A Centroid-based Approach for Hierarchical Classification

Mauri Ferrandin[1], Fabrício Enembreck[2], Júlio César Nievola[2], Edson Emílio Scalabrin[2]
and Bráulio Coelho Ávila[2]

[1]*Engenharia de Controle e Automação, Universidade Federal de Santa Catarina-UFSC, Campus Blumenau,*
Rua Pomerode, 710, 89065-300, Blumenau, SC, Brazil

[2]*Programa de Pós-Graduação em Informática, Pontifícia Universidade Católica do Paraná-PUCPR,*
Rua Imaculada Conceição, 1155, Prado Velho, 80215-901, Curitiba, PR, Brazil

Keywords: Data Mining, Hierarchical Classification, Centroid Classification.

Abstract: Classification is a common task in Machine Learning and Data Mining. Some classification problems need to take into account a hierarchical taxonomy establishing an order between involved classes and are called hierarchical classification problems. The protein function prediction can be considered a hierarchical classification problem because their functions may be arranged in a hierarchical taxonomy of classes. This paper presents an algorithm for hierarchical classification using a centroid-based approach with two versions named HCCS and HCCSic respectively. Centroid-based techniques have been widely used to text classification and in this work we explore it's adoption to a hierarchical classification scenario. The proposed algorithm was evaluated in eight real datasets and compared against two other recent algorithms from the literature. Preliminary results showed that the proposed approach is an alternative for hierarchical classification, having as main advantage the simplicity and low computational complexity with good accuracy.

1 INTRODUCTION

Classification is one of the most important problems in Machine Learning and Data Mining. The classification consists in associating one or more classes from a set of predefined classes to a not classified example (instance) from a database. The features (attributes) of each example will determine the classes it will be associated with.

The prediction of the functions of proteins is considered as a classification problem. The set of different proteins is considered the example database and the set of biological functions are the classes which can be associated to each example. The functions of the proteins may be arranged in a hierarchical taxonomy of classes, so the prediction of these functions is considered a hierarchical classification problem.

Centroid-based classifiers have been largely applied in text categorization problems showing good accuracy with low computational costs due to its simplicity in the representation of the information without to loose the capacity of summarize the main aspects present in the training examples.

Based on the wide diversity of hierarchical classification problems, specific algorithms in this area are being developed. This paper presents an algorithm for hierarchical classification that uses techniques of centroid-based classifiers that is an adaptation of the centroid-base algorithms used for text categorization. Experiments with biological data sets were done and the obtained results were compared with two other approaches - GMNB (Silla and Freitas, 2009) and HLCS (Romão and Nievola, 2012) - that were proposed to explore the same hierarchical classification problem.

The remainder of this paper is organized as follows: Section 2 presents background on hierarchical classification and centroid-based classifiers. Section 3 shows the related works in the subject of this paper. Section 4 discusses the new proposed algorithms for hierarchical classification. Section 5 presents the experimental setup and reports the computational results obtained with the algorithms proposed in this paper. Conclusions and some perspectives about future works are stated in Section 6.

2 BASIC FOUNDATIONS REVIEW

This section presents an overview about hierarchical classification problems and the main centroid-based techniques that will be used in this work.

Ferrandin M., Enembreck F., Nievola J., Scalabrin E. and Ávila B..
A Centroid-based Approach for Hierarchical Classification.
DOI: 10.5220/0005339000250033
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 25-33
ISBN: 978-989-758-096-3

2.1 Hierarchical Classification

A hierarchical classification problem has as main characteristic a taxonomy that imposes a hierarchical order between the set of classes present in the dataset. This order is represented by $(C, \prec)$ that represents a "IS-A" relationship among the classes and is asymmetric, reflexive and transitive where:

- The only one greatest element R is the root of the tree;
- $\forall c_i, c_j \in C$, if $c_i \prec c_j$ then $c_j \not\prec c_i$;
- $\forall c_i \in C$, $c_i \not\prec c_i$
- $\forall c_i, c_j, c_k \in C$, $c_i \prec c_j$ and $c_j \prec c_k$ imply $c_i \prec c_k$.

According to (Silla and Freitas, 2011b), three main features distinguish the hierarchical classification problems. Firstly the type of hierarchical taxonomy of classes which may be represented as a tree or a directed acyclic graph (DAG). Figure 1 represents both types of taxonomies with a tree - Figure 1 (a) - and a DAG - Figure 1 (b). In the representation, the "IS-A" relationship states that one instance that belongs to the class 2.2.1 also belongs to classes 2.2, 2 and root (R). When the taxonomy is represented as a DAG the scenario is even more complex because one class can have more than one parent node, so considering the representation in the Figure 1 (b) there are two classes named by 1.2/2.1 and 2.2.1/2.3.1 that have more than one parent and as consequence they also belongs to the classes of all of its ancestors nodes in different branches of the DAG.

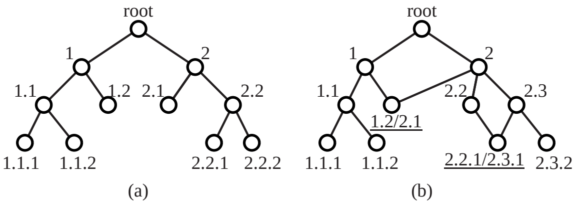

Figure 1: Different types of hierarchical class taxonomies. (a): tree-structured; (b): DAG-structured.

The second characteristic of the hierarchical classification problems is related to how deep the classification is performed in the hierarchy. That is, the hierarchical classification method can be implemented to always predict classes that are in the leaf nodes of the taxonomy - this approach is named as mandatory leaf node prediction (MLNP) - or the method can consider stopping the classification at any node from any level of the taxonomy - approach named non-mandatory leaf node prediction (NMLNP).

The third criterion considers how the hierarchical structure of the taxonomy is explored. The exploration could be local, when the system employs a set of local classifiers (i.e. one classifier per class is induced); global, when a single classifier is used to represent the entire class taxonomy; or flat classifiers which ignore the relationships among the classes, typically predicting only classes represented in the leaf nodes.

The algorithms for hierarchical classification can be classified by the three cited features and also by a additional property that indicates the capabilities of making single or multi-label predictions. Considering that the classes are organized by a taxonomy these capabilities are named Single Path of Labels (SPL) and Multiple Paths of Label (MPL) respectively.

2.1.1 Performance Measures for Hierarchical Classifiers

There are different measures used to evaluate the performance of hierarchical classifiers. The most accepted and used approach is an adaptation of traditional measures for classifiers known as Precision, Recall and F-Measure. In the context of a hierarchical classification problem, for each dataset the final values of hierarchical Precision (hP), hierarchical Recall (hR) and hierarchical F-measure (hF) are obtained by Equation 1 accordingly to the proposal of (Kiritchenko et al., 2005). In the Equation 1, according to the author we assume that one instance i belongs to a set of classes C and will have a set of predicted classes denoted by C'. The extended sets $\hat{C}$ and $\hat{C}'$ represent respectively the classes in the sets C and C' with the addition of all ancestors classes of each set considering the taxonomy.

$$hP = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i|}{\sum_i |\hat{C}'_i|},\ hR = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i|}{\sum_i |\hat{C}_i|},\ hF = \frac{2 * hP * hR}{hP + hR} \quad (1)$$

Although no hierarchical classification measure can be considered the best one in all possible hierarchical classification scenarios and applications, the main reason for recommending the hP, hR and hF measures is that, broadly speaking, they can be effectively applied to any hierarchical classification scenario; i.e., tree-structured, DAG-structured, SPL, MPL, MLNP or NMLNP problems (Silla and Freitas, 2011b).

2.2 Centroid-based Classification

Centroid-based approaches have been widely used in text categorization problems. In a centroid-based classification algorithm, the documents are represented using the vector-space model (Salton, 1989). According to (Han and Karypis, 2000), in this model, each document is considered to be a vector in the term-space. In its simplest form, each document is represented by the *term-frequency* (TF) vector as shown in Equation 2, where tf_i is the frequency of the ith term in the document.

$$d_{tf} = (tf_1, tf_2, ..., tf_n) \quad (2)$$

In addition, the *inverse document frequency* (IDF) refinement is commonly used as a refinement to consider that terms appearing frequently in many documents have limited discrimination power, and for this reason they need to be de-emphasized. This is done in (Salton, 1989) by multiplying the frequency of each term i by $log(N/df_i)$, where N is the total number of documents in the collection, and df_i is the number of documents that contains the ith term (i.e., document frequency).

The *tf-idf* representation explained above leads to a representation of a document as represented in the Equation 3. Finally, to deal with documents with different lengths the vectors are normalized to $||d_{tfidf}||_2 = 1$.

$$d_{tfidf} = (tf_1\ log(\frac{N}{df_1}), tf_2\ log(\frac{N}{df_2}), ..., tf_n\ log(\frac{N}{df_n})) \quad (3)$$

Considering a set S of documents belonging to a class x represented by it's *tf-idf* vectors, a centroid C_x is obtained by the average of the terms of the documents as represented in Equation 4.

$$C_x = \frac{1}{|S|} \sum_{d \in S} d \quad (4)$$

The classification process consists of to compute a centroid for each class in the training dataset. If there are k classes in the training set, this leads to a set of centroid vectors $\{C_1, C_2, ..., C_k\}$, where each C_i is the centroid for the ith class. The class of a new document x is determined as follows. First we use the document-frequencies of the various terms computed from the training set to compute the *tf-idf* weighted vector-space representation of x. Then, we compute the similarity between x and all centroids using the cosine measure as shown in the Equation 5.

$$cos(x, C) = \frac{x \cdot C}{||x||\ ||C||} \quad (5)$$

Finally, based on the obtained similarities measures, we assign the example x being classified to the class corresponding to the most similar centroid. This test phase is represented by Equation 6.

$$\arg \max_{j=1,..,k} (cos(x, C_j)) \quad (6)$$

Although the centroid-based classifier approach is considered very simple, it has the advantage that it's computational complexity of the learning phase is linear on the number of documents and the number of terms in the training set, the amount of time required to classify a new document is at most $O(km)$, where k is the number of classes and m is the number of terms present in x. Thus, the overall computational complexity of this algorithm is very low, and it is identical to fast document classifiers such as Naive Bayesian (Han and Karypis, 2000).

3 RELATED WORKS

In this Section we present the main works in the areas related to this paper, firstly the works related to hierarchical classification showing the historical and mainly works in the subject, and secondly the main works related to centroid-based classification.

3.1 Related Works in Hierarchical Classification

There are a great number of researches and proposals addressed to hierarchical classification problems. Some of them were created having as base other proposals addressed to text classification or other domain. In these works a lot of different approaches were used considering the built of local and global classification systems. Bellow the most recent and remarkable works are presented.

In the paper proposed by (Vens et al., 2008) the authors developed a hierarchical classification model named Clus for the DAG structure using the global approach capable of to make multi-label hierarchical classification (HMC). In this work the authors also discuss about two other versions of the Clus proposed before: single-label classification (SC) and hierarchical single-label classification (HSC). For the development of these classifiers the authors used the induction of decision trees firstly supporting only taxonomies represented by a tree and after showed how this model can be modified for use in hierarchical DAG structures. For the induction of decision trees a framework named predictive clustering trees (PCT) (Blockeel et al., 1998) was used. The PCT views a decision tree as a hierarchy of clusters: the top-node corresponds to one cluster containing all data, which is recursively partitioned into smaller clusters while moving down the tree. PCTs are constructed so that each split maximally reduces intra-cluster variance.

In the work of (Silla and Freitas, 2009) the authors proposed a method that is an extension of the flat classification algorithm naive Bayes adapted to hierarchical classification. The author compared the results of a local and two global version of the algorithm using biologic data for protein's functions prediction. The results of the global approach named Global Multi-label Naive Bayes (GMNB) will be compared with the methods proposed in this work. Lately in the work (Silla and Kaestner, 2013) the authors also evaluated the GMNB performance in a different domain to predict bird species with the presence of a taxonomy of species.

The proposal of (Romão and Nievola, 2012) adapted Learning Classifier Systems (LCS) in order to predict protein functions. The proposed approach, called HLCS (Hierarchical Learning Classifier System) builds a global classifier to predict all classes in the application domain and its is expressed as a set of IF-THEN classification rules.

A lot of different techniques were tested in different models and among then stand out: a model named Multi-Label Hierarchical Classification with an Artificial Immune System (MHC-AIS) that generates rules IF-THEN and two main versions were presented exploring both local and global approaches (Alves et al., 2008); the use of optimization based on ant colony to predict classes in problems of hierarchical classification (Otero et al., 2010).

A global method called Grammatical Evolution for Hierarchical Multi-label classification (GEHM) was proposed in (Cerri et al., 2013). The approach makes use of grammatical evolution for generating hierarchical multi-label classification rules. In this approach, the grammatical evolution algorithm evolves the antecedents of classification rules, in order to assign instances from a hierarchical multi-label classification dataset to a probabilistic class vector. The method is compared to bio-inspired algorithms in protein function prediction datasets. The empirical analysis conducted in the work showed that GEHM outperforms the bio-inspired algorithms with statistical significance, suggesting that grammatical evolution is a promising alternative to deal with hierarchical multi-label classification of biological datasets.

The authors of (Barros et al., 2013) developed a hierarchical multi-label classification algorithm for protein function prediction, named Hierarchical Multi-label Classification with Probabilistic Clustering (HMC-PC) that was based on probabilistic clustering making use of cluster membership probabilities in order to generate the predicted class vector. An extensive empirical analysis was performed comparing the proposed approach to four different hierarchical multi-label classification algorithms in protein function datasets structured both as trees and DAG. The presented results showed that HMC-PC achieves superior or comparable results when compared to the state-of-the-art method for hierarchical multi-label classification.

Finally, (Ferrandin et al., 2013) presented an algorithm for hierarchical classification using the global approach, called Hierarchical Multi-label Classifier System using Formal Conceptual Analysis and Similarity of Cosine (HMCS-FCA-SC) for hierarchical multi-label classification. The proposed algorithm combined FCA techniques for hierarchical classification.

For more content about hierarchical classification and related works we suggest the survey proposed by (Silla and Freitas, 2011b) with a large review about hierarchical classification demonstrating the major theories about the subject, classifying the algorithms and proposing a standard nomenclature for classify the works in this field or research.

3.2 Related Works in Centroid-based Classification

The initial adoption of a centroid-based classifier was in information retrieval and text classification with the work of (Rocchio, 1971). A centroid-based classifier when applied to text classification using tf-idf vectors to represent documents is known as the Rocchio classifier.

In (Han and Karypis, 2000) experiments with text categorization showed that the centroid-based classifier outperformed other algorithms such as Naive Bayesian, *k*-nearest-neighbours, and C4.5, on a wide range of datasets. The analysis showed that the similarity measure used by the centroid-based scheme allows it to classify a new document based on how closely its behaviour matches the behaviour of the documents belonging to different classes.

A method to improve the centroid-based classification accuracy was proposed by (Theeramunkong and Lertnattee, 2001) considering a number of statistical term weighting systems based on term-distribution, including factors of intra-class, inter-class, overall term frequency distribution and term length normalization. A number of experiments using drug information web pages and newsgroups data set were done. The results showed that the method outperforms standard tf-idf centroid-based, k-nearest neighbor and naive Bayesian classifiers to some extent.

(Tibshirani et al., 2002) used a centroid-based classifier to cancer class prediction from gene expression profiling. The method called nearest shrunken centroids identifies subsets of genes that best charac-

$$d_{tfidfic} = (tf_1\ log(\frac{N}{df_1})(\frac{tfic_1}{tf_1}), tf_2\ log(\frac{N}{df_2})(\frac{tfic_2}{tf_2}), ..., tf_n\ log(\frac{N}{df_n})(\frac{tfic_n}{tf_n})) \qquad (7)$$

terize each class. The method was highly efficient in finding genes for classifying small round blue cell tumors and leukemias.

The authors of (Enembreck et al., 2006) used a centroid-based approach for identifying people who have the most appropriate competencies to form a research and development team. The selection was done through the analysis of the curriculum vita of the candidate researchers.

For (Tan, 2008), in the context of text categorization, centroid-based classifiers proved to be a simple and yet efficient method but it often suffers from the inductive bias or model misfit incurred by its assumption. In order to address this issue, the author proposed a novel batch-updated approach to enhance the performance of centroid-based classifiers. The main idea behind this method is to take advantage of training errors to successively update the classification model by batch. The technique is simple to implement and flexible to text data. The experimental results indicate that the technique can significantly improve the performance of centroid-based classifiers.

A fast Class-Feature-Centroid (CFC) classifier for multi-class, single-label text categorization in which a centroid is built from two important class distributions: inter-class term index and inner-class term index was proposed by (Guan et al., 2009). CFC proposes a novel combination of these indexes and employs a denormalized cosine measure to calculate the similarity score between a text vector and a centroid. Experiments showed that CFC consistently outperformed the state-of-the-art SVM classifiers is more effective and robust than SVM when data is sparse.

4 PROPOSED ALGORITHM

Considering the centroid-based techniques employed for text classification demonstrated in the Section 2.2 an adaptation was made allow them to deal with hierarchical classification. In a first moment only the tf-idf for weighting the attributes was used and this classifier was named Hierarchical Centroid-Based Classifier System (HCCS) and is showed in Algorithm 1.

Algorithm 1: HCCS.

```
Require: The sets of instances for: training $TR$, testing
    $TE$; The class taxonomy $H$;
 1: Initialize a set of centroids for the classes in $H$;
 2: foreach ($tr_i \in TR$) do
 3:     Represent $tr_i$ attributes as tf-idf vector $trv_i$;
 4:     Add $trv_i$ to the centroid of the class it belongs
            and to the centroides of its descendants
            in $H$;
 5: end for
 6: Compute the average for all centroids;
 7: foreach ($te_i \in TE$) do
 8:     Represent $te_i$ attributes as tf-idf vector $tev_i$;
 9:     Find the centroid most similar to $tev_i$;
10:     Predict the class of the chosen centroid to $te_i$;
11: end for
12: Compute the results of classification process;
```

Every instance in the training partition receives the same treatment of a document in the text classification process so every attribute of the instance is weighted using the tf-idf (line 3) using the Equation 3. To deal with the relationships among the classes, every instance vector was added to the centroid vectors of all classes it belongs to regarding the taxonomy, e.g. the class that is explicitly assigned to the instance and all ancestors in the taxonomy. By this way, the centroid of one parent class will be the average of the centroids of all its children classes (line 4) and the instances directly assigned to this parent class. The average of the centroids (line 6) is computed using Equation 4.

The testing phase of the HCCS consists in to find the most similar centroid for every test instance, the selection (line 9) is done according to Equation 6. Finally the classifier hits and misses are computed (line 12) through the measures hP, hR and hF respectively showed in the Equation 1.

An improved version of HCCS named HCCSic (with ic meaning intra-class) was created adapting tf-idf weighting of the attributes to consider the intra-class attribute frequency. This variant version uses the same steps represented in the Algorithm 1 except for the line 3 where the weighting of attributes was done as shown in Equation 7, where the $tfic_i$ is the frequency of the attribute i in all training instances belonging to the same class of the instance d.

The main intention with the use of the intra-class frequency in the HCCSic is to weight the attributes considering its frequency among the instances of the same class. Summarizing, one attribute that is very frequent among all instances of the same class will have a bigger weight.

5 EXPERIMENTAL EVALUATION

In this Section the experimental evaluation realized with the proposed algorithm is presented along with

Table 1: Results obtained by HCCS and HCCSic compared with HLCS (Romão and Nievola, 2012) and GMNB (Silla and Freitas, 2009) algorithms.

Protein	Signature	HCCS			HCCSic			HLCS			GMNB		
Type	Type	hP	hR	hF	hP	hR	hF	hP	hR	hF	hP	hR	hF
Enzime	Interpro	79.76	83.00	81.35	88.63	92.26	90.55	87.80	85.36	86.56	94.96	89.58	90.53
	Pfam	74.73	78.61	76.62	86.75	90.95	88.80	86.34	81.47	83.83	95.15	86.94	88.72
	Prints	76.41	79.80	78.07	82.87	86.22	84.51	89.69	82.33	85.85	92.21	87.26	87.98
	Prosite	79.17	82.69	80.90	87.31	91.27	89.24	90.35	86.27	88.26	95.14	89.53	90.70
GPCR	Interpro	70.33	75.30	72.71	71.04	74.49	72.72	90.26	74.30	81.51	87.60	71.33	77.01
	Pfam	48.50	55.00	51.54	44.60	51.81	47.93	82.53	60.30	69.69	77.23	57.52	64.40
	Prints	67.14	73.21	70.04	68.27	73.07	70.59	86.50	68.18	76.26	87.06	69.42	75.38
	Prosite	46.86	52.65	49.59	42.14	47.32	44.58	79.42	60.45	68.65	75.64	53.73	61.14

the used datasets. Also, the directly and statistical comparisons of results against other algorithms from the literature is demonstrated.

5.1 Datasets

The two biological databases used in this article are from the family of G-Protein Coupled Receptor (GPCR) and Enzymes. The protein functional classes are given by unique hierarchical indexes by (Horn et al., 2003) in the case of GPCRs, and by Enzyme Commission Codes (Tipton and Boyce, 2000) in the case of enzymes. These databases were used in the works of (Silla and Freitas, 2009) and (Romão and Nievola, 2012), and are available at https://sites.google.com/site/carlossillajr/resources. Enzymes are catalysts that accelerate chemical reactions while GPCRs are proteins involved in signaling and are particularly important in medical applications as it is believed that from 40% to 50% of current medical drugs target GPCR activity (Filmore, 2004).

Each dataset has four different versions based on different kinds of predictor attributes, and in each dataset the classes to be predicted are hierarchical protein functions. Each type of binary predictor attribute indicates whether or not a “protein signature” (or motif) occurs in a protein (Silla and Freitas, 2009). The motifs used in this work were: Interpro Entries, FingerPrints from the Prints database, Prosite Patterns and Pfam. Apart from the presence/absence of several motifs according to the signature method, each protein has two additional attributes: the molecular weight and the sequence length.

Table 2 shows main characteristics of datasets after the pre-processing steps which are detailed in (Silla and Freitas, 2009). In all datasets, each protein (example) is assigned at least to one class at each level of the hierarchy.

Before performing the experiments, the following preprocessing steps were applied to the datasets: (i) Every class with fewer than 10 examples was merged with its parent class. If after this merge the class still had fewer than 10 examples, this process would be repeated recursively until the examples would be labeled to the root class. (ii) All examples whose most specific class was the root class were removed. (iii) A class blind discretization algorithm based on equal-frequency binning (using 20 bins) was applied to the molecular weight and sequence length attributes, which were the only two continuous attributes in each dataset.

Table 2: Enzime and GPCR dataset main characteristics.

Protein Type/Signature		Attributes	Examples	Classes/Level
Enzyme	Interpro	1,216	14,027	6/41/96/187
	Pfam	708	13,987	6/41/96/190
	Prints	382	14,025	6/45/92/208
	Prosite	585	14,041	6/42/89/187
GPCR	Interpro	450	7,444	12/54/82/50
	Pfam	75	7,053	12/52/79/49
	Prints	283	5,404	8/46/76/49
	Prosite	129	6,246	9/50/79/49

The data used in this paper is a subset of protein function datasets and a more detailed description about the dataset used in this work is presented in the work of (Silla and Freitas, 2011a).

5.2 Obtained Results

The experiments were performed using 10-fold cross-validation. Table 1 shows obtained results with the proposed approaches HCCS and HCCSic comparing their results with the algorithms HCLS (Romão and Nievola, 2012) and GMNB (Silla and Freitas, 2009). The results are represented by the three hierarchical measures hP, hR and hF.

A comparison of the results obtained by HCCS and HCCSic is showed in Table 4 in which is possible to see that the HCCSic outperformed the results

Table 3: Results obtained by HCCS and HCCSic compared with HLCS (Romão and Nievola, 2012) and GMNB (Silla and Freitas, 2009) algorithms. The ⊕ indicates the best result.

Protein	Signature	HCCS			HCCSic			HLCS			GMNB		
Type	Type	hP	hR	hF	hP	hR	hF	hP	hR	hF	hP	hR	hF
Enzime	Interpro					⊕	⊕				⊕		
	Pfam					⊕	⊕				⊕		
	Prints										⊕	⊕	⊕
	Prosite					⊕					⊕		⊕
GPCR	Interpro		⊕					⊕		⊕			
	Pfam							⊕	⊕	⊕			
	Prints		⊕							⊕	⊕		
	Prosite							⊕	⊕	⊕			

of HCCS in the most part of the datasets. That results show improvements in this particular classification problem afforded by the modification of tf-idf weighting to the tf-idf with intra-class frequency presented in the Equation 7.

Table 4: Comparison HCC and HCCic obtained results. The ⊕ indicates the best result.

Protein	Signature	HCCS			HCCSic		
Type	Type	hP	hR	hF	hP	hR	hF
Enzime	Interpro				⊕	⊕	⊕
	Pfam				⊕	⊕	⊕
	Prints				⊕	⊕	⊕
	Prosite				⊕	⊕	⊕
GPCR	Interpro		⊕		⊕		⊕
	Pfam	⊕	⊕	⊕			
	Prints		⊕		⊕		⊕
	Prosite	⊕	⊕	⊕			

Table 3 presents the comparison of results between the proposed algorithms against the HLCS (Romão and Nievola, 2012) and GMNB (Silla and Freitas, 2009) algorithms. Considering the distribution of the best results we see that the HCCSic obtained best values for recall and consequently for hF in the enzyme datasets while the GMNB obtained best precision. The HCLS approach outperformed the other algorithms for the GPCR datasets.

There is an apparent difference of performance in the GPCR datasets when comparing the HCCS and HCCSic results against the other algorithms, and except in the Prosite dataset in which all classifiers had a low performance, that the performance drop could be consequence of the number of the classes in the levels of the taxonomy and the number of examples in the dataset, revealing a greater sensitivity of the proposed algorithms to these variables. Looking at Table 2 is possible to see that the GPCR datasets have a bigger number of classes in the first levels of the hierarchy and a lower number of instances.

Statistical tests based on Friedman comparing the hF values between the classifiers with ($\alpha = 0.05$) indicated significant differences between the classifiers. The post-hoc analysis represented in Table 5 demonstrated that there is no significant statistical differences between the results of HCCSci and HCLS or GMNB algorithms. The statistical test also presented significance when comparing HCCS against HCCSic algorithms, showing that the improvement done on the second classifier achieve best results.

Table 5: Results of Friedman test considering the hF measure of classifiers.

	Friedman Test using hF	
Comparison	p-value	significance
HCCS - HCCSic	0.030488	*
HCCS - HLCS	0.000892	***
HCCS - GMNB	0.000482	***
HCCSic - HLCS	0.136864	
HCCSic - GMNB	0.085510	
HLCS - GMNB	0.799078	

6 CONCLUSION

This paper presented a new algorithm for the hierarchical classification problem of predicting protein functions supporting taxonomies organized as a tree. The algorithm is an adaptation of the centroid-based classifiers largely employed in text classification problems and was presented with two versions: the first one - named HCCS - using only tf-idf to weight the attributes and the second - named HCCSic - adding a intra-class frequency weight to tf-idf weighting. Both approaches here proposed are classified as global classifiers that support taxonomies organized as a tree, predict one class per instance in the hierarchy (SPL) and can predict classes at all levels of the taxonomy (NMLNP).

The main advantage of the proposed algorithms is the simplicity of the centroid-based classification that has a cost for training linear to the number of the training instances and the cost for testing linear to the number of testing instances and the number of classes in the taxonomy. In despite of its simplicity the obtained results are very competitive in comparison with other algorithms. Another advantage of the centroid-based approach is that it summarizes the characteristics of each class, using a centroid vector. The advantage of the summarization performed by the centroid vectors is that it combines multiple prevalent features together, even if these features are not simultaneously present in a single instance. This is useful because can capture individual features present only in a few examples. Also, in terms computational time although it's evaluation wasn't the main focus of this work, the centroid-based approaches here proposed showed clearly to require less time and resources than the rules (HLCS) and Naive Bayes (GMND) approaches.

On the other hand, centroid-based classifiers are dependent of a good set of examples for each class and can lead to wrong classifications if the partitioning of examples is unbalanced. Also, in the context of hierarchical classification, the addition of children data to train the centroids of the higher classes of the hierarchy needs to be more investigated because the average of the vectors from two children classes can not always truly represent the characteristics of the parent class. In a centroid-based approach it's important to ensure that the instances belonging to the same class will be proportionally distributed between the training and testing partitions, if all examples of one class remain in the same partition the centroid of this class wont be trained or wont have examples to classify.

As future researches we highlight a deeper analysis of the centroid relations between parent and children classes in the hierarchy using different datasets. Also this algorithm can be improved to support DAG taxonomies and to make multiple paths of label prediction (MPL). Another approach to be investigated is the selection of a set of k centroids for every instance being classified, the final centroid that would predict the class to the instance will be select by election in a similar way used in k-NN algorithm.

REFERENCES

Alves, R. T., Delgado, M. R., and Freitas, A. A. (2008). Multi-label hierarchical classification of protein functions with artificial immune systems. In *Proceedings of the 3rd Brazilian symposium on Bioinformatics: Advances in Bioinformatics and Computational Biology*, BSB '08, pages 1–12, Berlin, Heidelberg. Springer-Verlag.

Barros, R. C., Cerri, R., Freitas, A. A., and de Carvalho, A. C. P. L. F. (2013). Probabilistic clustering for hierarchical multi-label classification of protein functions. In *In proceeding of: Machine Learning and Knowledge Discovery in Databases (ECML 2013), At Prague, Czech Republic*, volume 8189 of *Lecture Notes in Computer Science*.

Blockeel, H., De Raedt, L., and Ramon, J. (1998). Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning*, pages 55–63. Morgan Kaufmann.

Cerri, R., Barros, R. C., de Carvalho, A. C. P. L. F., and Freitas, A. A. (2013). A grammatical evolution algorithm for generation of hierarchical multi-label classification rules. In *IEEE Congress on Evolutionary Computation*, pages 454–461. IEEE.

Enembreck, F., Scalabrin, E. E., Tacla, C. A., and Ávila, B. C. (2006). Automatic identification of teams based on textual information retrieval. In *CSCWD*, pages 534–538. IEEE.

Ferrandin, M., Nievola, J. C., Enembreck, F., Scalabrin, E. E., Kredens, K. V., , and Ávila, B. C. (2013). Hierarchical classification using fca and the cosine similarity function. In *Proceedings of the 2013 Internacional Conference on Artificial Inteligence (ICAI'13)*, volume 1, pages 281–287.

Filmore, D. (2004). It's a GPCR world. *Modern Drug Discovery*, 7:24–28.

Guan, H., Zhou, J., and Guo, M. (2009). A class-feature-centroid classifier for text categorization. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 201–210, New York, NY, USA. ACM.

Han, E.-H. and Karypis, G. (2000). Centroid-based document classification: Analysis and experimental results. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 424–431, London, UK, UK. Springer-Verlag.

Horn, F., Bettler, E., Oliveira, L., Campagne, F., Cohen, F. E., and Vriend, G. (2003). Gpcrdb information system for g protein-coupled receptors. *Nucleic Acids Research*, 31(1):294–297.

Kiritchenko, S., Matwin, S., and Famili, A. F. (2005). Functional annotation of genes using hierarchical text categorization. In *in Proc. of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology (held at ISMB-05*.

Otero, F. E. B., Freitas, A. A., and Johnson, C. G. (2010). A hierarchical multi-label classification ant colony algorithm for protein function prediction. *Memetic Computing*, pages 165–181.

Rocchio, J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System - Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall.

Romão, L. M. and Nievola, J. C. (2012). Hierarchical classification of gene ontology with learning classifier systems. In *Advances in Artificial Intelligence - IBERAMIA 2012*, volume 7637 of *Lecture Notes in*

Computer Science, pages 120–129. Springer Berlin Heidelberg.

Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Silla, C. N. and Freitas, A. A. (2009). A global-model naive bayes approach to the hierarchical prediction of protein functions. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 992–997, Washington, DC, USA. IEEE Computer Society.

Silla, C. N. and Freitas, A. A. (2011a). Selecting different protein representations and classification algorithms in hierarchical protein function prediction. *Intell. Data Anal.*, 15(6):979–999.

Silla, C. N. and Kaestner, C. A. A. (2013). Hierarchical classification of bird species using their audio recorded songs. In *SMC*, pages 1895–1900. IEEE.

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

Tan, S. (2008). An improved centroid classifier for text categorization. *Expert Syst. Appl.*, 35(1-2):279–285.

Theeramunkong, T. and Lertnattee, V. (2001). Improving centroid-based text classification using term-distribution-based weighting system and clustering.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572.

Tipton, K. F. and Boyce, S. (2000). History of the enzyme nomenclature system. *Bioinformatics*, 16(1):34–40.

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