.53GHz with 6GB of RAM, under Windows 7.

Table 5: Training and classification times (in minutes) for each hierarchical classifier for the REUTERS database.

Classifier	DOP		MP		SP		Top-Down	
	Train	Test	Train	Test	Train	Test	Train	Test
Random Forest	7.07	0.08	7.06	0.05	7.05	0.04	7.05	0.03
Naive Bayes	6.02	0.02	6.02	0.02	5.59	0.03	6.01	0.02
Classifier Selection							**42.4**	0.62

corresponds to the most probable path in the hierarchy. We introduced three global strategies for obtaining these paths, based on multiplication, sum and sorting of the individual class probabilities.

The proposed method was compared in terms of predictive performance and efficiency against the standard top-down approach, and a state-of-the-art technique that performs classifier selection. From the results of these experiments we can conclude that the proposed method is superior to the standard approach for hierarchical classification, and has a very competitive performance against a state-of-the-art algorithm but it is much more efficient.

As future work we plan to extend our method to consider non-mandatory leaf prediction and DAG hierarchies.

References

Barutçuoglu, Z., and DeCoro, C. 2006. Hierarchical shape classification using bayesian aggregation. In *Shape Modeling International*, 44. IEEE Computer Society.

Blockeel, H.; Schietgat, L.; Struyf, J.; Džeroski, S.; and Clare, A. 2006. Decision trees for hierarchical multilabel classification: a case study in functional genomics. In *Proc. 10th European conference on Principle and Practice of Knowledge Discovery in Databases*, PKDD'06, 18–29. Springer-Verlag.

Dumais, S. T., and Chen, H. 2000. Hierarchical classification of web content. In *SIGIR*, 256–263.

Escalante, H. J.; Hernández, C. A.; González, J. A.; López-López, A.; y Gómez, M. M.; Morales, E. F.; Sucar, L. E.; Pineda, L. V.; and Grubinger, M. 2010. The segmented and annotated iapr tc-12 benchmark. *Computer Vision and Image Understanding* 114(4):419–428.

Freitas, A., and de Carvalho, A. C. 2007. *A Tutorial on Hierarchical Classification with Applications in Bioinformatics.*, volume Research and Trends in Data Mining Technologies and Applications. Idea Group. chapter VII, 175–208.

Holden, N., and Freitas, A. A. 2008. Improving the performance of hierarchical classification with swarm intelligence. In Marchiori, E., and Moore, J. H., eds., *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, 6th European Conference*, volume 4973 of *Lecture Notes in Computer Science*, 48–60. Springer.

Kiritchenko, S.; Matwin, S.; Nock, R.; and Famili, A. F. 2006. Learning and evaluation in the presence of class hierarchies: application to text categorization. In *Proc. 19th international conference on Advances in Artificial Intelligence*, AI'06, 395–406. Springer-Verlag.

Koller, D., and Sahami, M. 1997. Hierarchically classifying documents using very few words. In *Proc. Fourteenth International Conference on Machine Learning*, ICML '97, 170–178. San Francisco, CA, USA: Morgan Kaufmann.

Ruepp, A.; Zollner, A.; Maier, D.; Albermann, K.; Hani, J.; Mokrejs, M.; Tetko, I.; Güldener, U.; Mannhaupt, G.; Münsterkötter, M.; and Mewes, H. W. 2004. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res.*

Secker, A.; Davies, M. N.; Freitas, A. A.; Timmis, J.; Mendao, M.; and Flower, D. R. 2007. An experimental comparison of classification algorithms for the hierarchical prediction of protein function.

Secker, A.; Davies, M. N.; Freitas, A. A.; Clark, E. B.; Timmis, J.; and Flower, D. R. 2010. Hierarchical classification of G-protein-coupled receptors with data-driven selection of attributes and classifiers. *Int. J. of Data Mining and Bioinformatics* 4:191–210.

Silla-Jr., C. N., and Freitas, A. A. 2009. Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In *SMC*, 3499–3504. IEEE.

Silla-Jr., C. N., and Freitas, A. A. 2011. A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov* 22(1-2):31–72.

Toutanova, K.; Chen, F.; Popat, K.; and Hofmann, T. 2001. Text classification in a hierarchical mixture model for small training sets. In *CIKM*, 105–112. ACM.

Valentini, G. 2009. True path rule hierarchical ensembles. In *Proc. 8th International Workshop on Multiple Classifier Systems*, MCS '09, 232–241. Springer-Verlag.

Vens, C.; Struyf, J.; Schietgat, L.; Džeroski, S.; and Blockeel, H. 2008. Decision trees for hierarchical multi-label classification. *Mach. Learn.* 73(2):185–214.

Wang, K., and Zhou, S. 2001. Hierarchical classification of real life documents. In *Proc. 1st SIAM International Conference on Data Mining*.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval* 1(1-2):69–90.