Expert Systems with Applications 41 (2014) 7671–7677

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Multidimensional hierarchical classification

Julio Hernández *, L. Enrique Sucar, Eduardo F. Morales

Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro No. 1, Sta. Ma. Tonantzintla, Puebla, Mexico

ARTICLE INFO

Article history:
Available online 23 June 2014

Keywords:
Hierarchical classification
Multidimensional classification
Information gain

ABSTRACT

Hierarchical classification can be seen as a multidimensional classification problem where the objective is to predict a class, or set of classes, according to a taxonomy. There have been different proposals for hierarchical classification, including local and global approaches. Local approaches can suffer from the *inconsistency* problem, that is, if a local classifier has a wrong prediction, the error propagates down the hierarchy. Global approaches tend to produce more complex models. In this paper, we propose an alternative approach inspired in multidimensional classification. It starts by building a multi-class classifier per each parent node in the hierarchy. In the classification phase, all the local classifiers are applied *simultaneously* to each instance, providing a probability for each class in the taxonomy. Then the probability of the subset of classes, for each *path* in the hierarchy, is obtained by combining the local classifiers results. The path with highest probability is returned as the result for all the levels in the hierarchy. As an extension of the proposal method, we also developed a new technique, based on information gain, to classifies at different levels in the hierarchy. The proposed method was tested on different hierarchical classification data sets and was compared against state-of-the-art methods, resulting in superior predictive performance and/or efficiency to the other approaches in all the datasets.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

A traditional classification process consists of assigning a class c, from a finite set C of classes, to a single instance $\mathbf{x}$, represented by a feature vector. A dataset $\mathbf{D}$, for this kind of classification, is composed of n examples: $(\mathbf{x_1}, c_1), \ldots, (\mathbf{x_n}, c_n)$. In contrast, the multidimensional classification process, assigns a subset of classes $J \subseteq C$ to a single instance $\mathbf{x}$. A dataset $\mathbf{D}$ for a multidimensional problem is composed of n examples: $(\mathbf{x_1}, \mathbf{j_1}), \ldots, (\mathbf{x_n}, \mathbf{j_n})$, where $\mathbf{j_i}$ is a class vector.

Hierarchical classification is a variant of the multidimensional task with the classes arranged in a hierarchy. This hierarchy can be either a tree or a Directed Acyclic Graph (DAG), where each node corresponds to a class. There are many fields where hierarchical classification has gain popularity, like musical genre classification (Silla & Freitas, 2009), web content (Dumais & Chen, 2000), bio-informatics (Valentini, 2009), and computer vision (Barutcuoglu & DeCoro, 2006); among others.

Different alternatives have been proposed for hierarchical classification, including local (top–down) and global (big-bang) approaches (Silla & Freitas, 2011). 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 miss-classifies the error propagates down the hierarchy. The global approach results in a more complex model which in general has not better predictive results than the local approaches.

In this work, we propose an alternative approach inspired on non-hierarchical multidimensional classification techniques.[1] In the training phase, a multi-class classifier per parent node in the hierarchy is built. 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 obtained. Then, a set of *consistent* classes, according to the hierarchy, is obtained. For this, the probability of the subset of classes $c_1, c_2, \ldots, c_l$ of each path q in the hierarchy is estimated. The path q^* with highest probability provides the resulting classes for all levels in the hierarchy.

In addition, we developed an extension of the hierarchical multidimensional classifier based on an information gain measure with the objective to make a prediction at any level in the hierarchy; in other words, a *non-mandatory leaf-node prediction*. This extension considers backtracking from the bottom classifier in

* Corresponding author. Tel.: +52 012222315058.
E-mail address: julio.hernandez.t@ccc.inaoep.mx (J. Hernández).

[1] Although the idea is inspired on multidimensional classification, the work is not directly related to recent work in non-hierarchical multidimensional classification, e.g. Read, Pfahringer, Holmes, and Frank (2009).

http://dx.doi.org/10.1016/j.eswa.2014.05.054
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

the hierarchy when there is not enough *confidence* in their results, returning a partial class subset.

We evaluated the proposed method with hierarchical classification data sets in different domains: text, images and genes; considering two different base classifiers: Naive Bayes and Random Forest. We compared the results in terms of standard and hierarchical precision measures against a top–down approach using the same base classifiers, and also a top–down method that does classifier selection for each node and is one of the current top performing techniques in the literature (Secker et al., 2007, 2010). For all the data sets our method has superior or similar performance than the other approaches, and it is also much more efficient than the top–down classifier selection method.

2. Hierarchical classification

According to Freitas and de Carvalho (2007, chap. VII) and Sun and Lim (2001), 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 node in the hierarchy, also known as *non-mandatory leaf-node prediction*. The final criterion is related on 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, in binary or multi-class problems, is the local or top–down approach. The training phase can be performed in three different ways: (1) using binary classifier per node, except the root node, (2) using multi-class classifier per parent node and (3) using multi-class classifier per level. In the classification phase the first classifier decides where the example belongs and passes the example to the classifier of the immediate level, this procedure is repeated until the example reaches a leaf node.

Typically, every node in the hierarchy uses the same classification algorithm. An important limitation of this type of methods is the *inconsistency problem*: A classification error, at any level of the hierarchy, will be propagated to all its descendants.

The use of binary classifiers are the most common way to face a hierarchical classification problem (Esuli, Fagni, & Sebastiani, 2008; Barutcuoglu & DeCoro, 2006; Bennett & Nguyen, 2009; Otero, Freitas, & Johnson, 2009; Valentini, 2009), but the multi-class classifiers are more adaptable to problems with big taxonomies.

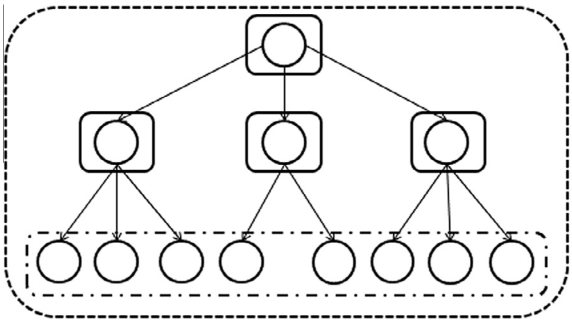

Fig. 1. Types of hierarchical classifiers. (i) Local classifier: a multi-label classification algorithm is used per parent node. The circles represent the classes and the solid squares represent multi-label classifiers. (ii) Global classifier: a classification algorithm (dashed square) that learns a global classification model that takes into account the whole hierarchy. (iii) Flat classifier: a flat multi-label classification algorithm (dash-dot square) which only predicts the leaf nodes.

Some binary approaches take into account the prediction of all classifiers in order to avoid the inconsistency problem, however, their taxonomies have a few number of levels. For example, In Weigend, Wiener, and Pedersen (1999) and Dekel, Keshet, and Singer (2005) the predicted probability in each node is multiplied according with the different paths. In Dumais and Chen (2000) the authors propose two methods,based on a threshold which need to be adjusted for every taxonomy, that consider the output of all binary classifiers in the hierarchy. In others approaches is used a pruning strategy (Valentini, 2009; Xue, Xing, Yang, & Yu, 2008), where the basic idea is that by evaluating all the classifier outputs it is possible to make consistent predictions by computing a consensus probability. Again, the use of a threshold is fundamental for the mentioned work. In Bi and Kwok (2012) they formulate the hierarchical classification problem as an optimization problem which is solved with a greedy optimization algorithm, called CSSA.

The problem with binary approaches is that used too many classifiers. For taxonomies with an important number of nodes the whole classification process will take a lot of time. Furthermore, if the taxonomy grows the probability to occur an inconsistency problem will grow. In our work we focused in the use of multi-class classifiers to avoid this drawbacks.

An alternative to binary approach is the multi-class classifier per parent node (Dumais & Chen, 2000; Holden & Freitas, 2008; Secker et al., 2007;Silla & Freitas, 2009). The use of multi-class classifiers has the advantage of use a fewer number of classifiers than the binary alternative. The multi-class classifiers are more adaptable to problems with big taxonomies compare to binary classifiers. The basic approach used the same classifier algorithm per each parent node. In Secker et al. (2007) the authors proposed an alternative strategy based on the premise that each local classifier should be adapted to the particular problem it solves. They developed the *classifier selection* technique in which a different classifier is selected at each node in the hierarchy from a set of possible models, based on the performance in a validation set. From this work several extensions have been developed (Holden & Freitas, 2008; Secker et al., 2010; Silla & Freitas, 2009). In general, these methods improve the performance of local approaches that use the same base classifier for all the hierarchy; however there is also a significant increase in the training time.

The main drawback with this approaches is that they treat each classifier as independent; the result of each classifier has no relation with the result of the other classifiers.

In our work we follow a multidimensional approach applied to the hierarchical classification problem. We take into account the prediction of all the classifiers using multi-class classifiers per parent node. We are interested in finding the best classification path, and we use a much simpler approach than the selection of classifiers that involves the product of the probabilities in each path. We also include a non-mandatory leaf prediction criterion based on information gain.

3. A multidimensional hierarchical classifier

In this section we describe the multidimensional hierarchical classifier (MHC). First we present the basic algorithm that predicts at all levels of the taxonomy, and then the extension for non-mandatory leaf node prediction.

We initially considered a tree-structured taxonomy, T, with $|t|$ nodes, each node represents a class. There are $|c|$ non-leaf nodes and $|l|$ leaf nodes, such that $\mid t \mid = \mid c \mid + \mid l \mid$. Each non-leaf node c_i has ns_i sons, which represent the direct subclasses of class c_i. We define a path q in the taxonomy graph as any set of nodes from the root to a leaf (following a trajectory), and assume that there are $\mid q \mid$ paths in the taxonomy. We also assume that there are m

attributes for each class, such that the same set of attributes are considered for all the classes (see Fig. 2).

The algorithm includes two phases: training and classification.

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

1. Partition the data base according to the subclasses (sons) of each non-leaf node c_i.
2. Learn a multi-class classifier for each non-leaf node c_i to classify its ns_i sons.

To train each local multi-class classifier, we consider the predefined taxonomy of each database. That is, all instances in the data set that correspond to the sons (subclasses) of the node c_i are considered, including their descendants; to train a classifier with ns_i classes.

Classification: Given an instance $\mathbf{x}$:

1. Classify $\mathbf{x}$ with all the $\mid c \mid$ local classifiers.
2. Combine the results of all the classifiers to obtain the probabilities for all the paths, $P(q_1),P(q_2),\ldots,P(q_{|q|})$,
3. Return the path q^* with highest probability.

The probability of each path is obtained by multiplying the probabilities given by the local classifiers in the path, as exemplified in Fig. 3. Next we provide a theoretical justification for this procedure.

The probability of the subset of classes $c_1,c_2,\ldots c_l$ (where 1 is the root and l a leaf) of a path h given the vector of attributes $\mathbf{x}$ is by the Chain rule:

$$P(c_l,c_{l-1},\ldots c_1\,|\,\mathbf{x}) = P(c_l\,|\,c_{l-1},\ldots,c_1,\mathbf{x})P(c_{l-1}\,|\,c_{l-2},\ldots,c_1,\mathbf{x})\vdots P(c_1\,|\,\mathbf{x}) \tag{1}$$

Given that each path is defined over a taxonomy:

$$c_l \subset c_{l-1} \subset \cdots \subset c_2 \subset c_1 \tag{2}$$

Each subclass, c_k, is independent of its ancestors (super-classes) given its direct parent in the taxonomy; thus, we can simplify Eq. (1) to:

$$P(c_l,c_{l-1},\ldots c_1 \mid \mathbf{x}) = P(c_l \mid c_{l-1},\mathbf{x})P(c_{l-1} \mid c_{l-2},\mathbf{x})\vdots P(c_1 \mid \mathbf{x}) \tag{3}$$

We make a further simplification by assuming that knowing the parent class does not have a significant impact on the probability of a class given its attributes, so we can rewrite Eq. (3) as:[2]

$$P(c_l,c_{l-1},\ldots c_1 \mid \mathbf{x}) = P(c_l \mid \mathbf{x})P(c_{l-1} \mid \mathbf{x})\ldots P(c_1 \mid \mathbf{x}) \tag{4}$$

So we can obtain the joint probability of the subset of classes of each path in the taxonomy with the product of the local classifiers for each node in the path (note that the probability of the root is not considered as this is one for a tree-structured taxonomy).

3.1. Non-mandatory leaf prediction based on information gain

The previous algorithm assumes that we have enough information to predict the class of an instance at all the levels in the taxonomy. We present an extension to predict a class up to certain level in the hierarchy when there is not enough confidence in the results at the leaf nodes. Our method decides the *best* level to stop the classification based on information gain.

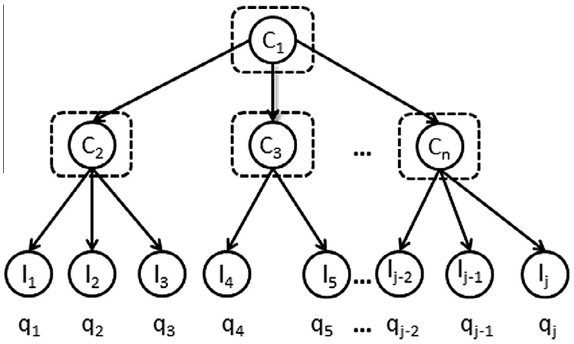

Fig. 2. Example of a tree-structured taxonomy. q_i represent the different paths in the hierarchy, c_i represent the non-leaf nodes and l_i represent the leaf nodes in the hierarchy.

The information gain is applied to each local classifier in the selected path (global prediction) in a bottom-up fashion. The information gain is defined as:

$$IG(c_i) = E(c_i) - \sum_{j=1}^{ns} w_{ch_j(c_i)} * E(ch_j(c_i)) \tag{5}$$

where E represents the entropy, $ch_j(c_i)$ is the child j of node i, and w is the weight of each child class of the node i, which corresponds to the proportional number of examples that belongs to each child.

If the information gain is less than zero for the bottom classifier, the decision is set as *unknown* for this classifier, and the next classifier, one level up, is analyzed. Otherwise, if the information gain is greater than zero, the prediction of the local classifier is accepted and the analysis ends. If the top level classifier is reached the process also ends. The subset of predicted classes (that could be from 1 to q, where q is the number of local classifiers in the path) is returned as the global prediction.

3.2. Extension to DAG taxonomies

In a DAG taxonomy there could be more than one path for a leaf node, for example in an image taxonomy a *turkey* could be a subclass of *animal* and *food*. The MHC was developed initially for tree-structured taxonomies; next we extend the basic algorithm for DAG structures.

The extension for DAGs implies modifying the training and classification stages. In the training stage we have to consider that a class, c_i, could have multiple parents, $pc_i^1,\ldots,pc_i^m$. Thus, when the multi-class classifier for each parent node is built, each one will contain c_i among its sons. We maintain c_i in all the m local classifiers.

In the classification stage, each local classifier will provide a different probability estimate, $P(c_i)^1,\ldots,P(c_i)^m$, for a multi-parent node c_i. Thus, a multi-parent node will have associated several probabilities, one per parent, which will be used to obtain the combined probabilities of each path that passes through that node. To obtain the probabilities of each path, we consider all possible paths between the root(s) and the leaf nodes. In the case of a multi-parent node there will be several paths that pass through that node, so the corresponding local probability will be used for calculating the path probability. For instance, if node c_i has two parents, pc_i^1 and pc_i^2, the probability of the path that comes from parent 1 will include the local probability $P(c_i)^1$, and the path that comes from parent 2, $P(c_i)^2$ (see Fig. 4).

Finally, the paths with highest probability, along with all the alternative connecting paths up the hierarchy, are returned as the set of classes for that instance.

[2] We performed preliminary experiments in which we incorporated the probability of the parent class as an additional attribute – in the spirit of chain classifiers (Read et al., 2009) –, but there was no significant difference in performance.

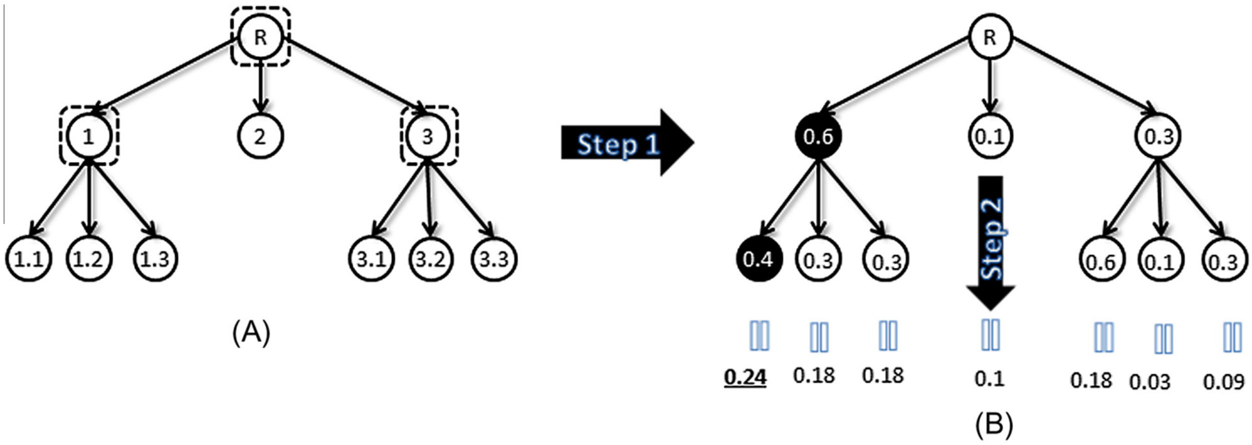

Fig. 3. An example of the calculation of the probabilities for each path. (A) The class taxonomy. (B) Each node (except the root) depicts the predicted probability of its class. The probabilities in each trajectory in the tree – for instance $P_1 \times P_{1.1}$ – are multiplied and the results are shown below; the trajectory with the highest probability is selected (underlined).

In the case of non-mandatory leaf prediction, the returned path is followed in inverse order, from leaf to root in the graph. If a node with multiple parents is found in the trajectory, the original path (with the highest probability) is followed towards the root, not considering alternative paths.

Although in the case of DAG taxonomies, the classifier could return a set of classes that includes several sub paths in the hierarchy, in this work we do not consider the case of multi-label hierarchical classification; that is, when an instance can be assigned to several paths in the hierarchy at the same time (this is left as future work).

4. Experiments and results

We evaluated MHC with four hierarchical databases and compared its performance against state of the art top–down classifiers.

First we describe the data sets, then present the experiments and results, to conclude with an analysis.

4.1. Datasets

We consider four hierarchical datasets from different domains: *Reuters-21578*, *FunCat* (Ruepp et al., 2004), *IAPR-TC12* (Escalante et al., 2010), and *MIREX 2005* (Mckay & Fujinaga, 2005).

Reuters-21578[3] is a popular database for text retrieval (Yang, 1999). It has 135 categories and a taxonomy proposed in Toutanova, Chen, Popat, and Hofmann (2001). *FunCat*[4] is a database in the domain of bio-informatics, in particular for protein function prediction (Ruepp et al., 2004). It includes 27 categories and for this work we only consider the category *Cellcycle*. *IAPR-TC12*[5] (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 six main categories.

For this work we consider the category *Landscape-Nature*. *MIREX 2005* (Mckay & Fujinaga, 2005) is a database for musical genre classification. In our evaluation, we used the Bodhidharma (Mckay & Fujinaga, 2004) features with a 38-leaf class hierarchy and 950-item symbolic genre dataset. The main properties of the four datasets are summarized in Table 1.

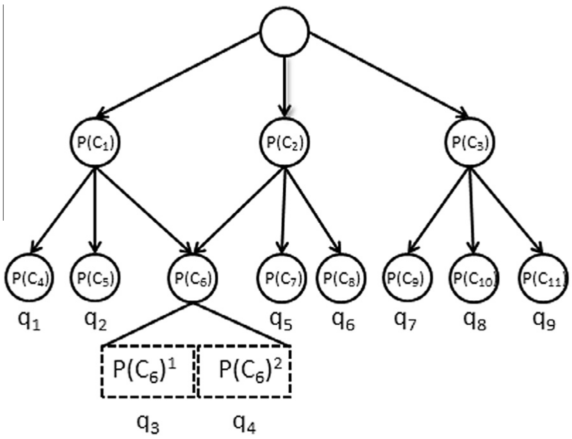

Fig. 4. Example of the test phase for a dag-structured taxonomy applying the MHC. q_i represents the different paths and $P(C_i)$ represents the probabilities predicted for the corresponding nodes. The dash squares represents the different probabilities predicted for the node with more than one parent.

4.2. Algorithms

We compared the proposed MHC 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, one uses as base classifier *Naive Bayes* and the other *Random Forest*. The third scheme uses the classifier selection method proposed by Secker et al. (2007). For each local classifier it select the *best* technique for the following set: Naive Bayes, Bayes Net, SVM, AdaBoost, 3-KNN, PART, J48 and Random Forest from Weka. This method has shown superior performance than the top–down approach using the same classifier for each node in the hierarchy.

We only compared the proposed approach against top–down classification techniques, as these suffer from the inconsistency problem, which our method tries to solve. Additionally, according to a recent survey (Silla & Freitas, 2011), the top–down classifier selection methods is one of the top general hierarchical classifiers in terms of performance (other approaches with good performance tend to be restricted to certain domains).

For MHC 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 evaluated the basic method

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

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

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

Table 1
Characteristics of the databases used in the experiments.

DataBase	Domain	# Classes	# Examples	# Levels	# Attributes	Type of hierarchy
FunCat[a]	Genetics	30	1433	3	77	Tree
Reuters-21578[b]	Text	25	6274	2	16145	Tree
IAPR-TC12[c]	Image	25	45347	2	23	Tree
MIREX2005	Music	38	950	2	100	DAG

[a] Subset *Cellcycle* of the original hierarchy.
[b] Subset *R52* of the taxonomy.
[c] The *Landscape* branch of the original hierarchy.

Table 2
Experimental results for the FUNCAT, REUTERS and IAPR-TC12 databases. Precisions in percentage. The scheme with the highest hierarchical/standard precision is shown in bold. An "*" is shown if it is statistically significantly better than the top–down classifier; and a "†" if it is statistically significantly better than the classifier selection method.

Base classifier	MHC		Top–down
	NMLP	MLP	
FUNCAT			
Hierarchical precision			
Naive Bayes	29.49	28.78	28.10
Random Forest	28.82*	27.72	26.93
Classifier selection	*N/A*	*N/A*	**31.11**
Standard precision			
Naive Bayes	22.22*†	16.67†	16.35
Random Forest	**26.51***†	17.94*†	13.33
Classifier selection	*N/A*	*N/A*	14.92
REUTERS			
Hierarchical precision			
Naive Bayes	78.15*	76.71	76.11
Random Forest	**90.04***	84.79	83.54
Classifier selection	*N/A*	*N/A*	89.27
Standard precision			
Naive Bayes	75.69*†	70.01	70.01
Random Forest	**91.27***†	79.04*	77.32
Classifier selection	*N/A*	*N/A*	85.40
IAPR-TC12			
Hierarchical precision			
Naive Bayes	51.81*†	50.84 *†	37.71
Random Forest	54.62*†	**55.35***†	44.65
Classifier selection	*N/A*	*N/A*	45.19
Standard precision			
Naive Bayes	55.59*†	39.52	41.72
Random Forest	**60.36***†	47.45	47.98
Classifier selection	*N/A*	*N/A*	49.38

(MBC), and the extension for non-mandatory leaf node prediction based on information gain (MHC-NMLP).

4.3. 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:

$$hP = \frac{\sum_i |\hat{C}_i \cap C_i|}{|C_i|} \tag{6}$$

where $\hat{C}_i$ is the set of predicted classes for the test sample $\mathbf{x}$ and C_i is the actual set of classes for $\mathbf{x}$; the class set includes the more specific class and all its ascendants in the hierarchy.

To perform the experiments we used the stratified five-fold cross validation procedure. Tables 2 and 3 summarize the results for the four datasets. Each cell has the symbol "*" if the precision reported in that cell is statistically significantly better than the precision reported of the corresponding top–down classifier. Likewise, each cell has the symbol "†" if the precision reported in that cell is statistically significantly better than the precision of the classifier selection method. Statistical significance was measured by the paired two-tailed Student's t-test, using a confidence level of 95%.

In these tables, *MLP* corresponds to the basic method that returns the complete path and *NMLP* specifies that the option for non-mandatory leaf prediction based on information gain was used. The results of the two variants of the MHC are depicted, considering the two base classifiers (Naive Bayes and Random Forest). For comparison the standard top–down approach is contrasted with the two variants, and also the top–down classifier selection method.

In general, MHC is statistically significantly better than the top–down approach for the four data sets. Compared to the top–down classifier selection method, the results are similar for the *Funcat* and *Reuters* datasets, and MHC is statistically significantly better in the case of the *IAPR-TC12* and *MIREX* datasets. For almost all the results, the information gain alternative presents better results, so it seems that the idea of stopping the classification at certain level based on information gain is useful. In terms of the base classifier, in most cases Random Forest has a superior predictive performance, although in a few cases Naive Bayes has better results.

4.4. Running times

We also compared the training and classification times of the MHC (MLP) versus the standard top–down and the top–down classifier selection methods. For this we considered the average training and 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 4. (Intel Processor Core I5 at 2.53 GHz with 6 GB of RAM, under Windows 7.) From this table, it can be observed that the MHC is similar in terms of efficiency to the standard top–down approach when the same base classifier is used. However, it is between 6 and 7 times faster in training time than the classifier selection method, depending on the base classifier.

In summary, the proposed approach is significantly better in terms of standard and hierarchical precision to the top–down approach; and it is very competitive and in most cases superior, in terms of predictive performance, to a state-of-the-art algorithm that selects the best classifier at each node (classifier selection), and at the same time it is significantly faster.

4.5. Analysis

From the results of these experiments we can derive the following conclusions:

1. MHCs reduce the inconsistency problem as demonstrated with their superior performance with respect to the top–down approach using the same base classifier.

Table 3
Experimental results of the MIREX 2005 data set. Precisions in percentage. The scheme with the highest hierarchical/ standard precision is shown in bold. An "*" is shown if it is statistically significantly better than the top–down classifier; and a "†" if it is statistically significantly better than the classifier selection method.

Base classifier	MHC		Top–down
	NMLP	MLP	
Hierarchical precision			
Naive Bayes	52.61*	48.76*	47.32
Random Forest	**59.9***†	55.12*	51.82
Classifier selection	*N/A*	*N/A*	58.52
Standard precision			
Naive Bayes	47.2*	37.4	43.4
Random Forest	**53.6***†	44.6*	43.4
Classifier selection	*N/A*	*N/A*	49.38

Table 4
Training and classification times (in seconds) for each hierarchical classifier for the REUTERS dataset.

Classifier	MHC		Top–down	
	Train	Test	Train	Test
Random Forest	7.06	0.05	7.05	0.03
Naive Bayes	6.02	0.04	6.01	0.02
Classifier selection			42.4	0.12

2. In the comparison between the MHC and the top–down classifier selection approach, there is not clear winner in terms of precision. Although MHC reduces the inconsistency problem, its local classifiers are not optimized as in the classifier selection technique, so depending on the database, one of these aspects could be more important.
3. In general the non-mandatory leaf prediction option has higher standard and hierarchical precisions, by eliminating part of the path based on information gain.
4. It seems that the benefit of the MHC depends on the difficulty of the problem as indicated by the precision of the standard top–down approach. In the databases where the precision of the top–down is about 50% or higher, the precision of the HMC is nearly 10 points higher and it is also significantly better than the classifier selection method.
5. The additional computational effort of the MHC with respect to the top–down with the same base classifier is minimum, resulting in very similar training times; however, the classifier selection method requires more training time.

5. Conclusions and future work

In this paper, we have described a multidimensional hierarchical classification algorithm that starts by building a multi-class classifier for each parent node in the hierarchy. Contrary to previous approaches, during classification phase, all the local classifiers are applied *simultaneously* to each instance and the output is given by considering the most probable path. In the traditional way only one path is considered through all the classification process. According with the experiments, if we consider all the possible paths the occurrence of the inconsistency problem will be reduced. We also developed an extension to decide when to stop in the hierarchy based on information gain, contrary to those based on threshold.

The main contribution of this work is a novel hierarchical classification scheme based on multidimensional classification. This approach explores all the possible paths avoiding the inconsistency problem. Additionally, our work incorporates non-mandatory leaf prediction based on information gain. Finally, our work can be applied to tree and DAG taxonomies.

The result of the experiments demonstrates that our work improves the traditional top–down approach for multi-class classifiers, however our results was not superior with respect to the classifier selection approach. The exploration of all the hierarchy implies the sum or multiplication of all the classification results in each path, however, we did not prove our method on imbalanced taxonomies. The information gain for non-mandatory leaf prediction have good results but we need to do more experiments to strengthen the results presented in this work.

As future work we plan to experiment with the idea of incorporate the information of each parent class predictions to their immediate predecessor along the different paths to improve the performance. This process is known as chain classification. Due that we explore the entire hierarchy, we want to extend the approach for multi-label hierarchical classification. We would also like to explore combining the outputs of different types of classifiers.

References

Barutcuoglu, Z., & DeCoro, C. (2006). Hierarchical shape classification using bayesian aggregation. In *Proceedings of the IEEE international conference on shape modeling and applications 2006, SMI '06* (p. 44). Washington, DC, USA: IEEE Computer Society.

Bennett, P. N., & Nguyen, N. (2009). Refined experts: Improving classification in large taxonomies. In *Proceedings of the 32Nd international ACM SIGIR conference on research and development in information retrieval, SIGIR '09* (pp. 11–18). New York, NY, USA: ACM.

Bi, W., & Kwok, J. T. (2012). Hierarchical multilabel classification with minimum Bayes risk. In *ICDM* (pp. 101–110).

Dekel, O., Keshet, J., & Singer, Y. (2005). An online algorithm for hierarchical phoneme classification. In *Proceedings of the first international conference on machine learning for multimodal interaction, MLMI'04* (pp. 146–158). Berlin, Heidelberg: Springer-Verlag.

Dumais, S., & Chen, H. (2000). Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '00* (pp. 256–263). New York, NY, USA: ACM.

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

Esuli, A., Fagni, T., & Sebastiani, F. (2008). Boosting multi-label hierarchical text categorization. *Information Retrieval, 11*(4), 287–313.

Freitas, A., & de Carvalho, A. C. (2007). A tutorial on hierarchical classification with applications in bioinformatics. In *Research and trends in data mining technologies and applications* (pp. 175–208). Idea Group.

Holden, N., & Freitas, A. A. (2008). Improving the performance of hierarchical classification with swarm intelligence. In E. Marchiori & J. H. Moore (Eds.), *Evolutionary computation, machine learning and data mining in bioinformatics. Lecture notes in computer science* (Vol. 4973, pp. 48–60). Springer.

Mckay, C., & Fujinaga. (2004). Automatic genre classification using large high-level musical feature sets. In *International conference on music information retrieval, ISMIR 2004* (pp. 525–530).

Mckay, C., & Fujinaga. (2005). The bodhidharma system and the results of the mirex 2005. In *International conference on music information retrieval.*

Otero, F. E. B., Freitas, A. A., & Johnson, C. G. (2009). A hierarchical classification ant colony algorithm for predicting gene ontology terms. In C. Pizzuti, M. D. Ritchie, & M. Giacobini (Eds.), *EvoBIO. Lecture notes in computer science* (Vol. 5483, pp. 68–79). Springer.

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classifier chains for multi-label classification. In *Proceedings of the European conference on machine learning and knowledge discovery in databases: Part II. ECML PKDD '09* (pp. 254–269). Berlin, Heidelberg: Springer-Verlag.

Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., et al. (2004). The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research, 32*(18), 5539–5545.

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

Secker, A., Davies, M. N., Freitas, A. A., Timmis, J., Mendao, M., & Flower, D. R. (2007). An experimental comparison of classification algorithms for the hierarchical prediction of protein function. *Expert Update, 9*(3), 17–22.

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

Silla, C. N., Jr., & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery, 22*(1-2), 31–72.

Sun, A., & Lim, E.-P. (2001). Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE international conference on data mining, ICDM '01* (pp. 521–528). Washington, DC, USA: IEEE Computer Society.

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

Valentini, G. (2009). True path rule hierarchical ensembles. In *Proceedings of the eighth international workshop on multiple classifier systems, MCS '09* (pp. 232–241). Springer-Verlag.

Weigend, A. S., Wiener, E. D., & Pedersen, J. O. (1999). Exploiting hierarchy in text categorization. *Information Retrieval, 1*(3), 193–216.

Xue, G.-R., Xing, D., Yang, Q., & Yu, Y. (2008). Deep classification in large-scale text hierarchies. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '08* (pp. 619–626). New York, NY, USA: ACM.

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