# Exploring Attribute Selection in Hierarchical Classification

Bruno C. Paes[1], Alexandre Plastino[1], Alex A. Freitas[2]

[1] Universidade Federal Fluminense, Brazil
{bpaes, plastino}@ic.uff.br
[2] University of Kent, United Kingdom
a.a.freitas@kent.ac.uk

**Abstract.** In the domain of many classification problems, classes have relations of dependency that are represented in hierarchical structures. These problems are known as hierarchical classification problems. Methods based on different approaches, considering hierarchical relations in different ways, have been proposed to solve them, in the attempt to achieve better predictive performance. In this work, we explore attribute selection techniques in conjunction with hierarchical classifiers from different categories, with the goal of improving their respective performances. Computational experiments, made with 18 hierarchical datasets, have indicated that the adopted classifiers attain better predictive accuracy when the most relevant attributes are considered in their construction.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Data Mining

Keywords: attribute selection, classification, data mining, hierarchical classification

## 1. INTRODUCTION

The task of classification aims at estimating the class of a new element from its characteristics. In most classification problems, known as flat classification problems, classes have no descendent relation among themselves. However, there are several problems in which classes have relations of dependency, that are represented in hierarchical structures, known as hierarchical classification problems. The methods for hierarchical classification should be able to consider the hierarchical organization of the classes, with the goal of obtaining a higher predictive capacity.

Examples of problems that have their classes structured in a hierarchical manner can be found in different areas of application. The domain of Bioinformatics has important works aimed at the classification of proteins and enzymes into functional classes, which are found hierarchically organized [Costa et al. 2008; Holden and Freitas 2007; 2009]. In the area of document classification, there are texts that can be characterized considering a hierarchical structure of subjects [Dumais and Chen 2000; Sun and Lim 2001]. In image recognition applications, objects can be categorized in geometrical forms that hold descendent relations [Barutcuoglu and DeCoro 2006].

Attribute selection is a technique widely used in data mining, especially in the classification task [Guyon and Elisseeff 2006]. In this context, its goal is to identify relevant attributes, aiming at gaining one or more of the following benefits: time reduction in the classification process, improvement of the predictive capacity, and achievement of a more compact representation of the concept to be learned.

In this work, we explore the use of attribute selection techniques, aiming at boosting the performance of hierarchical classifiers. Two different techniques for hierarchical classification will be considered: the first one, a traditional hierarchical strategy named *Per Parent Top Down* (PPTD), based on the paradigm of hierarchical classification "local per parent node", and the second one, named *Sum of*

This work was supported by CNPq and FAPERJ research grants.

*Weighted Votes* (SWV), and characterized as a "local per level" strategy, recently proposed by Paes et al. [2012], whose performance has been shown to be competitive when compared to that of other strategies from different paradigms of hierarchical classification.

Traditionally, attribute selection techniques are applied in a pre-processing stage. Attributes not selected are not taken in the training of the classifier nor in the classification of a new instance. In the work of Pereira et al. [2011], a new attribute selection method was proposed, named *lazy*, whose main goal is to postpone the selection of the attributes to the moment of classification of a new instance. The basic idea is to consider the values of the attributes in the instance to be classified and then choose the attributes that will be part of the classification. This way, the attributes selected will be specific for each instance, which may increase, as a result, the predictive capability of the classifier. In this work, apart from the traditional techniques, the *lazy* methods for attribute selection will also be explored.

The remainder of this work is organized as follows. Section 2 describes the hierarchical classifiers that are explored. Section 3 provides the concepts related to attribute selection and the definition of the attribute selection technique incorporated into the hierarchical classifiers. Section 4 describes the computational experiments undertaken, and Section 5 evaluates the results. Section 6 explores a *lazy* technique for attribute selection. Section 7 offers the conclusions of the work.

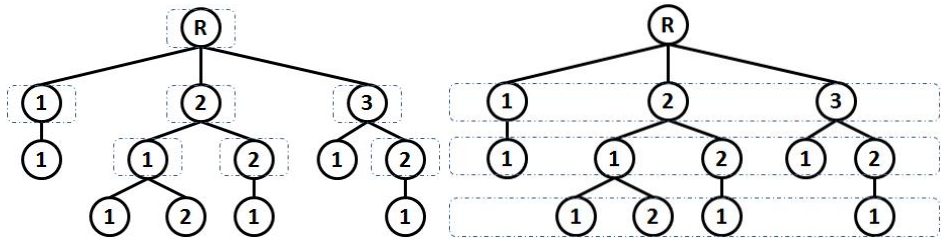
## 2. HIERARCHICAL CLASSIFICATION

The algorithms for hierarchical classification are organized in different categories [Silla and Freitas 2011]. Each one is different as regards the manner in which the hierarchical structure is explored, whether in the simplification of the hierarchy (flat classification approach), in the use of a set of traditional flat classifiers (local classification approach), or in the construction of a single classifier that takes all the class hierarchy into account (global classification approach).

The local classification approach is the most commonly explored, and considers the class hierarchy through a local perspective, with the combination of classifiers that consider, in an isolated manner, different parts of the hierarchy. In [Silla and Freitas 2011], the local classifiers are categorized according to the manner in which this local information is explored: *local classifier per node*, *local classifier per parent node* and the *local classifier per level* approaches.

In this work, two classifiers will be explored: *Per Parent Top Down* (PPTD) and *Sum of Weighted Votes* (SWV). The PPTD hierarchical classifier is based on the concepts of the "local per parent node" approach. In this approach the training of a flat classifier is carried out for each non-leaf class (internal node), as shown in Figure 1(a). In each flat classifier, represented by a dotted rectangle, only the child classes of the parent class are considered (only instances labeled with child classes are considered). This way, it is possible to obtain a hierarchy of flat classifiers. The classification of a new instance is done in a *top-down* fashion. At first, the instance is evaluated by the root node classifier, which chooses one amongst its child classes (e.g., class 2 in Figure 1(a)). The process goes on to the first level and the node classifier associated with the resulting class picks one amongst its child classes (e.g., class 2.1) and, this way, successively, until getting to a leaf class (e.g., class 2.1.2).

The SWV hierarchical classifier is considered as a "local per level" strategy. In this approach, a flat classifier per hierarchy level is trained, as shown in Figure 1(b). For each flat classifier, only the classes of the level at hand are taken into account (only instances labeled with theses classes are considered). To run the classification of a new instance, each classifier generated is executed to produce a class for each level (e.g., classes: 2, 2.1, and 2.1.2). However, one issue that has to be solved in local per level classifiers is the inconsistent set of classes obtained by the different classifiers associated with the different levels (e.g., classes: 2, 3.2, and 2.1.2). The SWV strategy, proposed by Paes et al. [2012], deals with this question, privileging the branch of the hierarchy that presents the largest number of classes estimated, named votes. In this strategy, the sum of the number of votes is weighted with the




(a) PPTD (local per parent node approach) (b) SWV (local per level approach)

Fig. 1. Hierarchical Classifiers

probabilities estimated by the flat classifiers when obtaining the classes for the different levels.

## 3. ATTRIBUTE SELECTION

This section presents concepts related to the process of attribute selection and defines how the incorporation of the attribute selection process will be done in the PPTD and SWV hierarchical classifiers, with the goal of improving the respective predictive accuracies.

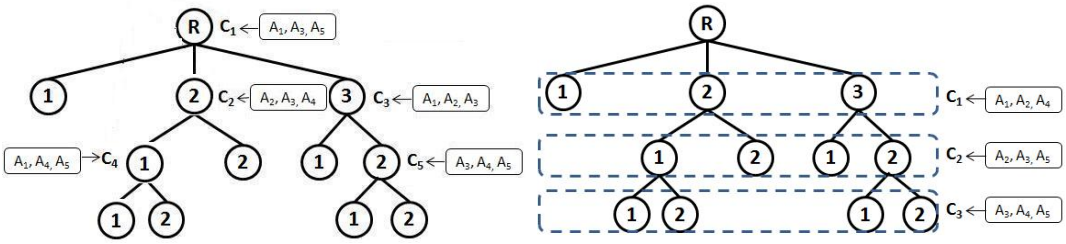
Attribute selection is a technique widely used in data mining, especially in the task of classification [Guyon and Elisseeff 2006]. In this realm, its goal is to identify relevant attributes, aiming at obtaining one or more of the following benefits: (a) reduction in the execution time for the classification process as, with less attributes assessed, the classification process tends to be run in a shorter processing time; (b) increase of the predictive capability of the classifier as the selection of the attributes seeks to remove redundant or irrelevant attributes from the dataset, allowing the generation of a classifier that is less prone to error; and (c) the generation of a more compact representation of the concept to be learned, as the knowledge will lie concentrated only in the attributes that are really important for the classification.

In general terms, the methods for attribute selection can be categorized into three large types. *Wrapper* methods evaluate the quality of the subsets of attributes using their own adopted classification algorithm. They usually have good predictive capability as they evaluate each subset of attributes using the same classification algorithm that will be used in the classification process. They require, however, several executions of the classification algorithm, which raises the computational cost in comparison with other methods.

*Filter* methods are independent from the classification algorithm that will be applied. They use specific measures to evaluate the quality of the attributes available. These methods can evaluate each attribute independently from the others, determining the degree of correlation that exists between each attribute and the class [Yang and Pedersen 1997] or can assess subsets of attributes, seeking through heuristic strategies the set that best identify the classes [Hall 2000; Liu and Setiono 1996]. In this work, *Filter* type methods will be used in conjunction with the hierarchical classifiers.

*Embedded* methods are incorporated into the classification algorithm. They are applied internally and in an integrated manner to the classification method. Algorithms to induce decision trees are typical examples as they internally select the attributes that will label the nodes of the tree generated.

Some examples of the use of attribute selection can be found in the area of hierarchical classification in specific datasets and domains. In [Koller and Sahami 1997], a *top-down* hierarchical document classifier is implemented in which the attributes are selected prior to the training of the classifier for each node of the hierarchy. Secker et al. [2010] proposed a hierarchical *top-down* classifier with attribute selection for a problem in the Bioinformatics domain. In that work the nodes of the hierarchy can be associated with different types of flat classifiers. The hierarchy of classifiers is defined by a selective method that identifies the most adequate classification algorithm for each node. The attribute selection is used to reduce the dimensional aspect of the data and improve predictive accuracy.




(a) In the PPTD strategy (b) In the SWV strategy

Fig. 2. Attribute Selection

The main contribution of this work is the application of attribute selection strategies integrated with general purpose hierarchical classifiers, i.e., not restricted to specific datasets or domains, with the goal of improving the predictive performance of the classifiers. Two hierarchical classifiers will be explored: the traditional *Per Parent Top-Down* (PPTD) and the hierarchical strategy that was recently proposed by Paes et al. [2012], named *Sum of Weighted Votes* (SWV).

The method applied for attribute selection produces a ranking of the attributes, considering the information gain [Han and Kamber 2011] measure, based on the concept of entropy and, after that, returns the $n\%$ best attributes, where $n$ is an entry parameter. It is a well-known method of the *Filter* type with large applicability in the domain of flat classification.

In this first approach explored, the attribute selection is carried out in a pre-processing stage, prior to the training of the flat classifiers which form the hierarchical classifiers. Section 6 explores the *lazy* selection approach which is executed only at the time of classifying a new instance.

Figure 2 shows the application of the attribute selection method in the PPTD and SWV hierarchical classifiers. In order to illustrate, we consider the original set of attributes $A_1, A_2, A_3, A_4$, and $A_5$, and that 60% of the attributes should be selected via the attribute selection method.

Figure 2(a) shows the application of attribute selection in the PPTD hierarchical classifier. For each parent node of the hierarchy, a set of attributes is selected prior to carrying out the training of the classifier. This way, different subsets of attributes are selected in each node. All the attributes are available in each node for the execution of attribute selection, i.e., the attributes are not lost by the child nodes when not selected by the parent node. The figure shows the result of the attribute selection applied to each parent node. From these different attribute subsets, the classifiers $C_1, C_2, C_3$, $C_4$, and $C_5$ are trained and associated with each parent node. For example, classifier $C_4$, associated with the node represented by class 2.1, is trained with attributes $A_1, A_4$, and $A_5$.

Figure 2(b) shows the application of attribute selection in the SWV hierarchical classifier. For each level of the hierarchy, a subset of attributes is selected, prior to the training of the respective flat classifier. The figure shows the attributes selected on each level of the hierarchy and, from them, the training of the classifiers $C_1, C_2$, and $C_3$, for the different levels, is done. For example, classifier $C_2$, associated with level 2 of the hierarchy, is trained with attributes $A_2, A_3$, and $A_5$

## 4. DESCRIPTION OF THE EXPERIMENTS

To evaluate the performance of the hierarchical classifiers with the application of attribute selection, 18 datasets were used, described below in two large groups. Group A consists of eight datasets, containing information on protein functions. These datasets are split into two subgroups: GPCR (*G-Protein-Coupled Receptor*) and EC (*Enzyme Commission*). The GPCR group consists of four datasets (GPCRpfam, GPCRprints, GPCRprosite, and GPCRinterpro). GPCRs are proteins that

Table I. Characteristics of the datasets

| Group | Dataset | #Classes | #Instances | Group | Dataset | #Classes | #Instances |
|---|---|---|---|---|---|---|---|
| A | **GPCRpfam** | 12/52/79/49 | 6524 | B | **Church** | 4/18/36/27 | 1677 |
| | **GPCRprints** | 8/46/76/49 | 4880 | | **CellCycle** | 4/17/34/23 | 1711 |
| | **GPCRprosite** | 9/50/79/49 | 5728 | | **Derisi** | 4/18/35/25 | 1661 |
| | **GPCRinterpro** | 12/54/82/50 | 6935 | | **Eisen** | 4/15/29/17 | 1163 |
| | **ECpfam** | 6/41/96/190 | 11057 | | **Expr** | 4/17/34/25 | 1688 |
| | **ECprints** | 6/45/92/208 | 11048 | | **Gasch1** | 4/17/34/25 | 1660 |
| | **ECprosite** | 6/42/89/187 | 11328 | | **Gasch2** | 4/17/33/25 | 1678 |
| | **ECinterpro** | 6/41/96/187 | 11101 | | **Phenotype** | 4/12/21/13 | 621 |
| | | | | | **Sequence** | 4/17/32/24 | 1680 |
| | | | | | **SPO** | 4/17/34/25 | 1649 |

relay signals from the external environment to inside the cell. The EC group consists of four datasets (ECpfam, ECprints, ECprosite, and ECinterpro), that represent enzyme functions. The datasets of the GPCR and EC groups have been used in several works that deal with hierarchical classification problems [Costa et al. 2008; Silla and Freitas 2011]. For the experiments carried out in this work a pre-processing procedure was carried out to remove all the instances (from each dataset), whose more specific class was not associated with a leaf node.

Group B consists of ten datasets that hold genic information. The datasets of this group come from the field of functional genomics and are related to the *Saccharomyces cerevisiae* fungus or to the *Yeast* fungus and are presented by Clare and King [2003]. They are originally multi-label and, for their use in this work (where it is considered that the instances are single-label), they were converted through the random choice of one of the classes associated with each instance.

All data sets have the class hierarchy represented by a non-complete tree structure with four levels. Apart from that, the more specific classes of the instances are associated only with leaf nodes of the class hierarchy. The characteristics of the datasets, shown in Table I, are: the group to which the dataset belongs (Group), the name of the dataset (Dataset), the number of classes for each hierarchy level (#Classes), and the total of instances for each dataset (#Instances).

All the hierarchical classifiers were implemented using the *JAVA* programming language, incorporating algorithms and functions of the data mining tool WEKA 3.7.0 (*Waikato Environment for Knowledge Analysis*) [Witten and Frank 2011]. Two traditional flat classifiers were used: one of the *eager* type, C4.5, and another of the *lazy* kind, k-NN. In order to represent these flat classifiers in the experiments, the versions provided in the WEKA tool, named, respectively, J48 e Ibk, were adopted. The *Filter* attribute selection method, provided in the WEKA tool with the name *InfoGainAttributeEval* was applied in the implemented hierarchical classifiers. The choice was based on its simplicity and on the fact that it is a widely known method. It should be pointed that this method has the number of attributes to be selected as its entry parameter.

The evaluation of the hierarchical classifiers was performed using 10-fold cross-validation. The *hierarchical f-measure (hF)* measure was adopted – as presented by Kiritchenko et al. [2005]. The hF measure is calculated as the harmonic mean of measures *hierarchical precision* ($hP$) and *hierarchical recall* ($hR$): $hF = 2 * hP * hR/(hP + hR)$. Where $hP$ is the result of the division between the sum (for all instances) of the number of common classes between the sets of predicted and true classes of each instance and the sum (for all instances) of the number of predicted classes for each instance, and $hR$ is the result of the division between the sum (for all instances) of the number of common classes between the sets of predicted and true classes of each instance, and the sum (for all instances) of the number of true classes of each instance.

In order to evaluate the statistical significance in the comparison between two averages, obtained by two distinct classifiers using 10-fold cross-validation, we used the two-tailed and paired version of the Student's t-test [Jain 1991], with a 95% level of confidence, that is, a p-value equal to 5%.

Table II. hF values obtained by PPTD classifier with and without attribute selection

| Datasets | 1-NN | | | | 7-NN | | | | 9-NN | | | | C4.5 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sel. | | | w/o Sel. | Sel. | | | w/o Sel. | Sel. | | | w/o Sel. | Sel. | | | w/o Sel. |
| **GPCRpfam** | **70.32** | (90) | - | **70.32** | **69.09** | (70) | - | 69.04 | **68.55** | (70) | - | 68.47 | **68.85** | (70) | - | 68.84 |
| **GPCRprints** | **82,97** | (80) | - | **82.97** | **80.95** | (70) | - | 80.89 | **80.41** | (80) | - | **80.41** | **79.22** | (50) | - | 79.19 |
| **GPCRprosite** | **69.26** | (70) | - | 69.25 | **67.38** | (70) | • | 67.31 | **66,57** | (90) | - | **66.57** | **67.67** | (70) | - | 67.63 |
| **GPCRinterpro** | **83,09** | (90) | - | **83.09** | **81.96** | (80) | - | 81.95 | **81.29** | (80) | - | **81.29** | 81.52 | (80) | - | **81.54** |
| **ECpfam** | **98.77** | (70) | - | **98.77** | **98.16** | (70) | - | **98.16** | **97.88** | (70) | - | **97.88** | **98.40** | (60) | - | 98.39 |
| **ECprints** | **98.19** | (80) | - | **98.19** | **97.37** | (80) | - | **97.37** | **97.05** | (80) | - | **97.05** | 97.34 | (80) | - | **97.35** |
| **ECprosite** | **98.81** | (70) | - | 98.80 | **98.29** | (70) | - | **98.29** | **98.03** | (80) | - | **98.03** | **98.46** | (70) | - | **98.46** |
| **ECinterpro** | **99.07** | (30) | - | **99.07** | **98.62** | (70) | - | **98.62** | **98.32** | (70) | - | **98.32** | 98.68 | (70) | • | **98.73** |
| **Total A** | **8** | | | 6 | **8** | | | 4 | **8** | | | 7 | **5** | | | 4 |
| **Church** | **21.64** | (10) | • | 19.38 | **23.06** | (10) | • | 19.66 | **23.07** | (10) | • | 19.82 | **25.29** | (10) | • | 21.53 |
| **CellCycle** | **24.75** | (40) | - | 24.56 | **29.29** | (50) | - | 28.38 | **29.91** | (40) | - | 29.40 | **22.93** | (80) | - | 22.19 |
| **Derise** | **20.09** | (70) | - | 18.89 | **22.07** | (70) | - | 20.21 | **22.71** | (30) | - | 20.89 | **22.82** | (10) | - | 20.47 |
| **Eisen** | **25.66** | (50) | - | 24.70 | **29.53** | (70) | - | 29.25 | **30.39** | (50) | - | 29.16 | **26.77** | (50) | - | 24.24 |
| **Expr** | **26.05** | (10) | - | 25.35 | **29.33** | (70) | - | 27.21 | **29.78** | (70) | - | 28.29 | **26.26** | (30) | - | 24.74 |
| **Gash1** | 27.81 | (70) | - | **28.29** | **31.37** | (80) | - | 30.97 | 27.92 | (70) | - | **30.08** | **24.34** | (90) | • | 22.90 |
| **Gash2** | 24.96 | (80) | - | **25.23** | **27.89** | (80) | - | 26.12 | **27.92** | (70) | • | 26.00 | **23.39** | (40) | - | 22.52 |
| **Phenotype** | **21.34** | (20) | - | 20.27 | **23.23** | (20) | - | 22.52 | **22.95** | (90) | - | **22.95** | **21.57** | (60) | - | 21.39 |
| **Sequence** | **25.01** | (90) | - | 24.02 | **24.88** | (80) | - | 23.73 | **25.63** | (30) | • | 23.90 | **25.26** | (10) | - | 22.86 |
| **SPO** | **21.70** | (20) | - | 18.86 | **24.55** | (30) | • | 21.10 | **26.05** | (30) | • | 22.29 | **23.83** | (10) | • | 19.36 |
| **Total B** | **8** | | | 2 | **10** | | | 0 | **9** | | | 2 | **10** | | | 0 |

## 5. COMPUTATIONAL RESULTS

This section provides the results and analyses of the computational experiments. The goal is to evaluate the impact of attribute selection when applied to the *Per Parent Top-Down* (PPTD) and *Sum of Weighted Votes* (SWV) hierarchical classifiers.

Tables II and III present, respectively, the evaluations of the PPTD and SWV hierarchical classifiers when executed with and without the application of attribute selection. We used each one of the four flat classifiers that obtained the best performance in the experiments carried out by Paes et al. [2012]: 1-NN, 7-NN, 9-NN and C4.5. For each combination of dataset and flat classifier adopted, the hF values obtained by the hierarchical classifier, with attribute selection (Sel.) and without attribute selection (w/o Sel.), are presented. Next to the hF value for the classifiers with attribute selection, it is presented the percentage of the attributes (10%, 20%, ..., 80% or 90%) that led the classifier to reach the best result. If two or more percentage values have generated the best result, the smaller percentage value will be reported. The best results for each flat classifier applied are provided in bold whereas the best results per dataset are underlined. The (•) symbol between the two hF values indicates that the difference between these averages holds statistical significance. The (-) symbol indicates that no statistical significance was observed. Finally, under each dataset group a line of totals is provided that presents the number of times one of the hierarchical classifiers presented an hF value higher than or equal to the hF value of the other, for each flat classifier adopted.

Table II shows the results for the PPTD classifier with and without attribute selection. It is possible to see, on the line of totals that, for the datasets of both groups and for all the flat classifiers used, that the PPTD hierarchical classifier had a higher number of better hF values when the attribute selection was applied. Out of the 12 results obtained that had statistical significance, the PPTD classifier with attribute selection had 11 and the PPTD classifier with no attribute selection got only one. Considering the best results found per dataset (underlined), the PPTD classifier with attribute selection found 18, whereas the PPTD classifier with no attribute selection got six of these results.

Table III shows the results for the SWV classifier, with and without attribute selection. It is possible to observe that in the lines of totals, for the datasets of the two groups and for all flat classifiers, the superior performance of the SWV strategy with attribute selection. All the 20 results with statistical significance were obtained by the SWV strategy with attribute selection. Considering the best results found, per dataset (underlined), the SWV strategy with attribute selection found 18, whereas the SWV strategy with no attribute selection got six of these results.

Table III. hF values obtained by SWV classifier with and without attribute selection

| Datasets | 1-NN | | | | 7-NN | | | | 9-NN | | | | C4.5 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sel. | | | w/o Sel. | Sel. | | | w/o Sel. | Sel. | | | w/o Sel. | Sel. | | | w/o Sel. |
| **GPCRpfam** | **70.31** | (90) | - | **70.31** | **68.78** | (50) | - | 68.68 | **68.25** | (40) | • | 68.07 | **68.76** | (50) | - | 68.70 |
| **GPCRprints** | **83.00** | (80) | - | **83.00** | **80.97** | (70) | - | 80.86 | **80.29** | (70) | - | 80.28 | **79.53** | (50) | - | 79.33 |
| **GPCRprosite** | **69.36** | (40) | - | 69.26 | **67.13** | (40) | • | 66.83 | **66.14** | (30) | - | 66.00 | **67.13** | (20) | - | 67.08 |
| **GPCRinterpro** | **83.09** | (90) | - | **83.09** | **81.69** | (80) | - | 81.68 | **81.31** | (80) | - | **81.31** | **81.98** | (90) | - | 81.80 |
| **ECpfam** | **98.77** | (70) | - | **98.77** | **98.30** | (70) | - | **98.30** | **98.19** | (70) | - | **98.19** | **98.43** | (60) | - | **98.43** |
| **ECprints** | **98.19** | (80) | - | **98.19** | **97.45** | (80) | - | **97.45** | **97.22** | (70) | - | **97.22** | **97.54** | (80) | • | 97.51 |
| **ECprosite** | **98.81** | (70) | - | 98.80 | **98.35** | (60) | - | 98.33 | **98.10** | (70) | - | 98.08 | **98.57** | (70) | - | **98.57** |
| **ECinterpro** | **99.08** | (70) | - | **99.08** | **98.82** | (70) | - | **98.82** | **98.70** | (70) | - | **98.70** | 98.78 | (70) | - | **98.79** |
| **Total A** | **8** | | | 6 | **8** | | | 3 | **8** | | | 4 | **7** | | | 3 |
| **Church** | **21.88** | (10) | • | 19.70 | **22.96** | (10) | • | 20.19 | **23.22** | (10) | • | 20.66 | **25.58** | (10) | • | 21.29 |
| **CellCycle** | **25.12** | (40) | - | 24.82 | **30.16** | (40) | - | 28.60 | **30.96** | (40) | - | 29.87 | **24.86** | (40) | - | 24.83 |
| **Derise** | **20.26** | (60) | • | 18.73 | **23.22** | (10) | - | 21.16 | **23.52** | (40) | • | 21.20 | **22.44** | (10) | - | 21.78 |
| **Eisen** | **27.26** | (50) | • | 24.54 | **32.43** | (50) | • | 29.56 | **32.51** | (50) | - | 30.77 | **28.20** | (70) | • | 25.79 |
| **Expr** | **26.18** | (30) | - | 25.62 | **29.85** | (50) | • | 28.12 | **30.42** | (40) | - | 29.26 | **28.92** | (60) | - | 26.61 |
| **Gash1** | **29.60** | (90) | - | 28.98 | **31.67** | (80) | - | 31.24 | 28.93 | (40) | - | **30.80** | **27.30** | (90) | - | 25.86 |
| **Gash2** | **26.46** | (50) | - | 25.03 | **28.39** | (50) | • | 25.55 | **28.93** | (40) | • | 26.21 | **24.62** | (30) | - | 23.39 |
| **Phenotype** | **26.25** | (10) | • | 22.46 | **26.69** | (10) | - | 24.42 | **26.46** | (10) | - | 25.45 | **27.94** | (10) | - | 26.37 |
| **Sequence** | **23.99** | (70) | - | 23.08 | **24.19** | (40) | - | 22.76 | **25.48** | (40) | - | 23.75 | **26.18** | (70) | - | 25.71 |
| **SPO** | **21.95** | (20) | • | 18.50 | **24.98** | (30) | • | 21.59 | **25.64** | (20) | • | 22.31 | **23.38** | (40) | • | 21.50 |
| **Total B** | **10** | | | 0 | **10** | | | 0 | **9** | | | 1 | **10** | | | 0 |

Table IV. Best results found per dataset

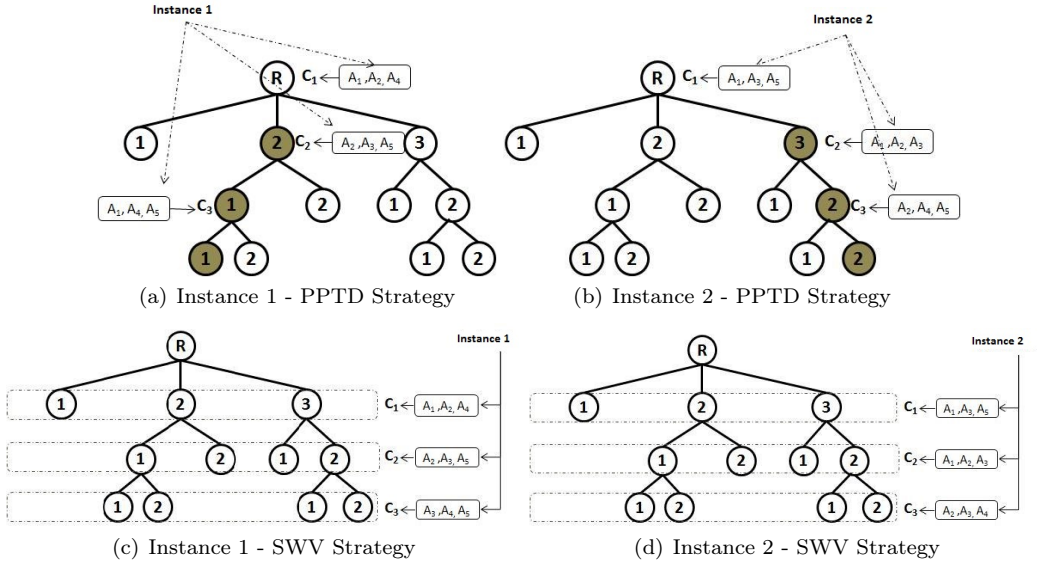
| Group | Datasets | hF | Strategy(ies) | Group | Datasets | hF | Strategy(ies) |
|---|---|---|---|---|---|---|---|
| **A** | **GPCRpfam** | 70.32 | PPTD/1-NN(90) | **B** | **Church** | 25.58 | SWV/C4.5(10) |
| | **GPCRprints** | 83.00 | SWV/1-NN(80) | | **CellCycle** | 30.96 | SWV/9-NN(40) |
| | **GPCRprosite** | 69.36 | SWV/1-NN(40) | | **Derisi** | 23.52 | SWV/9-NN(40) |
| | **GPCRinterpro** | 83.09 | PPTD/1-NN(90) e SWV/1-NN(90) | | **Eisen** | 32.51 | SWV/9-NN(50) |
| | **ECpfam** | 98.77 | PPTD/1-NN(70) e SWV/1-NN(70) | | **Expr** | 30.42 | SWV/9-NN(40) |
| | **ECprints** | 98.19 | PPTD/1-NN(80) e SWV/1-NN(80) | | **Gasch1** | 31.67 | SWV/7-NN(80) |
| | **ECprosite** | 98.81 | PPTD/1-NN(70) e SWV/1-NN(70) | | **Gasch2** | 28.93 | SWV/9-NN(40) |
| | **ECinterpro** | 99.08 | SWV/1-NN(70) | | **Phenotype** | 27.94 | SWV/C4.5(10) |
| | | | | | **Sequence** | 26.18 | SWV/C4.5(70) |
| | | | | | **SPO** | 26.05 | PPTD/9-NN(30) |

Table IV shows, for each dataset, the best result obtained and the strategies that reached them. The strategy is represented by the hierarchical classifier applied, flat classifier used, and the percentage of attributes that were selected. It is possible to see that, for all the 18 datasets, the best result was obtained by a hierarchical strategy with attribute selection. In no case all the attributes (100%) were used. This behaviour points to the importance, also in the hierarchical context, of the use of attribute selection techniques.

When comparing both PPTD and SWV strategies, it is possible to see a better performance in the SWV strategy. In the analysis made by Paes et al. [2012], with no attribute selection, this strategy also produced a performance superior than that of the PPTD strategy. Considering the eight Group A datasets, there was just a slight performance superiority in the SWV strategy, which attained seven best results, while the PPTD strategy got five times the best value for hF. However, for the Group B datasets, there is a clear performance superiority in the SWV strategy that produced nine best results against only one of the PPTD strategy.

## 6. *LAZY* ATTRIBUTE SELECTION

This section presents another important contribution of this work: the use, in the context of hierarchical classification, of a paradigm – recently proposed by Pereira et al. [2011] – to carry out attribute selection, named *lazy* attribute selection. This method executes attribute selection at the classification time of each instance. It is based on the hypothesis that the selection of the attributes can be more efficient if it considers the values of the attributes in the instance to be classified. This way, different subsets of attributes are selected for different instances, considering their specifics. The traditional attribute selection method, used in the previous sections will be named in this section as being of the




(a) Instance 1 - PPTD Strategy (b) Instance 2 - PPTD Strategy

(c) Instance 1 - SWV Strategy (d) Instance 2 - SWV Strategy

Fig. 3. *Lazy* attribute selection in hierarchical classifiers

*eager* type, as the selection of the attributes is done only one time, prior to the classification stage.

The *lazy* method uses an adaptation of the concept of entropy to individually assess each attribute value in the instance to be classified, regarding the quality to differentiate the classes. This strategy is also of the *Filter* type and receives as an entry parameter the number of attributes to be selected.

The *lazy* attribute selection method will be applied in conjunction with the *Per Parent Top-Down* (PPTD) and *Sum of Weighted Votes* (SWV) hierarchical classifiers, both using the k-NN as the flat classifier. Figures 3 illustrate the application of the *lazy* attribute selection in the PPTD and SWV hierarchical classifiers. We considered the original set of attributes $A_1, A_2, A_3, A_4$, and $A_5$, and that 60% of the attributes have to be selected via the attribute selection method.

Figure 3 (a) shows that, for each parent node of the hierarchy, a set of attributes is selected at the time of classification of Instance 1, considering the values of their attributes. At first, attributes $A_1, A_2$, and $A_4$ are selected for classifier $C_1$. After that, classifier $C_1$ predicts class 2. Again, considering the values of attributes of Instance 1, attributes $A_2, A_3$, and $A_5$ are selected for classifier $C_2$, associated with class 2. Then the classifier associated with the class 2 node is executed, producing class 2.1. Thus, attributes $A_1, A_4$, and $A_5$ are selected and used in classifier $C_3$, associated with class 2.1. Finally, classifier $C_3$ predicts class 2.1.1. Similarly, Figure 3 (b) presents the result of the *lazy* attribute selection, considering the values of the attributes in Instance 2, for each parent node in the hierarchy. In Figure 3 (c), for each level of the hierarchy, the attribute selection is performed at the time Instance 1 is submitted to the classification, selecting different attribute subsets for each level of the hierarchy. Figure 3 (d) shows that other attribute subsets can be selected on each level of the hierarchy, now considering the values of the attributes of another instance.

Next, we present the results and analyses of the computational experiments that were carried out to evaluate the impact of the *lazy* attribute selection, when applied to the PPTD and SWV hierarchical classifiers. The computational experiments followed the same conditions presented in Section 4, albeit with the application of the *lazy* attribute selection method.

Table V shows, for each dataset and hierarchical classifiers (PPTD and SWV), the best results obtained. For each dataset (row) and each hierarchical classifier analysed (column), the best obtained hF value is shown and, on the side, the configuration that obtained the best value, represented by: the flat classifier (1-NN, 7-NN, or 9-NN), the selection method used, (*eager* or *lazy*), and the percentage

Table V. Best results for PPTD and SWV hierarchical classifiers

| Group A Datasets | PPTD hF | PPTD Strategy | SWV hF | SWV Strategy | Group B Datasets | PPTD hF | PPTD Strategy | SWV hF | SWV Strategy |
|---|---|---|---|---|---|---|---|---|---|
| **GPCRpfam** | 70.32 | 1-NN-EAGER(90) | 70.31 | 1-NN-LAZY(50) | **Church** | 23.43 | 9-NN-LAZY(10) | 24.16 | 9-NN-LAZY(10) |
| **GPCRprints** | 82.97 | 1-NN-EAGER(80) | 83.00 | 1-NN-EAGER(80)<br>1-NN-LAZY(80) | **CellCycle** | 29.91 | 9-NN-EAGER(40) | 30.96 | 9-NN-EAGER(40) |
| **GPCRprosit** | 69.26 | 1-NN-EAGER(70) | 69.41 | 1-NN-LAZY(30) | **Derisi** | 22.71 | 9-NN-EAGER(10) | 23.52 | 9-NN-EAGER(40) |
| **GPCRinterp** | 83.09 | 1-NN-EAGER(90) | 83.09 | 1-NN-EAGER(90)<br>1-NN-LAZY(90) | **Eisen** | 30.88 | 7-NN-LAZY(80) | 32.74 | 9-NN-LAZY(30) |
| **ECpfam** | 98.77 | 1-NN-EAGER(70) | 98.79 | 1-NN-LAZY(20) | **Expr** | 30.49 | 9-NN-LAZY(30) | 31.74 | 7-NN-LAZY(40) |
| **ECprints** | 98.19 | 1-NN-EAGER(80) | 98.21 | 1-NN-LAZY(30) | **Gasch1** | 31.37 | 7-NN-EAGER(80) | 31.95 | 9-NN-LAZY(70) |
| **ECprosite** | 98.81 | 1-NN-EAGER(70) | 98.81 | 1-NN-EAGER(70) | **Gasch2** | 27.92 | 9-NN-EAGER(70) | 28.93 | 9-NN-EAGER(40) |
| **ECinterpro** | 99.07 | 1-NN-EAGER(30)<br>1-NN-LAZY(10) | 99.15 | 1-NN-LAZY(20) | **Phenotyp** | 23.23 | 7-NN-EAGER(20) | 27.08 | 9-NN-LAZY(10) |
| | | | | | **Sequence** | 25.63 | 9-NN-EAGER(30) | 25.48 | 9-NN-EAGER(40) |
| | | | | | **SPO** | 26.05 | 9-NN-EAGER(30) | 25.64 | 9-NN-EAGER(20) |

of attributes that led to the best result. As an example, for the *GPCRprosite* dataset, the best hF value attained was of 69.41, obtained by the SWV hierarchical classifier, using the 1-NN flat classifier, with the *lazy* attribute selection, and with a selection of 30% of the attributes. With the analysis of this table, it is possible to conclude that both attribute selection methods succeeded, in all 36 cases, in increasing the performance of the adopted hierarchical classifier. In no case did the use of all the attributes lead to the best result. The *lazy* strategy managed to improve even further some of the results found with the *eager* strategy. In three cases it obtained the same result found by the *eager* method, whilst in another 13 cases the *lazy* strategy managed to secure an even better result.

Table VI shows, for each dataset, the best result obtained and the strategies that reached them. Each strategy is represented by the hierarchical classifier applied, flat classifier used, type of attribute selection (*eager* or *lazy*), and the percentage of attributes that were selected. It is possible to see the importance of each attribute selection strategy, in an isolated manner, as each one individually obtained the best result for a different subset of datasets. The *eager* strategy got the better result, in an isolated manner, for seven datasets, and the *lazy* strategy got it for nine datasets, showing the importance of this new paradigm for attribute selection in the hierarchical classification domain.

Table VI. Best results found per dataset

| Datasets | hF | Strategy(ies) |
|---|---|---|
| GPCRpfam | 70.32 | PPTD/1-NN-EAGER(90) |
| GPCRprints | 83.00 | SWV/1-NN-EAGER(80), SWV/1-NN-LAZY(80) |
| GPCRprosite | 69.41 | SWV/1-NN-LAZY(30) |
| GPCRinterpro | 83.09 | PPTD/1-NN-EAGER(90), SWV/1-NN-EAGER(90), SWV/1-NN-LAZY(90) |
| ECpfam | 98.79 | SWV/1-NN-LAZY(20) |
| ECprints | 98.21 | SWV/1-NN-LAZY(30) |
| ECprosite | 98.81 | PPTD/1-NN-EAGER(70), SWV/1-NN-EAGER(70) |
| ECinterpro | 99.15 | SWV/1-NN-LAZY(20) |
| Church | 24.16 | SWV/9-NN-LAZY(10) |
| CellCycle | 30.96 | SWV/9-NN-EAGER(40) |
| Derisi | 23.52 | SWV/9-NN-EAGER(40) |
| Eisen | 32.74 | SWV/9-NN-LAZY(30) |
| Expr | 31.74 | SWV/7-NN-LAZY(40) |
| Gasch1 | 31.95 | SWV/9-NN-LAZY(70) |
| Gasch2 | 28.93 | SWV/9-NN-EAGER(40) |
| Phenotype | 27.08 | SWV/9-NN-LAZY(10) |
| Sequence | 25.63 | PPTD/9-NN-EAGER(30) |
| SPO | 26.05 | PPTD/9-NN-EAGER(30) |

## 7. CONCLUSION

In this work, we evaluated the introduction of different attribute selection strategies in two hierarchical classifiers. It was possible to notice that, for the 18 hierarchical datasets used in the computational experiments, the best result was obtained by the hierarchical classifiers when some attribute selection strategy was carried out. Therefore, the results showed the importance of adopting attribute selection techniques, also in the hierarchical classification domain.

We could also observe that, not only flat classification strategies, but also hierarchical classification methods can benefit from the use of the *lazy* attribute selection paradigm, recently proposed by Pereira et al. [2011]. The *lazy* strategy postpones the attribute selection until the moment of the classification of new instances. In the conducted experiments, the *lazy* attribute selection was able to obtain the best result, in an isolated manner, in nine of the 18 datasets explored.

## REFERENCES


Barutcuoglu, Z. and DeCoro, C. Hierarchical Shape Classification Using Bayesian Aggregation. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications*. Matsushima, Japan, pp. 44–44, 2006.

Clare, A. and King, R. D. Predicting Gene Function in Saccharomyces Cerevisiae. In *Proceedings of the European Conference on Computational Biology*. Paris, France, pp. 42–49, 2003.

Costa, E. P., Lorena, A. C., Carvalho, A. C. P. L. F., and Freitas, A. A. Top-down Hierarchical Ensembles of Classifiers for Predicting G-Protein-Coupled-Receptor functions. In Bazzan, Craven, and Martins (Eds.), *Advances in Bioinformatics and Computational Biology*. Lecture Notes in Computer Science, vol. 5167. pp. 35–46, 2008.

Dumais, S. and Chen, H. Hierarchical Classification of Web Content. In *Proceedings of Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 256–263, 2000.

Guyon, I. and Elisseeff, A. An Introduction to Feature Extraction. In *Feature Extraction, Foundations and Applications*. pp. 1–24, 2006.

Hall, M. A. Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In *Proceedings of the International Conference on Machine Learning*. pp. 359–366, 2000.

Han, J. and Kamber, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Third Edition, 2011.

Holden, N. and Freitas, A. A. A Hybrid PSO/ACO Algorithm for Classification. In *Proceedings of the GECCO Workshop on Particle Swarms: The Second Decade*. pp. 2745–2750, 2007.

Holden, N. and Freitas, A. A. Hierarchical Classification of Protein Function with Ensembles of Rules and Particle Swarm Optimisation. *Soft Computing* 13 (3): 259–272, 2009.

Jain, R. *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley, 1991.

Kiritchenko, S., Matwin, S., and Famili, A. F. Functional Annotation of Genes using Hierarchical Text Categorization. In *Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology*, 2005.

Koller, D. and Sahami, M. Hierarchically Classifying Documents Using Very Few Words. In *Proceedings of the International Conference on Machine Learning*. Nashville, USA, pp. 170–178, 1997.

Liu, H. and Setiono, R. A Probabilistic Approach to Feature Selection - A Filter Solution. In *Proceedings of the International Conference on Machine Learning*. pp. 319–327, 1996.

Paes, B. C., Plastino, A., and Freitas, A. A. Improving Local Per Level Hierarchical Classification. *Journal of Information and Data Management* 3 (3): 394–409, 2012.

Pereira, R. B., Plastino, A., Zadrozny, B., Merschmann, L. H. C., and Freitas, A. A. Lazy Attribute Selection: choosing attributes at classification time. *Intelligent Data Analysis* 15 (5): 715–732, 2011.

Secker, A., Davies, M. N., Freitas, A. A., Clark, E. B., Timmis, J., and Flower, D. R. Hierarchical Classification of G-Protein-Coupled Receptors with Data-Driven Selection of Attributes and Classifiers. *International Journal of Data Mining and Bioinformatics* 4 (2): 191–210, 2010.

Silla, C. and Freitas, A. A. A Survey of Hierarchical Classification Across Different Application Domains. *Data Mining and Knowledge Discovery* 22 (1-2): 31–72, 2011.

Sun, A. and Lim, E.-P. Hierarchical Text Classification and Evaluation. In *Proceedings of the IEEE International Conference on Data Mining*. pp. 521–528, 2001.

Witten, I. H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Third Edition, 2011.

Yang, Y. and Pedersen, J. O. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the International Conference on Machine Learning*. pp. 412–420, 1997.