An Extensive Evaluation of Decision Tree-Based Hierarchical Multi-Label Classification Methods and Performance Measures

Ricardo Cerri
Departamento de Ciências de Computação - Universidade de São Paulo - Campus de São Carlos
Av. Trabalhador São-carlense, 400, Centro, 13560-970, São Carlos, SP, Brazil
phone: +55 16 3373-8161; email: cerri@icmc.usp.br

Gisele L. Pappa
Departamento de Ciências da Computação - Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, Pampulha, 31270-010, Belo Horizonte, MG, Brazil
phone: +55 31 3409-5867; email: glpappa@dcc.ufmg.br

André Carlos P. L. F. Carvalho
Departamento de Ciências de Computação - Universidade de São Paulo - Campus de São Carlos
Av. Trabalhador São-carlense, 400, Centro, 13560-970, São Carlos, SP, Brazil
phone: +55 16 3373-9691; email: andre@icmc.usp.br

Alex A. Freitas
School of Computing - University of Kent - UK
Canterbury, Kent, CT2 7NF
phone: +44 0 1227 827220; email: A.A.Freitas@kent.ac.uk

Hierarchical Multi-Label Classification is a complex classification problem where an instance can be assigned to more than one class simultaneously, and these classes are hierarchically organized with superclasses and subclasses, i.e., an instance can be classified as belonging to more than one path in the hierarchical structure. This article experimentally analyses the behaviour of different decision tree-based hierarchical multi-label classification methods based on the local and global classification approaches. The approaches are compared using distinct hierarchy-based and distance-based evaluation measures, when they are applied to a variation of real multi-label and hierarchical datasets' characteristics. Also, the different evaluation measures investigated are compared according to their degrees of consistency, discriminancy and indifferency. As a result of the experimental analysis, we recommend the use of the global classification approach and suggest the use of the Hierarchical Precision and Hierarchical Recall evaluation measures.

Key words: hierarchical, multi-label, classification, performance measures, global and local approaches

1. INTRODUCTION

In most of the classification problems described in the literature, a classifier assigns a single class to a given instance $\mathbf{x}_i$ and the classes form a non-hierarchical, flat, structure, with no consideration of superclasses or subclasses. However, in many real-world classification problems, one or more classes can be divided into subclasses or grouped into superclasses, and instances can belong to more than one class simultaneously at a same hierarchical level. In this case, the classes follow a hierarchical structure, usually a tree or a Directed Acyclic Graph (DAG). These problems are known in the literature of Machine Learning (ML) as Hierarchical Multi-Label Classification (HMC) problems. They are more complex than conventional classification problems, which are flat and single-label, since new instances can be classified into the classes associated with two or more paths in the class hierarchy. These problems are very common, for example, in the classification of genes and identification of protein functions (Blockeel *et al.*, 2002; Clare and King, 2003; Struyf *et al.*, 2005; Kiritchenko *et al.*, 2005; Barutcuoglu *et al.*, 2006; Vens *et al.*, 2008; Alves *et al.*, 2008; Obozinski *et al.*, 2008; Valentini, 2009, 2011; Alves *et al.*, 2010; Schietgat *et al.*, 2010; Otero *et al.*, 2010; Cerri *et al.*, 2011; Cerri and Carvalho, 2011; Pugelj and Džeroski, 2011; Bi and Kwok, 2011), and text classification (Sun and Lim, 2001; Kiritchenko

et al., 2004; Rousu *et al.*, 2006; Cesa-Bianchi *et al.*, 2006; Mayne and Perry, 2009). HMC problems can be defined as complex classification problems which encompass the characteristics of both hierarchical single-label problems and non-hierarchical multi-label problems.

In hierarchical single-label classification problems, each instance is assigned to a single path of the hierarchical structure. The process of classification of new instances may be a mandatory leaf node classification, when a new instance must be assigned to a leaf node, or a non-mandatory leaf node classification, when the most specific class assigned to a new instance can be an internal (non-leaf) node of the class hierarchy (Freitas and Carvalho, 2007). Two approaches have been adopted in the literature to deal with the class hierarchy in hierarchical problems: top-down or local, and one-shot or global.

The local approach uses local information to consider the hierarchy of classes. During the training phase, the hierarchy of classes is processed level by level, producing one or more classifiers for each level of the hierarchy. This process produces a tree of classifiers. The root classifier is induced with all training instances. At each other level, a classifier is induced using just local instances associated with classes at that level. In the test phase, when an instance is assigned to a class that is not a leaf node, it is further classified into one subclass of this class. A deficiency of this approach is the propagation of classification errors in a class node to its descendant nodes in the class hierarchy. However, it allows the use of any traditional classification algorithm, because each local classification algorithm is a conventional, flat classification algorithm.

The global approach induces a unique classification model considering the class hierarchy as a whole, avoiding the error propagation problem of the local approach. After the model induction, the classification of a new instance occurs in just one step. Hence, traditional classification algorithms cannot be used, unless adaptations are made to consider the hierarchy of classes.

In non-hierarchical multi-label problems, each instance can be assigned to zero, one or more classes simultaneously. Similar to hierarchical single-label problems, where the local and global approaches can be used to solve the classification task, two main approaches can be used to solve non-hierarchical multi-label problems, named algorithm dependent and algorithm independent (Carvalho and Freitas, 2009). The algorithm independent approach transforms the original multi-label problem into a set of single-label problems and, as in the local approach for hierarchical problems, any traditional classification algorithm can be used. In the algorithm dependent approach, as the name suggests, new algorithms are developed specifically for multi-label problems, or traditional algorithms are modified to cope with these problems. The global approach used in hierarchical problems can be seen as an algorithm dependent approach, as new or modified algorithms are used.

In HMC problems, the characteristics of the hierarchical and multi-label problems are combined, and an instance can be assigned to two or more subtrees of the class hierarchy. As stated by Vens *et al.* (2008), the HMC problem can be formally described as follows:

Given:

- a space of instances $\mathbf{X}$;
- a class hierarchy $(C, \leqslant_h)$, where C is a set of classes and $\leqslant_h$ is a partial order representing the superclass relationship (for all $c_1, c_2 \in C : c_1 \leqslant_h c_2$ if and only if c_1 is a superclass of c_2);
- a set T of tuples $(\mathbf{x}_i, C_i)$ with $\mathbf{x}_i \in \mathbf{X}$ and $C_i \subseteq C$, such that $c \in C_i \Rightarrow \forall c' \leqslant_h c : c' \in C_i$;
- a quality criterion q that rewards models with high accuracy and low complexity.

Find:

- a function $f : \mathbf{X} \rightarrow 2^C$, where 2^C is the powerset of C, such that $c \in f(\mathbf{x}) \Rightarrow \forall c' \leqslant_h c : c' \in f(\mathbf{x})$ and f optimizes q.

The quality criterion q can be the mean accuracy of the predicted classes or the distances between the predicted and true classes in the class hierarchy. It can also consider that misclassifications in levels closer to the root node are worse than misclassifications in deeper levels. Besides, the complexity of the classifiers and the induction time can be taken into account as quality criteria.

Although the given HMC definition says that an instance belongs to and has to be classified into proper hierarchical paths, there are some works that allow inconsistent predictions. Examples are the works of Cesa-Bianchi *et al.* (2006), Kiritchenko *et al.* (2006), Obozinski *et al.* (2008), Valentini (2011) and Cerri and Carvalho (2011), where predictions inconsistent with the hierarchy are made, and then an additional step of making the class assignments consistent with the hierarchy is required.

An example of HMC problem is illustrated in Figure 1, where the class hierarchy is represented by a tree. In this example, a newspaper report can address subjects related to computer sciences and soccer and, therefore,

be classified into both sciences/computing and sports/collective/soccer classes. The class prediction for a new instance generates a subtree. In the figure, the nodes with a rectangle and the nodes with an ellipse represent two predicted paths in the tree for a new instance, sciences/computing and sports/collective/soccer, respectively.

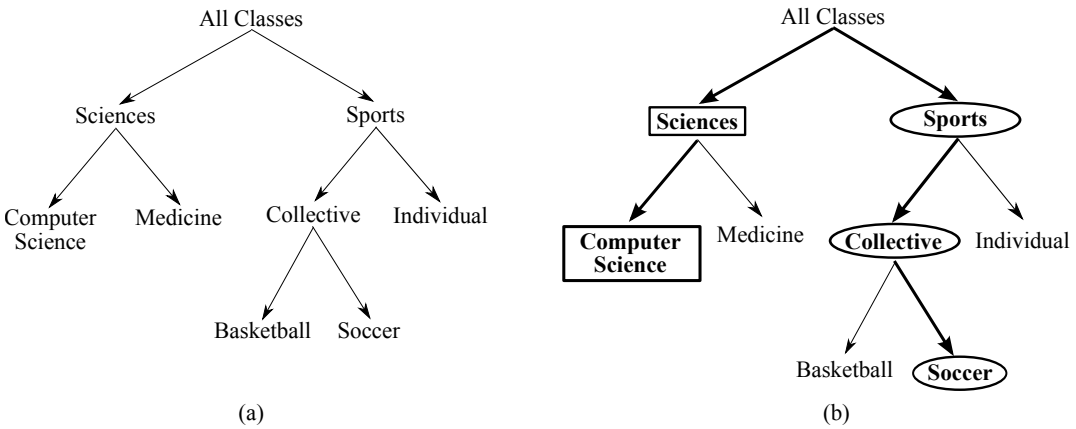


FIGURE 1. HMC problem structured as a tree. (a) Class hierarchy. (b) Predictions generating a subtree.

There are several works proposing HMC methods and using HMC or flat performance measures for specific datasets (Sun and Lim, 2001; Vens *et al.*, 2008; Alves *et al.*, 2010; Otero *et al.*, 2010; Cerri *et al.*, 2011; Cerri and Carvalho, 2011; Pugelj and Džeroski, 2011; Bi and Kwok, 2011). The work of Ceci and Malerba (2007) evaluates hierarchical classifiers using flat (non-hierarchical) evaluation measures. In (Sokolova and Lapalme, 2009) a series of flat, multi-label and hierarchical evaluation measures were analysed according to the type of changes to a confusion matrix that do not change a measure, but the analyses were only theoretical. In (Brucker *et al.*, 2011) the authors performed experiments with a series of flat multi-label classifiers. Hierarchies were then extracted from the flat results obtained, and then hierarchical and flat classification measures were used in the evaluation. In (Silla and Freitas, 2010), HMC evaluation measures were analysed, but no experiments were performed comparing the measures.

Although these works compare different methods and measures, we did not find guidelines associating the characteristics of hierarchical and multi-label datasets to the performance of different methods evaluated by distinct HMC performance measures. This paper experimentally compares different HMC methods and different HMC predictive performance measures specific for HMC problems. More precisely, the main contributions of this work are the following:

- The evaluation and comparison of hierarchy-based and distance-based predictive performance measures, which are specific for HMC problems, when used in a collection of 12 real datasets with different hierarchical and multi-label characteristics;
- The analysis of the predictive performance of four different decision tree-based HMC methods, two of them based on the local approach and two based on the global approach, in these 12 datasets.

In our experimental analysis, we vary four different characteristics of HMC problems, as follows: (i) the percentage of multi-label instances, (ii) the number of classes assigned to an instance, (iii) the unbalance of the class hierarchy, and (iv) the maximum number of child nodes per internal node. The experiments were designed to investigate the effect of different values of those problem characteristics (corresponding to different datasets) in the results of four decision tree-based HMC methods (two based on the local approach and two based on the global approach), as evaluated by 10 different performance evaluation measures. More precisely, for each of the aforementioned four problem (dataset) characteristics being varied, we address the following research questions:

- Q_1**:** Does a specific evaluation measure favour a specific classification approach (global or local) when used to compare global and local based methods?
- Q_2**:** Which classification approach (global or local) is better overall, considering the four aforementioned classification scenarios?
- Q_3**:** Are global/local methods better in predicting more specific/general classes?
- Q_4**:** How different hierarchical and multi-label characteristics influence different evaluation measures?

- Q_5**:** Which evaluation measure is more suitable to use in the classification scenarios investigated?

For the experiments performed in this work, we have chosen methods that induce decision trees, since there are works that have already shown that decision trees are a good alternative for HMC classification (Clare and King, 2003; Vens *et al.*, 2008; Alves *et al.*, 2010; Otero *et al.*, 2010), and also because the classifiers produced are interpretable.

The rest of this article is organized as follows: Section 2 reviews the hierarchical classification performance measures used in this work. Section 3 presents the HMC methods used in the experiments performed in this work. The experiments carried out are described in Section 4, together with an analysis of the results obtained. Finally, Section 5 presents the main conclusions regarding the experimental results and suggestions for future work.

2. REVIEW OF EVALUATION MEASURES

Classification accuracy measures for conventional (flat) classification problems are usually inadequate for hierarchical multi-label problems. Apart from not considering the problem's hierarchical class structure, and the fact that an instance can simultaneously belong to more than one class, conventional classification accuracy measures ignore that the difficulty of classification usually increases with the depth of the classes to be predicted. In hierarchical classification, more specific classes are often harder to predict than generic ones, and conventional measures assume misclassification costs to be independent of the positions of classes in the hierarchy. Furthermore, in multi-label classification, these measures do not consider that an instance can be assigned to just a subset of its true classes.

As alternatives to conventional evaluation measures for classification problems, specific measures for hierarchical, multi-label and hierarchical multi-label classifiers have been proposed. Here we are interested in two broad groups of hierarchical multi-label evaluation measures, namely: (i) Hierarchy-Based Evaluation Measures, and (ii) Distance-Based Evaluation Measures. While hierarchy-based measures are based only on the hierarchical class structure (only subclasses and superclasses), distance-based measures also consider the distance between the predicted and true classes in the hierarchy structure.

Although many works in the literature evaluate the performance of hierarchical multi-label classifiers, there is no consensus on which measure is more appropriate to which type of dataset or method. This section reviews the evaluation measures used in this work and discusses their pros and cons, in order to later contrast some of them in experiments involving datasets with different characteristics and different HMC methods.

2.1. Hierarchy-Based Evaluation Measures

Hierarchy-based evaluation measures consider both the ancestors and the descendants of the predicted classes in the hierarchy when evaluating a classifier. In subsection 2.1.1 we discuss two variations of Hierarchical Precision and Recall, and in subsection 2.1.2 we present the Hierarchical Loss Function, which is based on the traditional 0/1-loss measure.

2.1.1. *Hierarchical Precision and Recall.* In (Kiritchenko *et al.*, 2004), two evaluation measures based on the conventional precision and recall measures were proposed to take into account hierarchical relationships between classes. These two measures, named Hierarchical Precision and Hierarchical Recall, were formally defined in (Kiritchenko *et al.*, 2005). These evaluation measures were later used in (Eisner *et al.*, 2005) and (Kiritchenko *et al.*, 2006).

The Hierarchical Precision and Recall measures consider that an instance belongs not only to its predicted classes, but also to all its ancestor classes in the hierarchical structure. Hence, given an instance $(\mathbf{x}_i, C'_i)$, where $\mathbf{x}_i$ belongs to the space $\mathbf{X}$ of instances, C'_i is the set of predicted classes for $\mathbf{x}_i$, and C_i is the set of true classes of $\mathbf{x}_i$. The sets C_i and C'_i can be extended to contain their corresponding *ancestor classes* as: $\widehat{C_i} = \bigcup_{c_k \in C_i} Ancestors(c_k)$ and $\widehat{C'_i} = \bigcup_{c_l \in C'_i} Ancestors(c_l)$, where $Ancestors(c_k)$ denotes the set of ancestors of class c_k.

Equations (1) and (2) present the Hierarchical Precision and Recall (hP and hR) measures. These measures count the number of classes correctly predicted, together with the number of ancestor classes correctly predicted (Kiritchenko *et al.*, 2005). Figure 2 presents an example of how to calculate these measures. In the figure, each set of two hierarchical structures, one above and one below, represents the true and predicted classes for an instance. In Figure 2(a), solid circles represent the true classes of an instance, and in Figure 2(b), bold circles represent the predicted classes of the corresponding above instance, with an arrow showing the deepest predicted class.

$$hP = \frac{\sum_i |\widehat{C}_i \cap \widehat{C}'_i|}{\sum_i |\widehat{C}'_i|} \tag{1}$$

$$hR = \frac{\sum_i |\widehat{C}_i \cap \widehat{C}'_i|}{\sum_i |\widehat{C}_i|} \tag{2}$$

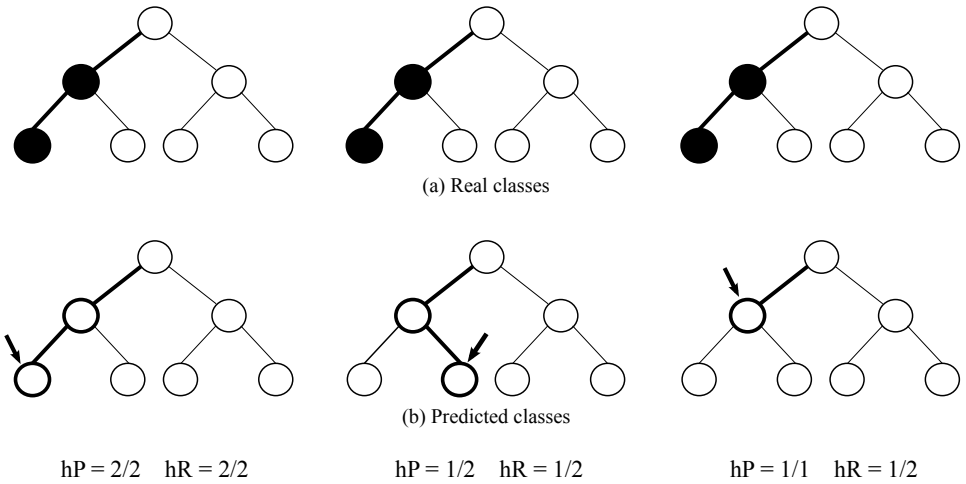


FIGURE 2. Graphical example of the use of the Hierarchical Precision and Recall Measures. Adapted from (Kiritchenko *et al.*, 2004).

As can be seen, all nodes in the path from the root node to the predicted class node for an instance are bold, indicating that the ancestor classes of the predicted classes are also assigned to the instance. The edges from the root node to the node that represents the deepest predicted class of an instance are also shown in bold. The hP and hR values for the three different predictions are also illustrated in the figure.

Either Hierarchical Precision or Hierarchical Recall used alone is not adequate for the evaluation of hierarchical classifiers (Sebastiani, 2002). Both measures have to be considered together or combined in a single F-measure. Thus, the hP and hR measures are combined on a hierarchical extension of the F-measure, named Hierarchical-F_β, presented in Equation (3). In Equation (3), β represents the importance assigned to the values of hP and hR. As the value of β increases, the weight assigned to the value of hR also increases. On the other hand, when the value of β decreases, the weight assigned to hP increases.

$$Hierarchical - F_\beta = \frac{(\beta^2 + 1) \times hP \times hR}{\beta^2 \times hP + hR} \tag{3}$$

In the same direction as Kiritchenko *et al.* (2005), Ipeirotis *et al.* (2001) also measured the Hierarchical Precision and Recall for an instance through the intersection of the predicted and true classes. However, unlike the definitions of hierarchical precision and recall previously described, Ipeirotis *et al.* (2001) expanded the set of true and predicted classes by including all their subclasses instead of their superclasses. Thus, given the set of predicted (C'_i) and true (C_i) classes, they are extended to contain their corresponding *descendant classes* as: $\widehat{C}'_i = \bigcup_{c_k \in C'_i} Descendants(c_k)$ and $\widehat{C}_i = \bigcup_{c_l \in C_i} Descendants(c_l)$, where $Descendants(c_k)$ denotes the set of descendants of the class c_k. This new definition of $\widehat{C}'_i$ and $\widehat{C}_i$ can be directly used in the formulas presented in Equations (1) and (2). Although the authors claimed that this measure captures the nuances of hierarchical classification, we do not think it is totally correct for the HMC task, because expanding a set of classes to contain their corresponding subclasses can result in a wrong classification. As an example, if a document is classified in the class "sports", it is not necessarily classified in both subclasses "basketball" and "soccer".

2.1.2. *Hierarchical Loss Function.* The Hierarchical Loss Function (H-Loss), proposed in (Cesa-Bianchi *et al.*, 2006), is based on the concept that, when a misclassification occurs in a class of the hierarchy, no additional penalizations should be given to misclassifications in the subtree of this class. That is, if a misclassification occurs in class c'_j, additional errors in the subtree rooted at c'_j are not important. As an example, if a classifier erroneously classifies a document as belonging to the class "sports", this classifier should not be penalized again by erroneously classifying it in the subclass "soccer".

Consider that the set of true classes assigned to a given instance $\mathbf{x}_i$ is any subset of the set C formed by all classes, including the empty set. This subset is represented by a vector $(c_1, \ldots, c_{|C|})$, where a class c_j belongs to the subset of classes of instance $\mathbf{x}_i$ if and only if $c_j = 1$. Before defining the H-Loss function, two measures regarding the discrepancy between a multi-label prediction for $\mathbf{x}_i$ ($C' = (c'_1, \ldots, c'_{|C|})$), and the true set of classes of $\mathbf{x}_i$ ($C = (c_1, \ldots, c_{|C|})$), for each instance, need to be introduced. The first is the zero-one loss ($l_{0/1}(C, C')$), presented in Equation (4). The second is the symmetric difference loss ($l_\Delta(C, C')$), defined in Equation (5). Note that these equations do not consider the hierarchical structure of the problem, only multiple labels. Based on these two measures, Cesa-Bianchi *et al.* (2006) proposed the H-Loss function ($l_H(C, C')$), defined in Equation (6). In the equations, $\mathbb{1}\{\cdot\}$ is an indicator function that yields 1 if the provided equation is true and 0 otherwise.

$$l_{0/1}(C, C') = 1, \text{ if } \exists j \in \{1, \ldots, |C|\} : c_j \neq c'_j \tag{4}$$

$$l_\Delta(C, C') = \sum_{j=1}^{|C|} \mathbb{1}\{c_j \neq c'_j\} \tag{5}$$

$$l_H(C, C') = \sum_{j=1}^{|C|} \mathbb{1}\{c_j \neq c'_j \wedge Ancestors(c_j) = Ancestors(c'_j)\} \tag{6}$$

This measure is based on the fact that, given a hierarchical structure G, this structure can be considered a forest composed by trees defined on the set of classes of the problem. A multi-label classification $C' \in \{0, 1\}^{|C|}$ respects the structure G if and only if C' is the union of one or more paths of G, where each path starts in a root class and not necessarily ends up in a leaf class. Hence, all paths of G, from a root class to a leaf class, are examined. When a class c'_j is found and $c'_j \neq c_j$, the value 1 is added to the H-Loss function, and all predictions in the subtrees rooted in the class c'_j are discarded. Given this definition, we can say that $l_{0/1} \leqslant l_H \leqslant l_\Delta$.

Figure 3 shows the concepts and use of the H-Loss function. In the four class hierarchies illustrated, round gray nodes represent the classes being predicted for an instance, while squared gray nodes represent the true classes of the instance. Note that in Figure 3(a) the classes predicted do not respect the hierarchical structure of G (parents of predicted leaf nodes are not predicted), whereas in Figure 3(b) the structure is respected. Figure 3(c) shows the true classes of the instance classified in Figure 3(b), and Figure 3(d) shows the application of the H-Loss function considering the multi-label classifications illustrated in (b) and (c). Only the nodes marked with an "X" are considered when calculating the H-Loss. As can be seen, the values of the zero-one loss and symmetric difference loss functions are 1 and 6, respectively. The H-Loss function returns the value 4. Recall that the lower the value of the function H-Loss, the better the performance of the classifier.

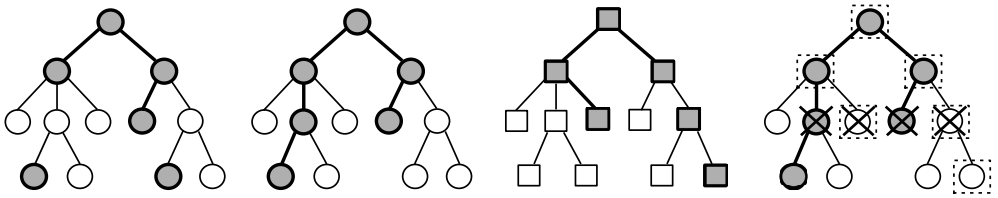


FIGURE 3. Graphical representation of the H-Loss function. Adapted from (Cesa-Bianchi *et al.*, 2006).

As the Hierarchical Loss Function measure ignores errors in subtrees of classes erroneously assigned to instances, the error propagation problem present in hierarchical classification is not taken into account. Hence, although some authors work with this measure, it cannot be easily compared to others in the literature.

2.2. Distance-Based Evaluation Measures

This class of measures is based on the assumption that closer classes in the hierarchy tend to be more similar to each other (representing a smaller classification error) than distant classes. Hence, these measures consider the distance between the true and predicted classes during evaluation. Section 2.2.1 reviews the Micro/Macro Distance-Based Hierarchical Precision and Micro/Macro Distance-Based Hierarchical Recall, and Section 2.2.2 discusses the most common ways of calculating distances between hierarchy nodes.

2.2.1. *Micro/Macro Distance-Based Hierarchical Precision and Recall.* The Micro/Macro Hierarchical Precision and Micro/Macro Hierarchical Recall measures, proposed by Sun and Lim (2001), are based on the distance between predicted and true classes. The Macro Hierarchical Precision and Recall initially calculate the performance obtained in each class separately, and return the average of these values for each measure. The Micro Hierarchical Precision and Recall measures, on the other hand, calculate the average of the performance obtained in each instance of a dataset. Hence, while the macro measures are considered a per class mean performance measure, the micro measures are considered a per instance mean performance measure (Yang, 1999).

For each of these measures, it is necessary to first define, for each class, the *contribution* of the instances erroneously assigned to that class. This contribution is defined according to an acceptable distance (number of edges (Dis_θ)) between a predicted and a true class, which must be higher than zero. As an example, when using the value $Dis_\theta = 2$, the instances which are "slightly" misclassified (with just 2 edges between the predicted and true class in the class hierarchy) give zero contribution in the calculation of the measures, while the instances that are more seriously misclassified (with more than 2 edges between the predicted and true class) contribute negatively to the values of the measures. Equations (7) and (8) specify the contribution of an instance $\mathbf{x}_i$ to a class c_j, where $\mathbf{x}_i.agd$ and $\mathbf{x}_i.lbd$ are, respectively, the predicted and true classes of $\mathbf{x}_i$. $Dis(c, c'_j)$ is the distance between a true class c and a predicted class c'_j, and can be calculated using any of the approaches described in Section 2.2.2.

- If $\mathbf{x}_i$ is a False Positive:

$$Con(\mathbf{x}_i, c'_j) = \sum_{c \in \mathbf{x}_i.lbd} \left(1.0 - \frac{Dis(c, c'_j)}{Dis_\theta}\right) \tag{7}$$

- If $\mathbf{x}_i$ is a False Negative:

$$Con(\mathbf{x}_i, c'_j) = \sum_{c \in \mathbf{x}_i.agd} \left(1.0 - \frac{Dis(c, c'_j)}{Dis_\theta}\right) \tag{8}$$

The contribution of an instance $\mathbf{x}_i$ is then restricted to the values $[-1, 1]$. This refinement, denoted by $RCon(\mathbf{x}_i, c'_j)$, is defined in Equation (9).

$$RCon(\mathbf{x}_i, c'_j) = min(1, max(-1, Con(\mathbf{x}_i, c'_j))) \tag{9}$$

The total contribution of False Positives (FP) ($FpCon_j$) and False Negatives (FN) ($FnCon_j$), for all instances, is defined in Equations (10) and (11).

$$FpCon_j = \sum_{\mathbf{x}_i \in FP_j} RCon(\mathbf{x}_i, c'_j) \tag{10}$$

$$FnCon_j = \sum_{\mathbf{x}_i \in FN_j} RCon(\mathbf{x}_i, c'_j) \tag{11}$$

After the calculation of the contributions of each instance, the values of the Hierarchical Precision and Recall for each class are calculated as defined in Equations (12) and (13).

$$Pr_j^{CD} = \frac{max(0, |TP_j| + FpCon_j + FnCon_j)}{|TP_j| + |FP_j| + FnCon_j} \tag{12}$$

$$Re_j^{CD} = \frac{max(0, |TP_j| + FpCon_j + FnCon_j)}{|TP_j| + |FN_j| + FpCon_j} \tag{13}$$

Finally, the extended values of Hierarchical Precision and Recall (Hierarchical Micro Precision and Recall) are presented in Equations (14) and (15), where m represents the number of classes. According to the value of Dis_θ, the values of $FpCon_j$ and $FnCon_j$ can be negative. Therefore, a *max* function is applied to the numerators of the Equations (14) and (15) to make their values not lower than zero. As $FpCon_j \leqslant |FP_j|$, when $|TP_j| + |FP_j| + FnCon_j \leqslant 0$, the numerator $max(0, |TP_j| + FpCon_j + FnCon_j) = 0$. The $\hat{Pr}^{\mu CD}$ value can be considered zero in this case. The same rule is applied to the calculation of $\hat{Re}^{\mu CD}$ (Sun and Lim, 2001).

$$\hat{Pr}^{\mu CD} = \frac{\sum_{j=1}^{m}(max(0, |TP_j| + FpCon_j + FnCon_j))}{\sum_{j=1}^{m}(|TP_j| + |FP_j| + FnCon_j)} \tag{14}$$

$$\hat{Re}^{\mu CD} = \frac{\sum_{j=1}^{m}(max(0, |TP_j| + FpCon_j + FnCon_j))}{\sum_{j=1}^{m}(|TP_j| + |FN_j| + FpCon_j)} \tag{15}$$

The Hierarchical Macro Precision and the Hierarchical Macro Recall measures can also be obtained using Equations (16) and (17), where m represents the number of classes.

$$\hat{Pr}^{M_{CD}} = \frac{\sum_{j=1}^{m} Pr_j^{CD}}{m} \tag{16}$$

$$\hat{Re}^{M_{CD}} = \frac{\sum_{j=1}^{m} Re_j^{CD}}{m} \tag{17}$$

Just like the *hP* and *hR* measures, used in (Kiritchenko *et al.*, 2005), the Hierarchical Micro/Macro Precision and Recall measures can also be combined into the Hierarchical-F_β measure (Equation (3)).

2.2.2. *Methods for Calculating Distances Between Classes.* The Micro/Macro Hierarchical Precision and Recall use the distance between two classes in the hierarchy to evaluate the predictions made by a classifier. This section describes a few methods that can be employed to calculate these distances, which are usually defined as a function of two components: (i) the number of edges between the predicted class and the true class, and (ii) the depth of the predicted and true classes in the hierarchy.

The most common method, used in the standard version of measures, is to consider the distance as the number of edges that separate the true and predicted classes. Additionally, weights can be assigned to each edge of the class hierarchy, so that the misclassification between the predicted and true classes is given by the sum of the weights of the edges in the path between the two classes.

There are different ways of calculating the paths between classes depending on the hierarchy structured being considered. If the structure is a tree, there can be only one path between two classes, but if the hierarchy is a DAG, there can be more than one path between two classes, based on the number of superclasses of a class. In the final classification, one can consider two interpretations of the class hierarchy: if an instance belongs to a class c_j, it belongs to all superclasses of c_j, or it belongs to at least one superclass of c_j. Although in theory an evaluation measure could indeed use any of the previous two types of interpretation, in practice only the former (a class belongs to all its superclasses) is used, and corresponds to the HMC definition we use in this paper.

Moreover, in the experiments performed , we consider only hierarchies structured as trees, as done in (Wang *et al.*, 1999; Dekel *et al.*, 2004). When using hierarchies structured as trees, Wang *et al.* (1999) considered the distances between the true and predicted classes in a hierarchical structure in order to rank hierarchical classification rules. They defined the distance between two classes as the shortest path (number of edges) between the classes. Dekel *et al.* (2004) also used the distances between true and predicted classes in a tree hierarchy to evaluate a final classification. However, in the latter, the authors defined a distance function $\gamma(c_j, c'_j)$ as the number of edges in the unique path between a true class c_j and a predicted class c'_j.

In order to consider the importance of the classes according to the levels they belong to, there are many ways of choosing weights for edges. One of the most common is to consider that weights of edges at deeper levels should be lower than weights of edges in higher levels. Holden and Freitas (2006), for example, assigned weights that were exponentially decremented as the depth of the edges in a tree hierarchy increased. In (Vens

et al., 2008), the authors proposed a weighting technique that can be applied to DAG and tree hierarchies. The authors defined the weight of a class c_j as the recurrence relation $w(c_j) = w_0 \cdot w(par(c_j))$, with $par(c_j)$ being the parent class of c_j, and the weights assigned to the first level classes equal to w_0. The generalization for DAG hierarchies can be obtained replacing $w(par(c_j))$ by an aggregation function (*sum, min, max, average*) computed over the weights assigned to the parents of class c_j (Vens *et al.*, 2008). Assigning weights to the edges of the hierarchy, however, presents some problems, specially when the hierarchy is very unbalanced, and its depth varies significantly by different leaf nodes. In this case, a misclassification involving predicted and true classes near the root node receives a lower penalization than a misclassification involving classes at levels more distant from the root node.

In this direction, Lord *et al.* (2003) showed that when two classes are located in different subtrees of the hierarchy, and the route between them has to go through the root node, the fact that one class is in a deeper level than the other does not necessarily means that the class located in the deeper level provides more significant information than the class located in the higher level. Therefore, considering depth without considering the information associated with the classes may be a problem.

Figure 4 illustrates the problem of assigning weights to edges of a hierarchy. Here the filled ellipses represent the true classes of an instance and the bold ellipses represent the predicted classes. Consider an instance which belongs to class "11.04.03.01" (True), and two predicted classes "11.02.03.04" (Predicted 1) and "11.06.01" (Predicted 2). In the latter case, Predicted 2 would receive a lower penalization because the path between the predicted and true classes is shorter. This penalization is unfair, as the only reason the prediction was made in a class closer to the root was because the corresponding subtree does not have leaf nodes.

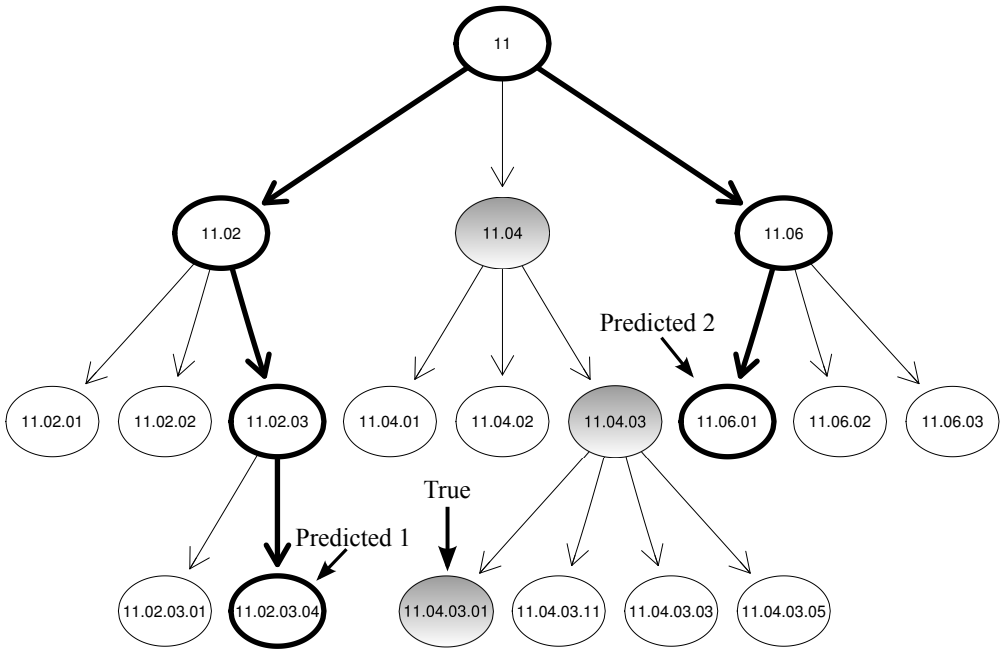


FIGURE 4. Example of class hierarchy.

3. HIERARCHICAL MULTI-LABEL METHODS

This section presents the HMC methods to be used in the experiments reported in this work, namely HMC-Binary-Relevance, HMC-Label-Powerset, HMC4.5 and Clus-HMC. First, the methods are categorized according to the hierarchical classification algorithm's taxonomy proposed by Silla and Freitas (2010). In this taxonomy, a hierarchical classification algorithm is described by a 4-tuple $< \Delta, \Xi, \Omega, \Theta >$, where:

- Δ: indicates if the algorithm is Hierarchical Single-Label (SPP - Single Path Prediction) or Hierarchical Multi-Label (MPP - Multiple Path Prediction);
- Ξ: indicates the prediction depth of the algorithm - MLNP (Mandatory Leaf-Node Prediction) or NMLNP (Non-Mandatory Leaf-Node Prediction);
- Ω: indicates the taxonomy structure the algorithm can handle - T (Tree structure) or D (DAG structure);
- Θ: indicates the categorization of the algorithm under the proposed taxonomy - LCN (Local Classifier per Node), LCL (Local Classifier per Level), LCPN (Local Classifier per Parent Node) or GC (Global Classifier).

Table 1 briefly presents the selected methods, which are explained in more details in the next subsections. Note that the first two methods can be applied exclusively in trees (T). The HMC-BR method works with non-mandatory leaf-node prediction (NMLNP) and is based on the LCN approach, and the HMC-LP method works with mandatory leaf-node prediction (MLNP) and uses a LCPN approach. The last two, in contrast, work with non-mandatory leaf-node prediction (NMLNP) and are global (GC) methods. While HMC4.5 works with trees only, Clus-HMC can also deal with graphs.

It is important to recall the main differences between local and global methods. Local methods build classification models for a single node or level in the tree, generating a set of models. Global methods, in contrast, create a single model for the whole hierarchy, considering dependencies among classes.

TABLE 1. Main Characteristics of the Methods used in the Experiments.

Method	Categorization	Description
HMC-Binary-Relevance (Tsoumakas *et al.*, 2010)	$< MPP, NMLNP, T, LCN >$	Local method based on the popular Binary-Relevance classification method (Tsoumakas *et al.*, 2010), where a classifier is associated with each class and trained to solve a binary classification task.
HMC-Label-Powerset (Cerri and Carvalho, 2010)	$< MPP, MLNP, T, LCPN >$	Based on local label combination (Tsoumakas and Vlahavas, 2007), where the set of labels assigned to an instance, in each level, is combined into a new class.
HMC4.5 (Clare and King, 2003)	$< MPP, NMLNP, T, GC >$	Global hierarchical multi-label variation of the C4.5 algorithm (Quinlan, 1993), where the entropy formula is modified to cope with HMC problems.
Clus-HMC (Vens *et al.*, 2008)	$< MPP, NMLNP, D, GC >$	Global method based on the concept of Predictive Clustering Trees (PCTs) (Blockeel *et al.*, 1998), where a decision tree is structured as a cluster hierarchy.

3.1. HMC-Binary-Relevance (HMC-BR)

The HMC-BR method follows the local classification approach, and uses binary classifiers as base classifiers for each class in the hierarchy. It has the advantage of using any classifier to induce the models, and is a hierarchical variation of the popular Binary-Relevance classification method (Tsoumakas *et al.*, 2010). The method works with $|C|$ classifiers, where $|C|$ is the total number of classes present in the class hierarchy. To show how it works, suppose we have a HMC problem with three hierarchical levels, being 2/3/4 the number of classes in each hierarchical level, respectively. As each base classifier is associated with a class in the hierarchy, two classifiers are trained for the classes in the first level, three for the second level, and four for the third level. To choose the set of positive and negative instances for the training process, the sibling policy, as described in (Silla and Freitas, 2010) was chosen. As an example, the set of positive instances of the class "11.04" in Figure

4 consists of the instances assigned to the class "11.04" and all its subclasses, and the set of negative instances consists of the instances assigned to the classes "11.02" and "11.06" and all their subclasses.

The training process of the classifiers at the first level occurs in the same way as a non hierarchical multi-label classification problem, using the one-against-all strategy, i.e., instances assigned to the class node are considered as positive, and instances belonging to any other class are considered as negative. From the second level onwards, when a classifier is trained for a given class c_j, the training process is carried out considering only the instances that belong to the parent class of c_j. This procedure is repeated until a leaf class is reached or all binary classifiers' outputs are false.

When the training process is finished, a hierarchy of classifiers is obtained, and the classification of new instances is performed following a top-down strategy. Beginning with the first class of the first level and going until the last, when an instance is assigned to a class c_j in level l, the classification algorithm recursively calls all classifiers representing the children of class c_j in level $l+1$, until a leaf class is reached or the outputs of all binary classifiers are negative. Algorithm 1 presents the classification process of HMC-BR. In the algorithm, C is initially the set of the first level classes of the hierarchy.

Algorithm 1: Classification process of the HMC-BR method.

```
Procedure HMC-BR(x_i, C)
Input: instance x_i, set of classes C
Output: Classes
Classes ← ∅
foreach class c_j in C do
    if instance x_i predicted as being from class c_j then
        if c_j not leaf node then
            Children ← child classes of c_j in C
            Classes ← Classes ∪ {c_j} ∪ HMC-BR(x_i, Children)
        else
            Classes ← Classes ∪ {c_j}
return Classes
```

The HMC-BR is a simple method, but presents some disadvantages. First, it assumes that all classes are independent from each other, which is not always true. By ignoring possible correlations between classes, a classifier with poor generalization ability can be obtained. Another disadvantage is that, as many classification models are generated, the set of all classifiers become complex. Hence, if the base algorithm used is, for example, a rule generator, such as C4.5 (Quinlan, 1993) or Ripper (Cohen, 1995), the interpretability of the models is much more difficult than interpreting a tree from HMC4.5 or Clus-HMC.

Finally, the induction time of the model is high, as many classifiers are involved. On the other hand, the classification process happens in a more natural manner, since discriminating classes level by level is a classification process more similar to the classification performed by a human being. Additionally, the fact that each classifier deals with fewer classes may result in a simpler classification process.

3.2. HMC-Label-Powerset (HMC-LP)

HMC-LP uses a label combination process that transforms the original HMC problem into a hierarchical single-label problem. This label combination process considers the correlations between the sibling classes in order to overcome the previously mentioned disadvantage of HMC-BR (i.e., considering all classes as independent).

The HMC-LP method was proposed by Cerri and Carvalho (2010), and is a hierarchical adaptation of a non-hierarchical multi-label classification method named Label-Powerset, used in the works of Tsoumakas and Vlahavas (2007) and Boutell *et al.* (2004). For each instance, the method combines all the classes assigned to it, at a specific level, into a new and unique class.

Given an instance belonging to classes $A.D$ and $A.F$, and a second instance belonging to classes $E.G$, $E.H$, $I.J$ and $I.K$, where $A.D$, $A.F$, $E.G$, $E.H$, $I.J$ and $I.K$ are hierarchical structures such that $A \leqslant_h D$, $A \leqslant_h F$, $E \leqslant_h G$, $E \leqslant_h H$, $I \leqslant_h J$ and $I \leqslant_h K$ with A, E and I belonging to the first level and D, F, G, H, J and K belonging to the second level, the resulting combination of classes for the two instances would be a new hierarchical structure $C_A.C_{DF}$ and $C_{EI}.C_{GHJK}$, respectively. In this example, C_{DF} is a new label formed by the

combination of the labels D and F, and C_{GHJK} is a new label formed by the combination of the labels G, H, J and K. Figure 5 illustrates this process of label combination.

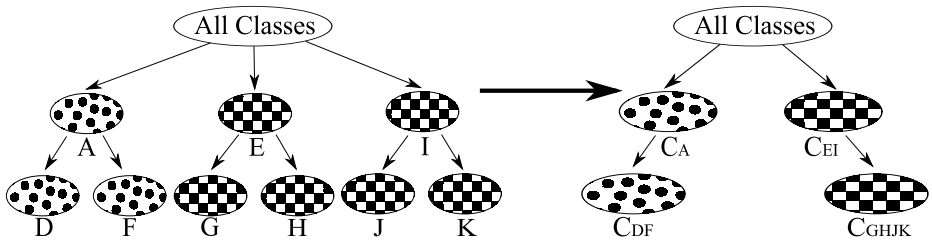


FIGURE 5. Label combination process of the HMC-LP method.

After the combination of classes, the original HMC problem is transformed into a hierarchical single-label problem, and a top-down approach is employed, using one or more multiclass classifiers per level. At the end of the classification, the original multi-label classes are recovered. Algorithm 2 shows the label combination procedure of the HMC-LP method.

Algorithm 2: Label combination procedure of the HMC-LP method.

Procedure LabelCombination(X, C)
Input: set of instances X, set of classes C
Output: *NewClasses*
foreach *instance* $\mathbf{x}_i$ *of the set of instances* X **do**
  foreach *level* l *of the class hierarchy* **do**
    $C_i \leftarrow$ subset of classes from C, assigned to an instance $\mathbf{x}_i$ in level l
    Get a new class $c_{i,l}$ for the instance $\mathbf{x}_i$ from C_i
    $NewClasses_{i,l} \leftarrow c_{i,l}$
return *NewClasses*

This procedure can considerably increase the number of classes involved in the problem. This happens when there are many possible multi-label combinations in the dataset, so that after the label combination, the new formed classes have few positive instances, resulting in a sparse dataset. Despite this disadvantage, if multiclass classifiers are used at each internal node instead of binary classifiers, the induction time might decrease considerably when compared with the HMC-Binary-Relevance method.

3.3. HMC4.5

The HMC4.5 method was proposed by Clare and King (2003), and is a variation of the C4.5 algorithm. The main modification introduced was the reformulation of the original entropy formula, to use the sum of the number of bits needed to describe membership and non-membership of each class instead of just the probability (relative frequency) of each class. The new entropy also uses information of the descendant classes of a given class in the hierarchy, incorporating the tree size in the entropy formula. The entropy can be defined as the amount of information necessary to describe an instance of the dataset, which is equivalent to the amount of bits necessary to describe all the classes of an instance.

Different from a standard C4.5 decision tree, the new formulation of the entropy allows leaf nodes of the HMC4.5 tree to represent a set of class labels. Thus, the classification output for a new instance $\mathbf{x}_i$ can be a set of classes, represented by a vector. The new entropy formula is presented in Equation (18) (Clare, 2003):

$$entropy = -\sum_{j=1}^{N}((p(c_j)\log_2 p(c_j)) + (q(c_j)\log_2 q(c_j)) - \alpha(c_j)\log_2 treesize(c_j)) \quad (18)$$

where

- N = number of classes of the problem;
- $p(c_j)$ = probability (relative frequency) of class c_j;
- $q(c_j)$ = 1 - $p(c_j)$ = probability of not belong to class c_j;

- $treesize(c_j) = 1 +$ number of descendant classes of class c_j (1 is added to represent c_j itself);
- $\alpha(c_j) = 0$, if $p(c_j) = 0$ or a user-defined constant (default = 1), otherwise.

This new entropy formula is now composed of three parts: the uncertainty in the choice of the classes ($(p(c_j) \log_2 p(c_j)) + (q(c_j) \log_2 q(c_j))$ and the uncertainty in the specificity of the classes ($\log_2 treesize(c_j)$), which means transmitting the size of the class hierarchy under the class in question (Clare, 2003). The final output of HMC4.5, for a given instance $\mathbf{x}_i$, is a vector of true values $\mathbf{v}_i$. If the value of $v_{i,j}$ is above a given threshold θ, the instance is assigned to the class c_j.

The HMC4.5 method used can be freely obtained at http://www.aber.ac.uk/en/cs/research/cb/dss/c45modifications/.

3.4. Clus-HMC

This global HMC method builds decision trees using a framework named Predictive Clustering Trees (PCTs) (Blockeel *et al.*, 1998), where decision trees are constructed as a cluster hierarchy. The root node contains all the training instances, and is recursively partitioned in small clusters as the decision tree is traversed towards the leaves. The PCTs can be applied both to clustering and classification tasks, and they are built using an algorithm similar to others used for decision tree induction, such as CART (Classification and Regression Trees) (Breiman *et al.*, 1984) or C4.5.

The method works as follows. Initially, the labels of the instances are represented as boolean vectors $\mathbf{v}$, where the j^{th} position of a class vector of an instance receives the value 1 if the instance belongs to class c_j, and 0 otherwise. The vector that contains the arithmetic mean, or prototype, of a set of vectors V, denoted by $\overline{\mathbf{v}}$, has, as its j^{th} element, the proportion of instances of the set that belongs to the class c_j. The variance of a set of instances X, shown in Equation (19), is given by the mean square distance between each class vector $\mathbf{v}_i$ of each instance $\mathbf{x}_i$ and the prototype class vector $\overline{\mathbf{v}}$. The prototype of V is presented in Equation (20).

$$Var(X) = \frac{\sum_i d(\mathbf{v}_i, \overline{\mathbf{v}})^2}{|X|} \tag{19}$$

$$\overline{\mathbf{v}} = \frac{\sum_{\mathbf{v}_i \in V} \mathbf{v}_i}{|V|} \tag{20}$$

As classes at deeper levels represent more specific information than classes at higher levels, the weighted Euclidean distance between the classes is used to consider the depth of the classes in the hierarchy. Equation (21) shows the calculation of this distance, where $v_{i,j}$ is the j^{th} element of the class vector $\mathbf{v}_i$ of a given instance $\mathbf{x}_i$, and the weights $w(c)$ decrease as the depth of the classes in the hierarchy increases ($w(c) = w_0^{depth(c)}$, with $0 < w_0 < 1$). The heuristic used to choose the best test to be placed in a tree node is the maximization of the variance reduction of a set of instances (Vens *et al.*, 2008).

$$d(\mathbf{v}_1, \mathbf{v}_2) = \sqrt{\sum_j w(c_j) \times (v_{1,j} - v_{2,j})^2} \tag{21}$$

Different from a common decision tree, in a PCT the leaf nodes store the mean of the instances' class vector covered by that leaf, i.e., the prototype of a group of instances ($\overline{\mathbf{v}}$). The proportion of instances in a leaf that belongs to a class c_j is denoted by $\overline{\mathbf{v}}_j$, and can be interpreted as the probability of an instance being assigned to class c_j. When an instance reaches a leaf node, if the value of $\overline{\mathbf{v}}_j$ is above a given threshold θ_j, the instance is assigned to class c_j. In order to ensure the integrity of the hierarchical structure, i.e., to ensure that when a class is predicted its superclasses are also predicted, the threshold values must be chosen in a way that $\theta_j \leqslant t_k$ always that $c_j \leqslant_h c_k$, i.e., always that c_j is a superclass of c_k (Vens *et al.*, 2008).

The Clus-HMC program used was implemented in the work of Vens *et al.* (2008), and is freely available at http://www.cs.kuleuven.be/~dtai/clus/.

4. EXPERIMENTS AND DISCUSSION

We have previously presented a set of measures used to evaluate hierarchical multi-label classification problems, and four methods used to perform the HMC task. This section evaluates these four methods, namely

HMC-BR, HMC-LP, HMC4.5 and Clus-HMC, using a set of ten hierarchy-based and distance-based evaluation measures.

In the HMC-BR and HMC-LP methods, the decision tree induction algorithm C4.5 was used as the base classifier. Besides being the most used decision tree induction algorithm, it is also the base algorithm modified to generate the HMC4.5 method. The methods were implemented using the R language (R Development Core Team, 2008), and the C4.5 algorithm used was the implementation of the RWeka package (Hornik *et al.*, 2009) with its default parameter values. The HMC4.5 and Clus-HMC methods were used with their default parameter values for all datasets. As the final classification of the global methods are vectors with real values indicating the probability of the instances to belong to each of the classes, a threshold value equal to 0.4 was used to define the membership to the classes, so that only those classes with a probability higher than or equal to 0.4 are assigned to an instance. The threshold value 0.4 was chosen based on previous experiments with different thresholds, showing the best results in the majority of the datasets. When applying the thresholds, we made sure not to generate predictions inconsistent with the class hierarchy. Unlike the global methods, the vectors of predicted classes of the local methods contain only binary values: 1, if an instance belongs to a given class, and 0 if it does not belong.

In the experiments, the value $Dis_\theta = 2$ was chosen as the acceptable distance between two nodes for the evaluation of the distance-based measures. Thus, if the number of edges (in the class hierarchy) between a predicted class and a true class for a given instance is equal to 2, that prediction will not be counted as a false positive or false negative. On the other hand, if the number of edges between a predicted class and a true class is larger than 2, this distance is counted as either a false positive or a false negative. The value $Dis_\theta = 2$ was also used in the experiments reported by Sun and Lim (2001).

As the basic idea of the evaluation measure is to consider that closer classes in the hierarchy are more similar to each other, the use of $Dis_\theta = 2$ defines that when the distance between a predicted class and a true class is equal to 2 edges, this error should not contribute negatively to the measure value, because it will consider that these two classes are similar. When the distance is larger than 2, the error should contribute negatively to the value of the measure. Also, in our evaluations, we did not use weights associated to the edges of the class hierarchies.

Table 2 lists the selected evaluation measures. We show values of precision and recall separately, as they give a better idea of how each method/measure performs in the different scenarios. We then provide the corresponding F-Measure values and analyse the performances of the methods based on their values.

TABLE 2. Hierarchical Multi-Label Classification measures used in the experiments.

Hierarchy-based Measures	Distance-based Measures
• Hierarchical Loss Function	• Hierarchical Micro Precision
• Hierarchical Precision	• Hierarchical Micro Recall
• Hierarchical Recall	• Hierarchical Macro Precision
• Hierarchical F-Measure	• Hierarchical Macro Recall
	• Hierarchical Micro F-Measure
	• Hierarchical Macro F-Measure

These methods and measures were evaluated considering datasets with different hierarchical and multi-label characteristics. For such, we generated 12 variations of a real-world bioinformatics dataset, varying hierarchical and multi-label characteristics of the original dataset, as described in the next subsection.

4.1. Datasets

For the generation of the datasets, an R program was implemented using the HCGene R package (Valentini and Cesa-Bianchi, 2008). The HCGene package implements methods to process and analyse the Gene Ontology (Ashburner *et al.*, 2000) and the FunCat (Ruepp *et al.*, 2004) hierarchical taxonomies in order to support the functional classification of genes. All generated datasets are real subsets of the original data from (Spellman *et al.*, 1998) yeast cell cycle microarray experiment. The datasets generated have 77 attributes and at most 4 hierarchical levels.

The original yeast dataset is hierarchically structured as a tree according to the FunCat schema. It has 506 classes structured in a hierarchy up to six levels deep, with 5645/3893/3653/2116/676/28 instances in each level, and with each instance having until 21 classes assigned to it. Other characteristics are shown in Table 3.

TABLE 3. Characteristics of the original yeast dataset.

N. Attrib.	N. Classes	N. Instances		Avg. N. Instances per Class for each level						Avg. N. Classes per Instance for each level					
		Total	Multi-Label	L1	L2	L3	L4	L5	L6	L1	L2	L3	L4	L5	L6
77	506	5645	3541	313.61	48.66	20.29	14.49	8.66	7	2.07	2.20	1.66	0.83	0.18	0.03

When varying a multi-label or a hierarchical characteristic of a dataset, we try to keep the others unchanged as much as possible, to isolate the effects of that change. However, this is a rather difficult task, since we are working with a real dataset. As an example, suppose we want to vary the number of classes assigned to an instance while keeping a minimum label cardinality, so that we have enough instances for training. It is difficult to extract a subset of a real dataset, keeping the majority of instances assigned to more than four or five classes and still keeping a sufficient number of instances with the same label cardinalities for training.

As another example of the difficulty in generating many datasets from a specific real-world one, consider the task of generating a dataset with 10% of its instances assigned to a class A, 20% of its instances assigned to a class B, 30% of its instances assigned to a class C, and 40% of its instances assigned to a class D. Let us consider also that we need to: (i) keep the label cardinality of the instances unchanged, (ii) ensure that all instances are classified in the last hierarchy level in order to obtain a complete level balance, and (iii) ensure that all internal nodes have a specific number of child nodes. Hence, although we preserve as much as possible the original characteristics of the dataset, sometimes varying one characteristic inevitably changes another. Despite this difficulty, overall, the generated datasets are useful to show how different hierarchical and multi-label variations influence the different classification methods and evaluation measures.

Table 4 shows the variations investigated and the datasets generated. The first two characteristics varied (shown in the first column of the table) are multi-label characteristics, whilst the last two are hierarchical characteristics. The first dataset variation was the percentage of multi-label instances in the dataset. It is important to study the effect of this variation because, in general, the larger the number of instances having more than one class labels, the more difficult the multi-label classification problem is. Four values were considered for this characteristic: 20%, 40%, 60% and 80% of multi-label instances.

TABLE 4. Variations performed in the datasets generated.

Characteristic varied	Dataset	Variation values	Number of instances per level	Equal characteristics in all datasets
Percentage of multi-label instances	Data1	20%	858 / 818 / 572 / 376	• Maximum of 4 hierarchical levels • Minimum cardinality of leaf nodes equals to 30 • Number of child nodes per parent between 1 and 7
	Data2	40%	909 / 874 / 654 / 453	
	Data3	60%	940 / 919 / 719 / 524	
	Data4	80%	960 / 945 / 784 / 578	
Number of classes assigned to the majority of the instances	Data5	1 to 2	3422 / 2302 / 2031 / 1096	• Maximum of 4 hierarchical levels • Minimum cardinality of leaf nodes equals to 10 • Number of child nodes per parent between 1 and 10
	Data6	2 to 4	2291 / 2291 / 2249 / 1540	
	Data7	4 to 6	692 / 692 / 686 / 545	
Unbalance of the hierarchy	Data8	1 level	2869 / 2869 / 2869 / 1559	• Maximum of 4 hierarchical levels • Minimum cardinality of leaf nodes equals to 30 • Number of child nodes per parent between 1 and 8
	Data9	2 levels	3261 / 3261 / 1559 / 1559	
	Data10	3 levels	4405 / 1559 / 1559 / 1559	
Maximum number of child nodes of each internal node	Data11	5	3403 / 3403 / 1979 / 1035	• Maximum of 4 hierarchical levels • Minimum cardinality of leaf nodes equals to 30 • Percentage of multi-label instances around 50% and 60%
	Data12	8	3391 / 3391 / 3108 / 1334	

To generate the datasets *Data1*, *Data2*, *Data3* and *Data4*, all instances from the original dataset, which respected the constraints shown in Table 4 were selected. Table 5 shows, for each of the datasets, the distribution of the classes over the instances.

TABLE 5. Distribution of classes over the instances when varying the percentage of multi-label instances. For each dataset, the first row shows the number of instances assigned to the number of classes shown in the second row.

Data1	661	129	53	11	2	1	1	
	1	2	3	4	5	6	7	
Data2	521	241	106	28	9	2	1	1
	1	2	3	4	5	6	7	8
Data3	355	360	161	45	15	2	1	1
	1	2	3	4	5	6	7	8
Data4	178	485	209	64	18	4	1	1
	1	2	3	4	5	6	7	8

The second characteristic varied was the number of classes assigned to the majority of the instances. Three alternative values were considered: datasets with the majority of their instances assigned to 1 to 2, 2 to 4, and 4 to 6 classes. Varying the values of this characteristic is important because, in principle, different numbers of classes per instance will result in different values of predictive performance measures, such as hierarchical precision and hierarchical recall. More precisely, as the number of classes per instance increases, one would expect precision to increase (there is a higher chance that a predicted class is really a true class simply because the instance has more true classes) and recall to decrease (there is a higher chance that a true class is not predicted only because there are more true classes).

To generate the *Data5* dataset, all instances with 1 or 2 labels, respecting the constraints described in Table 4, were selected from the original dataset (2521 instances with 1 class and 901 instances with 2 classes). Due to the difficulties previously described for the generation of the datasets, it was not possible to generate datasets with all instances assigned to 2 to 4 classes and to 4 to 6 classes respecting all constraints described in Table 4. All the instances were randomly selected from the original dataset. Table 6 shows, for each dataset, the distribution of the classes over the instances. It is possible to see that, as desired, we varied the multi-label characteristic for all datasets.

TABLE 6. Distribution of classes over the instances when varying the number of classes assigned to the majority of the instances. For each dataset, the first row shows the number of instances assigned to the number of classes shown in the second row.

Data5	2521	901				
	1	2				
Data6	91	1155	727	218		
	1	2	3	4		
Data7	18	34	132	287	141	80
	1	2	3	4	5	6

Different levels of hierarchy unbalance were also tested, using 1, 2 and 3 levels of unbalance. This characteristic was varied because it can have a large influence on the performances of the classification methods. It is expected that as the hierarchy becomes more unbalanced, the divide and conquer mechanism of the local methods is affected. As the global methods deal with all classes at the same time, it is also interesting to see how unbalanced hierarchies influence their predictive performance.

To vary the unbalance of the hierarchies, complete trees were generated, where each instance is classified into a leaf node. As can be seen in the fourth column of Table 4, as *Data8* is one level unbalanced, all the instances reach the third level, and just some instances reach the fourth hierarchical level. The same happens for the *Data9* and *Data10* datasets. Again, all instances from the original dataset respecting the constraints shown in Table 4 were selected. Table 7 shows, for each dataset, the distribution of the classes over the instances.

TABLE 7. Distribution of classes over the instances when varying the unbalance of the hierarchy. For each dataset, the first row shows the number of instances assigned to the number of classes shown in the second row.

Data8	1421	786	420	146	77	14	4	1		
	1	2	4	4	5	6	7	10		
Data9	1496	873	507	231	93	43	13	3	1	1
	1	2	3	4	5	6	7	8	9	11
Data10	2803	905	393	181	89	27	5	2		
	1	2	3	4	5	6	7	8		

Finally, datasets with a variation in the maximum number of children per internal node were generated, to have the majority of their nodes with 5 and 8 children. The increase in the number of children per internal node has a large influence on the number of leaf classes and on the number of multi-label instances, affecting the performance of the methods. It is expected that the classification task becomes more difficult as the number of children is increased, harming the performance of the classification methods. The *Data11* and *Data12* datasets were generated similarly to the other datasets, by selecting all instances from the original dataset respecting the constraints presented in Table 4. Table 8 shows, for each dataset, the distribution of the classes over the instances, and Table 9 shows other characteristics of all the datasets generated.

TABLE 8. Distribution of classes over the instances when varying the maximum number of child nodes per internal node. For each dataset, the first row shows the number of instances assigned to the number of classes shown in the second row.

Data11	1675	953	472	196	69	29	5	3	1	
	1	2	3	4	5	6	7	8	9	
Data12	1612	1236	620	297	97	38	26	2	2	1
	1	2	3	4	5	6	7	8	9	10

TABLE 9. Characteristics of the generated datasets, varying the number of multi-label instances and the characteristics of the hierarchy.

Dataset	N. Attrib.	N. Classes	N. Instances		Avg. N. Instances per Class				Avg. N. Classes per Instance			
			Total	Multi-Label	L1	L2	L3	L4	L1	L2	L3	L4
Datasets varying the percentages of multi-label instances												
Data1	77	135	858	197	57.20	20.97	13.61	9.64	1.20	1.21	0.90	0.72
Data2	77	139	909	388	60.60	22.41	14.53	11.32	1.43	1.49	1.13	0.92
Data3	77	139	940	585	62.66	23.56	15.97	13.10	1.65	1.76	1.29	1.02
Data4	77	139	960	782	64.00	24.23	17.42	14.45	1.85	2.01	1.45	1.02
Datasets varying the numbers of classes assigned per instance												
Data5	77	163	3422	901	228.13	50.04	29.86	32.23	1.21	1.34	1.15	0.77
Data6	77	235	2291	2200	134.76	40.19	22.04	26.10	2.14	2.32	2.01	1.16
Data7	77	236	692	674	40.70	12.14	6.66	9.23	3.22	3.68	3.10	1.65
Datasets varying the levels of unbalance in the hierarchies												
Data8	77	181	2869	1448	204.92	89.65	45.53	21.65	1.53	1.66	1.78	1.41
Data9	77	159	3261	1765	232.92	101.90	38.02	21.65	1.70	1.91	1.26	1.41
Data10	77	147	4405	1602	314.64	77.95	38.02	21.65	1.54	1.20	1.26	1.41
Datasets varying the maximum numbers of children per internal node												
Data11	77	198	3403	1728	226.86	83.00	28.27	14.37	1.49	1.71	1.23	1.02
Data12	77	206	3931	2319	262.06	89.34	41.44	18.52	1.59	1.85	1.47	0.60

Observing the statistics for the datasets, it is possible to see that there are some redundancies between them. It is possible to see that as the number of classes assigned to the majority of the instances is increased, the percentage of multi-label instances is also increased. As the number of classes in datasets *Data5*, *Data6* and *Data7* is increased from 1 to 2 until 4 to 6 classes per instance, the multi-label percentages of these datasets are 26.32%(1 to 2), 96.02% (2 to 4) and 97.39% (4 to 6). These redundancies occur only when the percentage of multi-label instances is either small (26.32%) or too large (96.02% and 97.39%). The percentage variations adopted in the *Data1*, *Data2*, *Data3* and *Data4* datasets range from 20%, 40%, 60% and 80%, allowing better insights regarding the performances of the methods and measures with more variations.

It is also possible to see a redundancy between the variations concerning the unbalance of the hierarchies and the maximum number of child nodes for each internal node. The multi-label percentages in the *Data8*, *Data9* and *Data10* datasets are 50.47%, 54.12% and 36.36%, respectively, while the multi-label percentages in the *Data11* and *Data12* datasets are, respectively, 50.77% and 58.99%. Despite these redundancies, the number of child nodes for each node in the *Data8*, *Data9* and *Data10* datasets did not suffer much variation, while in the *Data11* and *Data12* datasets this variation was higher.

The next sections present the results for all dataset variations considering different measures and methods. There are a few observations that hold for all graphs plotted from now on. In Figures 6, 7, 8 and 9, the values of the predictive performance evaluation measures are divided into two types of graphs: on the left column we always report precision-related measures plus the hierarchical loss, and on the right column are reported the recall-related measures. These figures are also divided into four parts – denoted (a),(b),(c),(d) – each of them representing results for one specific method. The reader may also note that the values reported may be considered considerably low for standard classification applications. However, in hierarchical multi-label datasets, this is quite common, given the difficulty of the tasks being solved.

Figures 10, 11, 12 and 13 present the values considering the F-Measure variations. Each figure is divided in four parts – (a),(b),(c),(d) – each of them representing the F-Measure results for a specific method.

It is important to point out that these graphs can be read in many different ways. Here, we focus on analysing the behaviour of the evaluation measures when used with different datasets' characteristics and different HMC methods. We analyse how the performance of different HMC methods is affected by the use of different evaluation measures or dataset characteristics, and we compare the performances of the global and local classification approaches based on the measures investigated.

We also compared the evaluation measures considering their degrees of consistency, discriminancy and indifferency, as suggested by Huang and Ling (2005). The results of these comparisons are shown in Tables 10, 11 and 12. Due to space restrictions, in the tables we represented the methods Clus-HMC, HMC4.5, HMC-BR and HMC-LP as, respectively, Clus, HC4.5, BR and LP.

4.2. Results varying the percentage of multi-label instances

Figure 6 presents the results obtained when varying the percentage of multi-label instances. It is suggested that, as the percentage of multi-label instances grows, there is a tendency to have a larger number of instances per class at each level, which can increase the confidence of the predictions made. On the other hand, the number of classes per instance also grows, and this might affect precision and recall, depending on how the methods being evaluated work.

Let us now analyse the behaviour of the precision measures as the number of multi-label instances grows. The first observation is that the H-Macro Precision presents almost no variation for all HMC methods, being stable to changes in the number of multi-label instances. Remember that the H-Macro Precision refers to the predictive performance per class. As the distribution of instances per class did not change significantly, so did not the values for the H-Macro Precision (see Table 9).

Regarding the other two measures directly related with precision, the values of H-Micro Precision do not vary substantially as the number of multi-label instances grows, being even constant for the Clus-HMC. Although Hierarchical Micro Precision and Hierarchical Precision present the same behaviour in most cases (see the graphs in Figure 6), it is interesting to notice that the values of H-Precision are always superior to those of H-Macro and H-Micro Precision. This is due to the differences between these evaluation measures. While the Hierarchical Precision measure just counts the number of correct and incorrect predicted classes, the Hierarchical Micro Precision takes into account the distance between the true and the predicted classes. If this distance is higher than 2, the misclassification gives a negative contribution to the measure, which causes its value to decrease.

The only measure presented here that is not based on standard precision or recall is the Hierarchical Loss

Function (it is also the only measure that, the lower its value, the better its result). According to this measure, HMC-BR obtained less errors than the other methods, except in the dataset with 60% of multi-label instances, where HMC-LP has the best performance. These best results of the local methods, however, have to be considered together with the characteristics of the evaluation measure used. The Hierarchical Loss Function does not take into account error propagations, i.e., if a misclassification occurs in a node of the class hierarchy, no additional penalization should be given to misclassifications in the subtree of this node. This kind of evaluation may favour methods based on the local approach, as they can make more mistakes in the first hierarchical levels, while the global methods make their predictions directly on the more specific nodes.

Concerning the values of recall, which appear in the graphs on the right column of Figure 6, the measures are more consistent, frequently following the same pattern, although one of the measures may vary slightly more than another. In addition, they also show that the methods cope well with the increasing in the number of multi-label instances, once the values of recall do not abruptly decrease as the number of multi-label instances grows, with exception of the HMC-BR method. For Clus-HMC, the recall increases together with the number of multi-label instances, which seems to indicate that the method has a better coverage for datasets with a higher label cardinality.

Regarding the four methods tested, HMC-LP was the best method according to the Hierarchical Micro Precision, but presented lower recall than the global methods. Hierarchical Precision considered HMC-LP the best method for the datasets with 20% and 80% of multi-label instances, being HMC-BR the best method in the other two cases.

Considering the analysis with the F-Measure variations (Figure 10), we can see that the global methods obtained better results compared with the local methods. The performance of HMC-BR indicates that one binary classifier per node is not a good strategy when dealing with datasets with a great number of multi-label instances. The fact that HMC-BR does not consider label correlations may have contributed to decrease its predictive power. The analysis of the F-Measure curves for HMC-LP, however, shows that this method could deal well with a great number of multi-label instances. Although the label-powerset procedure creates a high number of classes, it seems that the label correlations could be maintained, and together with the high number of examples for training, led to good results if compared to the global methods. This suggests that label-powerset can be a good classification alternative.

4.3. Results varying the number of classes assigned to the majority of the instances

Figure 7 shows the results after varying the number of classes per instance in the dataset. Again, as the number of classes grows, we might have more instances per class to learn from. However, for this variation, notice that the number of total instances presented a high variation from one experiment to another. While the dataset where the number of classes varies from 1 to 2 has 3422 examples, when assigning 4 to 6 classes per example, we end up with 692 examples.

Once more, in these experiments, the behaviour of H-Macro Precision did not suffer a large variation. The results also suggest the H-Loss seems to favour local methods when they are compared with global methods. According to H-Loss, global methods are less appropriate to datasets with the characteristics presented in this study (see Table 9), as its values for HMC4.5 and Clus-HMC are always worse than those obtained by HMC-BR and HMC-LP. As previously explained, this happens because local methods usually make more mistakes in the first levels of the hierarchy. As H-Loss does not count errors in the subclasses of these erroneously predicted classes, and the first levels have fewer classes, the H-Loss value can be lower for these methods. Global methods, on the other hand, classify examples directly on their more specific nodes, and hence their errors are more concentrated in the deeper levels of the hierarchy. As the number of classes increases with the depth of the hierarchy, H-Loss will count more errors for the global methods, since the number of classification errors made by these methods in the last levels is higher than in the first levels.

For all other measures based on precision, the performances of global methods tend to improve as the number of classes assigned to an instance increases, which is not always true for local methods. It is possible to say that the higher number of classes may have favoured the global methods, and harmed the performance of local methods. The latter occurred because, as the top-down strategy used by local methods propagates errors to the hierarchy, the more classes are assigned to each example, the more errors are propagated.

Although the performances of global methods improve with the number of classes, in general, according to the Hierarchical Micro Precision and Hierarchical Precision, the local methods are still better overall. The HMC-BR method is better than all methods regarding Hierarchical Precision, and only loses to HMC-LP, according to

Hierarchical Micro Precision, when 2 to 4 classes are assigned to the majority of instances. In contrast, all the recall measures (right column of Figure 7) always grow as the number of classes increases. Additionally, global methods always get better recall values than local methods as the number of classes is increased to 4 to 6 classes per instance. Again, we believe this is a consequence of a what seems to be characteristic of the global methods, which are better in classifying the classes in the last hierarchical levels, getting a better coverage of the instances. The local methods, on contrary, seem to give preference to predictions in the first levels, not covering as many instances but being more precise.

Different from what we thought when we generated these datasets, the recall values did not decrease as the number of classes assigned to the majority of instances increased. One would expect a decrease in the recall values, once, with more true classes, there is a higher chance that a true class is not predicted. However, as already said, we can see that all recall values increased with the number of classes, which shows that the methods can deal well with a higher number of classes. Additionally, if we compare the increases of the recall values regarding the global and local methods, we can see that the increase of the recall values for the global methods is more accentuated, which again suggests that global methods are better in predicting classes at deeper hierarchical levels.

Although the local methods achieved the best performances considering the Precision measures, if we analyse the performances of the methods considering the F-Measure variations, it is possible to see that the global methods have better performances than the local methods as the number of classes increases. According to the graphs shown in Figure 11, the best performances of the global methods are more evident when the dataset has 4 to 6 classes assigned to each instance, whilst local methods achieved better performances when the dataset has 1 to 2 classes assigned to each instance. These results suggest that global methods are more suitable for datasets where instances have a larger number of labels.

4.4. Results varying the balance of the hierarchy

Figure 8 shows the results when the hierarchy balance of the dataset is varied. Let us start with the H-Precision. For global methods, it presents the same behaviour, not varying much from one to two levels, but decreasing when we get to three levels of unbalance. These results may be due to a peculiarity of the FunCat taxonomy, which has one specific class (*Unclassified proteins*) in the first level that has no subclasses. In the datasets generated with one and two levels of unbalance, this specific class does not appear in the hierarchy, since it does not have any subclasses. When the hierarchy is three levels unbalanced, this class comes back to the hierarchy, and as a consequence the classifiers make more mistakes in this specific class. As the HMC-BR method creates a specific binary classifier for this class, this may have helped its performance in this specific case. This behaviour of the classifiers suggests that, even if the hierarchy is unbalanced, having fewer classes, the classification performance can be worse than when the hierarchy is completely balanced, having more classes. We can observe this behaviour when the classes of the first level have many instances assigned to them, which can be very frequent in real datasets, as the one we are working with here.

Analyzing the results obtained by the Hierarchical Micro Precision measure, it is possible to observe that the performances of the local methods tend to increase as the level of unbalance of the hierarchy increases. This happens because the classification process of the local methods starts at the root node, and as the hierarchy becomes more unbalanced, less errors are propagated to deeper levels. The same behaviour cannot be always observed in the global methods, because their predictions are made directly in the more specific nodes. As the parent classes of predicted classes are also considered as predicted classes, the misclassifications of the global methods are more uniformly distributed across all levels of the classification hierarchy. When the evaluation considers the Hierarchical Macro Precision, it is possible to observe that the performance of both local and global methods tend to increase as the level of unbalance of the hierarchy increases, and the best results were obtained by the HMC4.5 and Clus-HMC methods.

The behaviours of the H-Precision and the Hierarchical Micro and Macro Precision are different because the latter measures consider the distances between the predicted and true classes in the hierarchy. It seems natural that, as the hierarchy becomes more unbalanced, the distances between the true and predicted classes decrease, specially considering that when the hierarchy is three levels unbalanced, the classifiers made more mistakes in the specific FunCat class mentioned above. As the Macro measures are considered per class mean performance measures, the decrease in the number of classes in the hierarchy (as it becomes more unbalanced) may explain why the value of this measure tends to increase. The only exception can be observed in the Clus-HMC method, in which the value of the H-Micro Precision decreased in the hierarchy with three levels unbalanced. This may

be explained by the fact that the Clus-HMC method was the method that made fewer mistakes in the specific class *Unclassified proteins*. So, the distances between the predicted and true classes, in this case, were higher, contributing to the decrease in the value of the measure.

Another factor that contributed to the increase of the values obtained with the distance based measures, as the hierarchy becomes more unbalanced, was the choice of $Dis_\theta = 2$. It seems that this setting can artificially increase a little the precision/recall values, due to the fact that misclassifications involving similar true and predicted classes are not counted as errors.

Observing the Hierarchical Loss, is possible to see that it is well-behaved, and improves as the level of unbalance increases. The behaviour of the H-Loss seems natural because, as the hierarchy becomes more unbalanced, there are fewer classes to be predicted and less misclassifications are made according to this measure. Additionally, if a correct prediction is made to a class without child classes, the prediction process stops. In contrast, if the predicted class has child classes, the prediction process would continue and the classifier could make more mistakes at the deeper levels, contributing to the value of the H-Loss.

The comparison of the results of H-Loss with H-Precision presents interesting insights. The values of both measures decrease as the hierarchies become more unbalanced. It was already mentioned that H-Loss does not count mistakes made in subclasses of wrongly predicted classes. The classifiers increased the number of mistakes in the first hierarchical levels as the hierarchies became more unbalanced. While more mistakes in the first hierarchical levels contribute to decrease the value of H-Loss, the propagation of these mistakes to deeper levels lead to a reduction of the H-Precision value.

The recall measure values show variations regarding local and global methods. While H-Recall has a small decrease for local methods, for the global methods, it first increases with the difference of two levels, decreasing for three levels. The H-Micro Recall increases as the level of unbalance increases for local methods, while in global methods, it first increases and later decreases. The H-Macro Recall is the only measure with a consistent behaviour across all HMC methods. Note that, even with their recall values decreasing in the hierarchy with three levels of unbalance, the global methods still have better recall values than the local methods.

The more accentuated decrease of the values of the Hierarchical Precision and Recall, and Hierarchical Micro Precision and Recall, observed in the Clus-HMC method, again may be explained by the fact that this method committed less errors in the specific class *Unclassified proteins*, being its errors more concentrated in the balanced size of the hierarchy. In this case, if a class in the first level is misclassified, it means that its leaf classes in the last level were misclassified, increasing the distance between the predicted and true classes, and decreasing the values of the distance based measures. The values of H-Precision and H-Recall, of course, also tend to decrease as more mistakes are committed.

Figure 12 shows the comparison of global and local methods considering the F-Measure variations. Note that the increase in the unbalance of the hierarchy contributed to harm the performances of the global methods, specially Clus-HMC, considering the F-Measure and Micro F-Measure. The Macro F-Measure shows that the global methods have a better per class performance as the unbalance of the hierarchy increases, while for local methods, the Macro F-Measure performance remains almost the same. The best results in the more unbalanced hierarchy were obtained by HMC-BR and HMC4.5. Considering Micro F-Measure, it seems that the increase of the H-Micro Precision compensated the decrease of the H-Micro Recall observed for HMC4.5 (Figure 8(c)), resulting in a better F-Measure.

4.5. Results varying the maximum number of child nodes per internal node

Figure 9 shows the results when the number of child nodes per internal node was increased. It is expected that this should affect all methods, because increasing the number of child nodes per internal node increases a lot the number of multi-label instances. As can be seen in Table 9, increasing the number of child nodes increased the number of multi-label instances from 1728 to 2319. Additionally, both methods will have to build more decision boundaries between classes since, for each internal class, there are more classes to decide regarding the classification of an instance. Hence, the values of the measures are expected to decrease, which is confirmed by the graphs of Figure 9. Although the performances of all methods become worse in the dataset with more children per internal node, it is possible to note that the global methods are those that present the best general results according to all measures, except Hierarchical Loss and Hierarchical Precision.

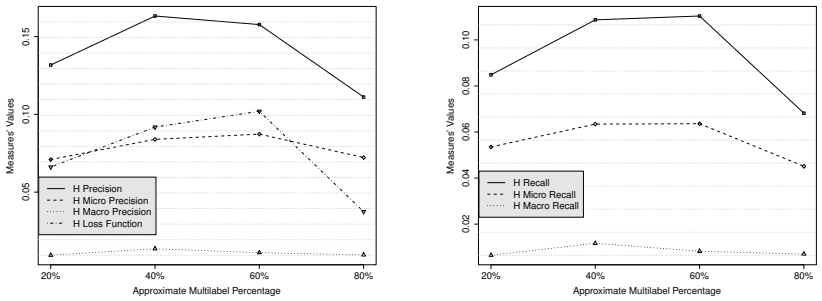
Considering the Hierarchical Macro Precision and Recall measures, the performance of the local methods remains almost the same as the maximum number of children per internal node increases. Looking at the results obtained by the global methods, it is possible to observe that their performances tend to decrease with the

increase of the maximum number of child nodes according to these two evaluation measures. These results show that the global methods are better than the local methods for hierarchies with a high number of child nodes per each internal node. As shown in Figure 9, the performances of the local methods considering these measures are always very poor for both variations (five and eight children). The global methods perform better than the local methods when the hierarchy has five child nodes per each internal node.

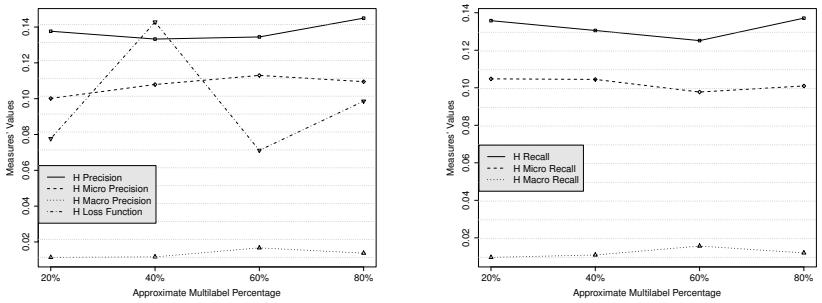
The results with the Hierarchical Micro Precision show that the best results were obtained by the Clus-HMC method in the dataset with a maximum of five children per node. It can also be observed that while the performance of HMC-LP was slightly increased when the maximum number of child nodes increased from five to eight, all other methods had a slightly decreasing performance. This is clearer for the global methods. As the maximum number of children per internal node increases, the number of leaf classes is also increased.

Just like could be observed in the datasets with different levels of unbalance in the hierarchies, the best results when the evaluation was carried out using Hierarchical Macro Precision and all the Hierarchical Recall measures were obtained by the global methods. Again, the errors made by these methods were more distributed over all classes of the hierarchy, which may have increased the values of the macro measures. It seems that the variations in the number of child nodes had a larger impact on the local methods, which performed worst in the majority of cases. This larger influence can be due to the divide-and-conquer mechanism used by the local methods during their classification process.

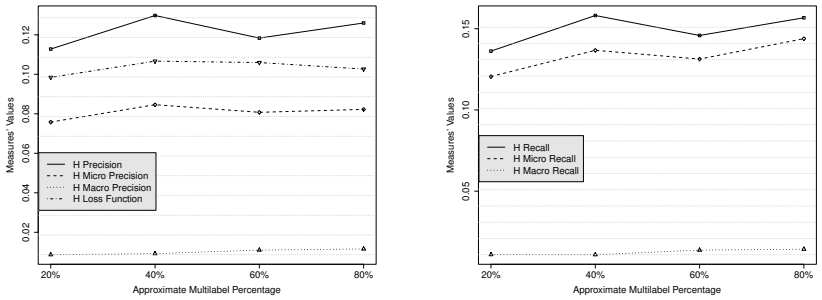
The comparison of the methods considering the F-Measure variations (Figure 13) shows that the best performances, for the majority of the measures, were obtained by the global methods. Although all methods had a decrease in the classification performance, the global methods in general still present better results.



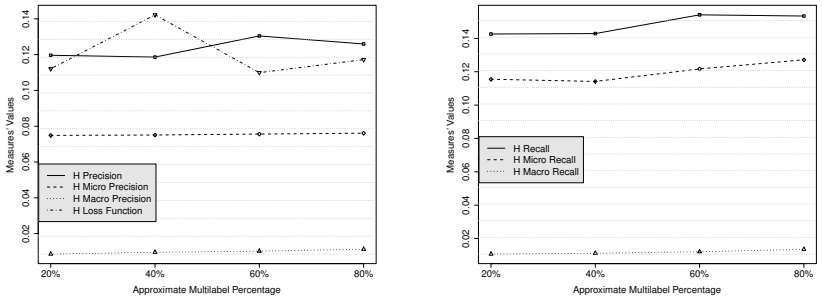
(a) HMC-BR method



(b) HMC-LP method

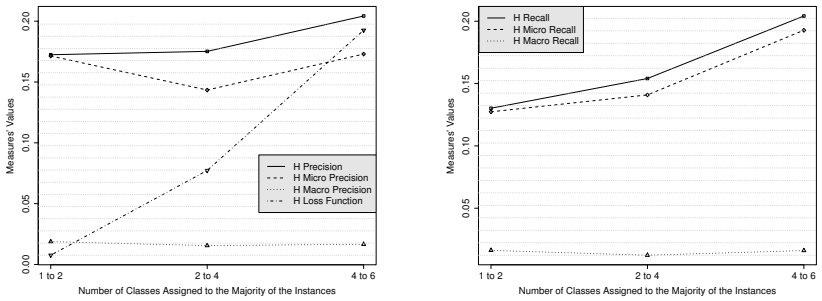


(c) HMC4.5 method

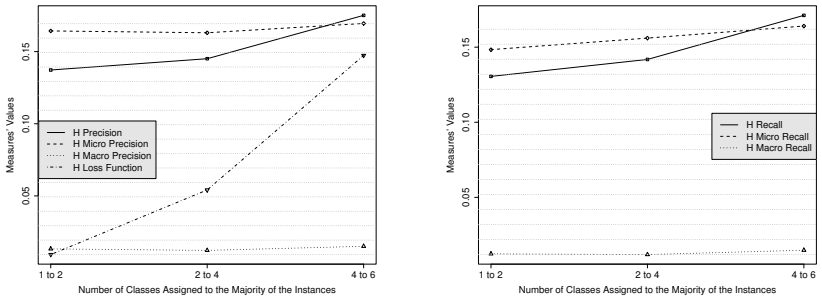


(d) Clus-HMC method

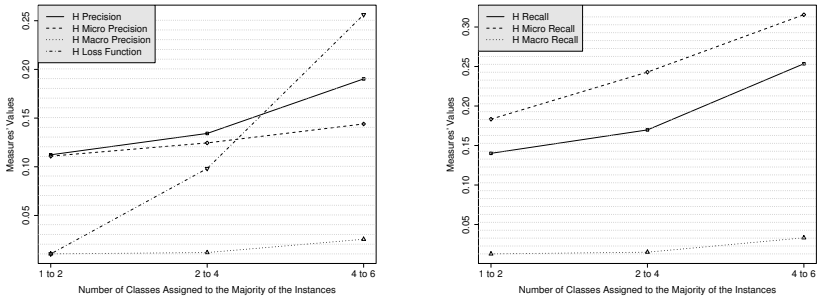
FIGURE 6. Results for different evaluation measures varying the percentage of multi-label instances



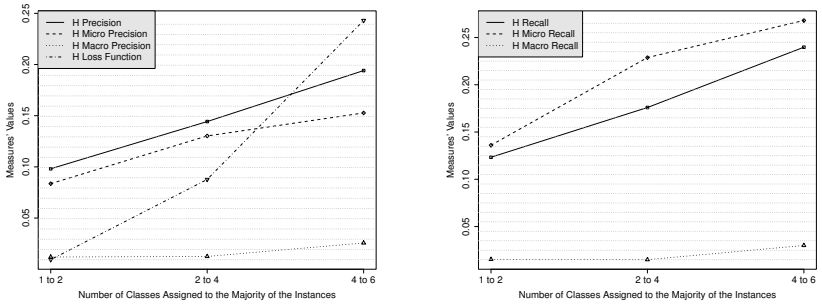
(a) HMC-BR method



(b) HMC-LP method

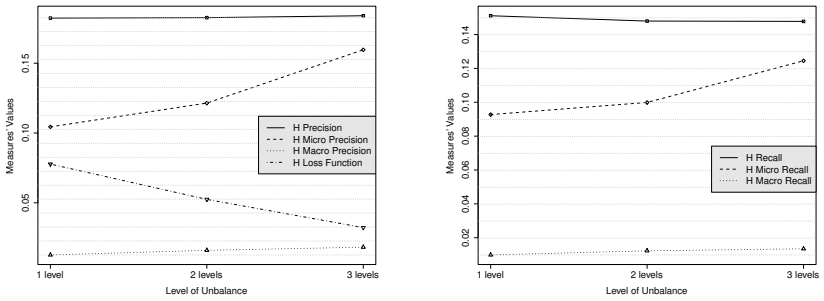


(c) HMC4.5 method

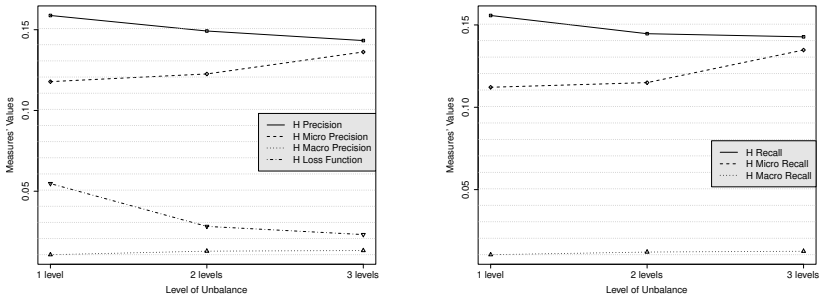


(d) Clus-HMC method

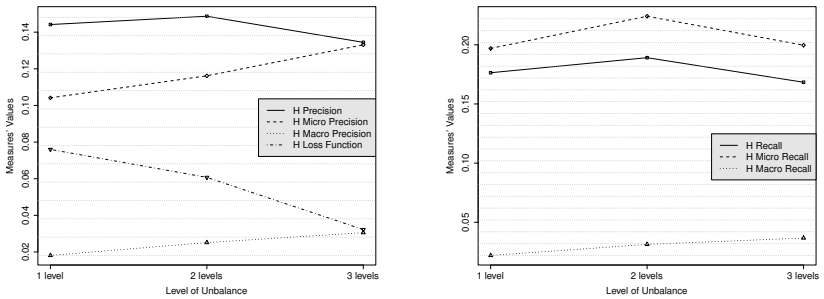
FIGURE 7. Results for different evaluation measures varying the number of classes assigned to the majority of the instances



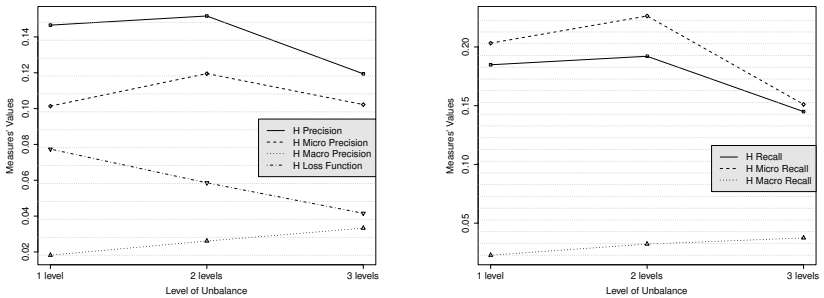
(a) HMC-BR method



(b) HMC-LP method

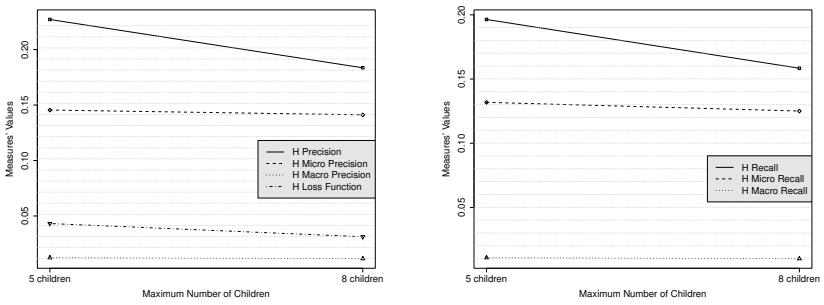


(c) HMC4.5 method

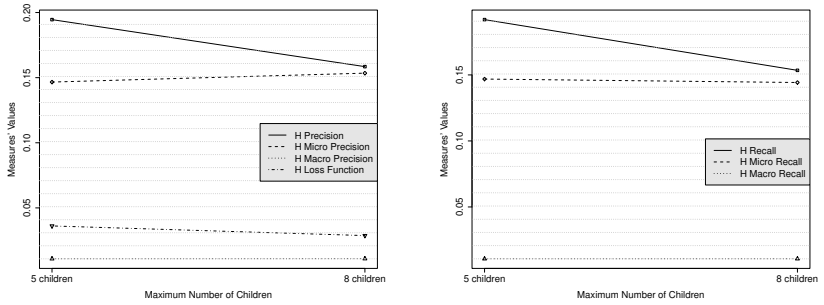


(d) Clus-HMC method

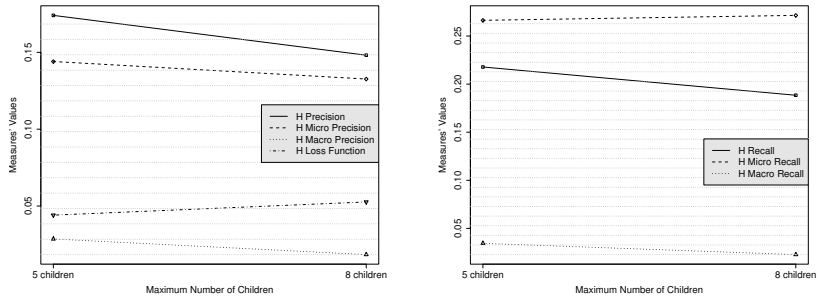
FIGURE 8. Results for different evaluation measures varying the unbalance of the hierarchy



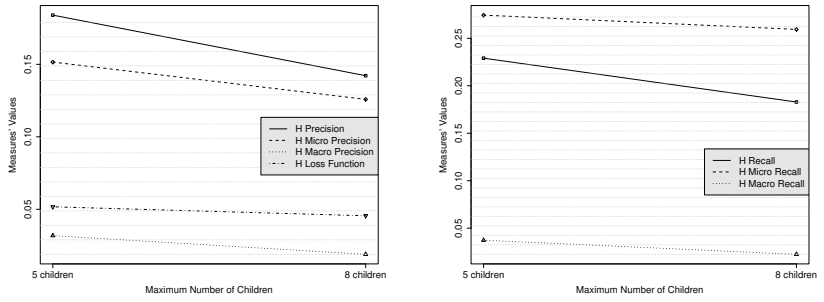
(a) HMC-BR method



(b) HMC-LP method

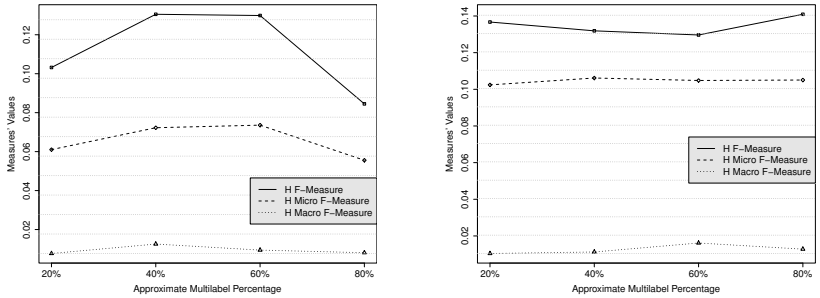


(c) HMC4.5 method



(d) Clus-HMC method

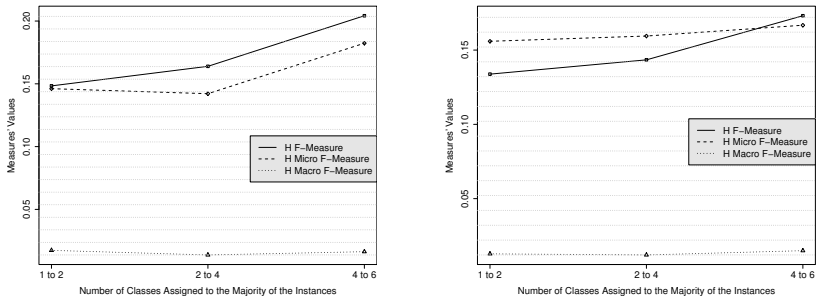
FIGURE 9. Results for different evaluation measures varying the maximum number of child nodes of each internal node



(a) HMC-BR method (b) HMC-LP method

(c) Clus-HMC method (d) HMC4.5 method

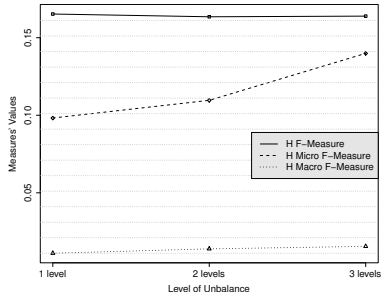
FIGURE 10. Results for different Hierarchical F-Measures varying the percentage of multi-label instances



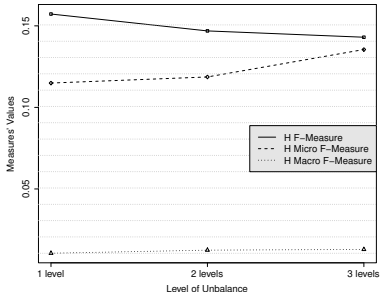
(a) HMC-BR method (b) HMC-LP method

(c) Clus-HMC method (d) HMC4.5 method

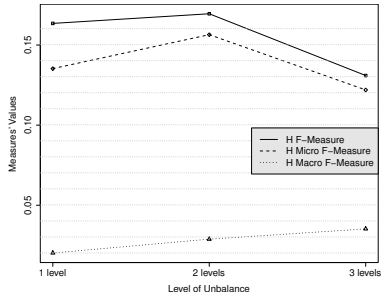
FIGURE 11. Results for different Hierarchical F-Measures varying the number of classes assigned to the majority of the instances



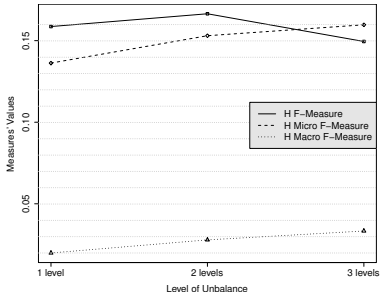
(a) HMC-BR method



(b) HMC-LP method

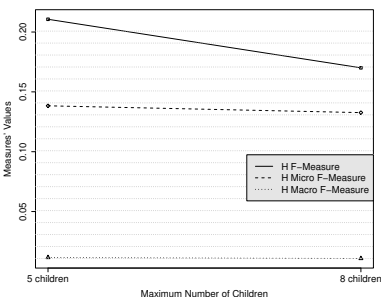


(c) Clus-HMC method

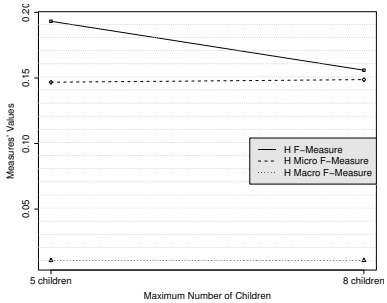


(d) HMC4.5 method

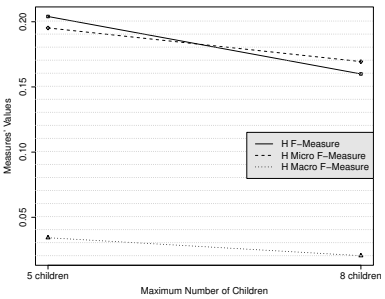
FIGURE 12. Results for different Hierarchical F-Measures varying the unbalance of the hierarchy



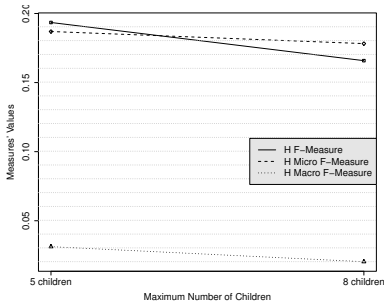
(a) HMC-BR method



(b) HMC-LP method



(c) Clus-HMC method



(d) HMC4.5 method

FIGURE 13. Results for different Hierarchical F-Measures varying the maximum number of child nodes of each internal node

4.6. Comparison of Measures

Tables 10, 11 and 12 show the results obtained when comparing the evaluation measures according to their degrees of consistency, discriminancy and indifferency.

Considering that Ψ is a domain of two evaluation measures f and g. The degrees of consistency, discriminancy and indifferency can be defined as (Huang and Ling, 2005):

- **Degree of consistency:** if $R = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) > g(b)\}$ and $S = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) < g(b)\}$, the degree of consistency of f and g is represented by $\mathbf{C}(0 \leqslant \mathbf{C} \leqslant 1)$, where $\mathbf{C} = \frac{|R|}{|R|+|S|}$;
- **Degree of discriminancy:** if $P = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) = g(b)\}$ and $Q = \{(a,b)|a,b \in \Psi, g(a) > g(b), f(a) = f(b)\}$, the degree of discriminancy for f over g is given by $\mathbf{D} = \frac{|P|}{|Q|}$;
- **Degree of indifferency:** if $V = \{(a,b)|a,b \in \Psi, a \neq b, f(a) = f(b), g(a) = g(b)\}$ and $U = \{(a,b)|a,b \in \Psi, a \neq b\}$, the degree of indifferency for f and g is given by $\mathbf{E} = \frac{|V|}{|U|}$.

We can say that if two measures f and g are consistent to degree $\mathbf{C}$ when evaluating two algorithms A and B, then when the measure f stipulates that A is better than B, there is a probability $\mathbf{C}$ that the measure g will agree. If a measure f is $\mathbf{D}$ times more discriminating than a measure g, then we can say that it is $\mathbf{D}$ times more likely that f can says that A and B are different but g cannot, than vice-versa. So, we can say that a measure f is better than a measure g (f is statistically consistent and more discriminant than g) if and only if $\mathbf{C} > 0.5$ and $\mathbf{D} > 1$ (Huang and Ling, 2005).

In the first column of Tables 10, 11 and 12, g is represented by the measures that are positioned in the left of the / symbol, while f is represented by the measures that are positioned in the right of the / symbol. Hence, we are always trying to see if measure f is better than measure g. As in (Huang and Ling, 2005), we compared the performances of the methods in each cross-validation test set. As we did a 5-fold cross-validation in each dataset, and we have 12 datasets, we performed $12 \times 5 = 60$ comparisons. In the tables, $f+$ means that when comparing algorithm A with algorithm B, A is better than B according to measure f, $-$ means A is worse than B, $=$ means A is the same as B and $\neq$ means that A is not the same as B. As an example, in the first row, second column of Table 10, when comparing the methods Clus-HMC and HMC4.5, there were 15 times where Clus-HMC was better than HMC4.5 both in hP and H-Loss or Clus-HMC was worse than HMC4.5 both in hP and H-Loss, which means that the measures were consistent in 15 of 60 comparisons. Recall that in some comparisons, when calculating the degree of discriminancy, the value of $|Q|$ is equal to 0 resulting in a division by 0 ($\mathbf{D} = \frac{|P|}{|Q|}$). In this case we considered that one measure was $|P|$ times more discriminant than the other.

Let's start our analysis comparing all precision related measures and H-Loss. Analysing the results obtained, we observe that Hierarchical Precision can be considered a better measure than H-Loss. The degrees of consistency between the two measures is above 0.5 in the majority of the comparisons, while their degrees of discriminancy is never higher than 1. Also, if we now consider H-Loss function as g and hP as f (now we verify if hP is better than H-Loss), we see that the degrees of discriminancy between the two measures is now above 1 in all comparisons, which means that hP is statistically consistent and more discriminant than H-Loss in the majority of the comparisons. Recall that the degrees of consistency of two measures are symmetric, which means that the degree of consistency of f and g is the same as the degree of consistency of g and f (Huang and Ling, 2005). It is also interesting to see that we cannot say that there are differences between these two measures when comparing the local methods with each other and the global methods with each other. We can say that hP is better than H-Loss in the comparisons of global *vs.* local methods.

When comparing H-Loss with Hierarchical Micro Precision and Macro Precision, we can see that $\hat{Pr}^{\mu CD}$ is better than H-Loss in all comparisons involving global methods *vs.* local methods. When comparing HMC-BR with HMC-LP, there are no differences between the two measures. In the comparison of Clus-HMC *vs.* HMC4,5, however, we can say that H-Loss is consistent and more discriminant than $\hat{Pr}^{\mu CD}$. Although nothing can be said about which measure is better, H-Loss or $\hat{Pr}^{M_{CD}}$, we can see that $\hat{Pr}^{M_{CD}}$ is much more discriminant than H-Loss in the majority of the comparisons.

Comparing the Hierarchical Precision and Hierarchical Micro and Macro Precisions, we can say that hP is a better measure than $\hat{Pr}^{\mu CD}$, as it is consistent and more discriminant than $\hat{Pr}^{\mu CD}$ in the majority of the comparisons. Remember again that, according to the tables, $\hat{Pr}^{\mu CD}$ is better than hP, but it can be easily shown that hP is more discriminant than $\hat{Pr}^{\mu CD}$. So, if we want to see if hP is better than $\hat{Pr}^{\mu CD}$, the discriminancy values of the comparisons will be higher than 1 in the majority of the cases. When comparing hP with $\hat{Pr}^{M_{CD}}$, we cannot conclude which measures is the best. The hP measure is better than $\hat{Pr}^{M_{CD}}$ when comparing HMC-BR *vs.* HMC-LP and worse than $\hat{Pr}^{M_{CD}}$ when comparing HMC4.5 *vs.* HMC-LP. For the other comparisons involving

these two measures, nothing can be said about what is the best measure. We can only say that $\hat{Pr}^{M_{CD}}$ was more discriminant in the majority of the cases.

Analysing the results of the comparisons between $\hat{Pr}^{\mu CD}$ and $\hat{Pr}^{M_{CD}}$, we can see that the Hierarchical Macro Precision, as well as the Hierarchical Precision, is better than Hierarchical Micro Precision. The $\hat{Pr}^{M_{CD}}$ measure is consistent and more discriminant than $\hat{Pr}^{\mu CD}$ in the majority of the datasets.

Considering the comparisons involving all the recall measures (Table 11), we can see that $\hat{Re}^{\mu CD}$ and $\hat{Re}^{M_{CD}}$ are better than hR in the majority of the comparisons. There are, however, some exceptions. When comparing Clus-HMC *vs.* HMC4.5, we cannot say which measure is better, if hR or $\hat{Re}^{\mu CD}$, because the degree of discriminancy over them is 1. Also, hR is a better measure than $\hat{Re}^{M_{CD}}$ when comparing HMC-BR *vs.* HMC-LP. It is interesting to note the degrees of discriminancy of the micro and macro measures in the Clus-HMC *vs.* HMC4.5 and HMC-BR *vs.* HMC-LP comparisons. While $\hat{Re}^{M_{CD}}$ is more discriminant than hR in the Clus-HMC *vs.* HMC4.5 comparison, $\hat{Re}^{\mu CD}$ is more discriminant than hR in the HMC-BR *vs.* HMC-LP comparison. Also, $\hat{Re}^{M_{CD}}$ is better than $\hat{Re}^{\mu CD}$ in the Clus-HMC *vs.* HMC4.5 comparison, while $\hat{Re}^{\mu CD}$ is better than $\hat{Re}^{M_{CD}}$ in the HMC-BR *vs.* HMC-LP comparison. As global methods are more similar, and macro measures give equal importance to each label, these results may suggest that $\hat{Re}^{M_{CD}}$ can be a more discriminant measure if we want to compare global methods.

Table 12 shows the results of the comparisons involving all the F-Measure variations. It is interesting to see that both Hierarchical Micro and Macro F-Measures are consistent and more discriminant than Hierarchical F-Measure only in the comparisons involving methods from different classification approaches. In the Clus-HMC vs HMC4.5 comparison, we can see that hF is better than $h\hat{F}^{\mu CD}$. Also, in the HMC-BR *vs.* HMC-LP comparison, we can say that hF is better than both $h\hat{F}^{\mu CD}$ and $h\hat{F}^{M_{CD}}$. When comparing $h\hat{F}^{\mu CD}$ and $h\hat{F}^{M_{CD}}$ we can see that Hierarchical Macro F-Measure is better than Hierarchical Micro F-Measure in the majority of the datasets. Again the micro measure is better than the macro measure when comparing the local methods within themselves (HMC-BR *vs.* HMC-LP).

Looking at the degrees of indifferency obtained in all comparisons involving all the measures investigated, Clus-HMC and HMC4.5 methods are the ones that obtained more similar results. This seems natural as the global methods share more similar characteristics, since they build one decision tree to handle all classes and both work with non-mandatory leaf node classification. The local methods, however, present more differences. The HMC-BR method uses one binary classifier for each node while HMC-LP transforms the original problem into a hierarchical single-label one. Additionally, HMC-BR works with non-mandatory leaf node classification while HMC-LP works with mandatory leaf node classification.

TABLE 10. Comparison of precision measures considering degrees of consistency, discriminancy and indifferency.

	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hP+$/HLoss+ or $hP-$/HLoss-	15	27	28	27	28	12
$hP+$/HLoss- or $hP-$/HLoss+	28	17	20	21	21	32
Degree of consistency (**C**)	0.349	0.614	0.583	0.562	0.571	0.273
$hP+/\hat{P}r^{\mu CD}+$ or $hP-/\hat{P}r^{\mu CD}-$	33	40	41	44	37	25
$hP+/\hat{P}r^{\mu CD}-$ or $hP-/\hat{P}r^{\mu CD}+$	9	10	9	9	13	20
Degree of consistency (**C**)	0.786	0.8	0.82	0.83	0.74	0.556
$hP+/\hat{P}r^{M_{CD}}-$ or $hP-/\hat{P}r^{M_{CD}}-$	23	21	25	26	28	43
$hP+/\hat{P}r^{M_{CD}}-$ or $hP-/\hat{P}r^{M_{CD}}+$	22	32	29	32	27	10
Degree of consistency (**C**)	0.511	0.396	0.463	0.449	0.51	0.811
HLoss+$/\hat{P}r^{\mu CD}+$ or HLoss-$/\hat{P}r^{\mu CD}-$	24	27	36	27	36	14
HLoss+$/\hat{P}r^{\mu CD}-$ or HLoss-$/\hat{P}r^{\mu CD}+$	15	18	11	18	11	23
Degree of consistency (**C**)	0.615	0.6	0.766	0.6	0.766	0.378
HLoss+$/\hat{P}r^{M_{CD}}-$ or HLoss-$/\hat{P}r^{M_{CD}}-$	19	18	19	21	14	12
HLoss+$/\hat{P}r^{M_{CD}}-$ or HLoss-$/\hat{P}r^{M_{CD}}+$	25	30	34	29	28	31
Degree of consistency (**C**)	0.432	0.375	0.358	0.42	0.270	0.279
$\hat{P}r^{\mu CD}+/\hat{P}r^{M_{CD}}-$ or $\hat{P}r^{\mu CD}-/\hat{P}r^{M_{CD}}-$	23	31	25	30	28	26
$\hat{P}r^{\mu CD}+/\hat{P}r^{M_{CD}}-$ or $\hat{P}r^{\mu CD}-/\hat{P}r^{M_{CD}}+$	19	22	28	25	25	17
Degree of consistency (**C**)	0.548	0.585	0.472	0.545	0.528	0.605
	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hP =$/HLoss $\neq$	7	5	5	2	3	3
$hP \neq$/HLoss $=$	9	11	7	10	7	13
Degree of discriminancy (**D**)	0.778	0.454	0.714	0.2	0.429	0.231
$hP =/\hat{P}r^{\mu CD} \neq$	6	5	4	2	4	3
$hP \neq/\hat{P}r^{\mu CD} =$	10	5	5	5	6	12
Degree of discriminancy (**D**)	0.6	1	0.8	0.4	0.667	0.25
$hP =/\hat{P}r^{M_{CD}} \neq$	7	5	5	2	4	2
$hP \neq/\hat{P}r^{M_{CD}} =$	7	2	1	0	1	4
Degree of discriminancy (**D**)	1	2.5	5	2	4	0.5
HLoss $=/\hat{P}r^{\mu CD} \neq$	9	10	7	10	7	11
HLoss $\neq/\hat{P}r^{\mu CD} =$	11	4	6	5	5	10
Degree of discriminancy (**D**)	0.818	2.5	1.167	2	1.4	1.1
HLoss $=/\hat{P}r^{M_{CD}} \neq$	8	10	6	10	7	12
HLoss $\neq/\hat{P}r^{M_{CD}} =$	6	1	0	0	0	4
Degree of discriminancy (**D**)	1.333	10	6	10	7	3
$\hat{P}r^{\mu CD} =/\hat{P}r^{M_{CD}} \neq$	10	5	6	5	6	12
$\hat{P}r^{\mu CD} \neq/\hat{P}r^{M_{CD}} =$	6	2	1	0	1	5
Degree of discriminancy (**D**)	1.667	2.5	6	5	6	2.4
	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hP =$/HLoss $=$	1	0	0	0	1	0
Degree of indifferency (**E**)	0.0167	0	0	0	0.1667	0
$hP =/\hat{P}r^{\mu CD} =$	2	0	1	0	0	0
Degree of indifferency (**E**)	0.033	0	0.017	0	0	0
$hP =/\hat{P}r^{M_{CD}} =$	1	0	0	0	0	1
Degree of indifferency (**E**)	0.0167	0	0	0	0	0.0167
HLoss $=/\hat{P}r^{\mu CD} =$	1	1	0	0	1	2
Degree of indifferency (**E**)	0.0167	0.0167	0	0	0.0167	0.033
HLoss $=/\hat{P}r^{M_{CD}} =$	2	1	1	0	1	1
Degree of indifferency (**E**)	0.033	0.0167	0.0167	0	0.0167	0.0167
$\hat{P}r^{\mu CD} =/\hat{P}r^{M_{CD}} =$	2	0	0	0	0	0
Degree of indifferency (**E**)	0.033	0	0	0	0	0

TABLE 11. Comparison of recall measures considering degrees of consistency, discriminancy and indifferency.

	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hR+/\hat{R}e^{\mu CD}+$ or $hR-/\hat{R}e^{\mu CD}-$	33	51	48	53	55	36
$hR+/\hat{R}e^{\mu CD}-$ or $hR-/\hat{R}e^{\mu CD}+$	9	3	5	3	2	14
Degree of consistency (**C**)	0.786	0.944	0.906	0.946	0.965	0.72
$hR+/\hat{R}e^{M_{CD}}-$ or $hR-/\hat{R}e^{M_{CD}}-$	28	48	42	51	49	39
$hR+/\hat{R}e^{M_{CD}}-$ or $hR-/\hat{R}e^{M_{CD}}+$	19	6	13	3	6	6
Degree of consistency (**C**)	0.596	0.889	0.764	0.944	0.891	0.867
$\hat{R}e^{\mu CD}+/\hat{R}e^{M_{CD}}-$ or $\hat{R}e^{\mu CD}-/\hat{R}e^{M_{CD}}-$	30	53	45	51	48	37
$\hat{R}e^{\mu CD}+/\hat{R}e^{M_{CD}}-$ or $\hat{R}e^{\mu CD}-/\hat{R}e^{M_{CD}}+$	16	6	8	7	9	12
Degree of consistency (**C**)	0.652	0.898	0.849	0.879	0.842	0.755
	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hR =/\hat{R}e^{\mu CD} \neq$	8	6	2	4	3	6
$hR \neq/\hat{R}e^{\mu CD} =$	8	0	4	0	0	3
Degree of discriminancy (**D**)	1	6	0.5	4	3	2
$hR =/\hat{R}e^{M_{CD}} \neq$	8	5	3	4	2	7
$hR \neq/\hat{R}e^{M_{CD}} =$	3	0	2	2	2	8
Degree of discriminancy (**D**)	2.67	5	1.5	2	1	0.875
$\hat{R}e^{\mu CD} =/\hat{R}e^{M_{CD}} \neq$	9	0	5	0	0	3
$\hat{R}e^{\mu CD} \neq/\hat{R}e^{M_{CD}} =$	4	1	2	2	3	7
Degree of discriminancy (**D**)	2.25	0	2.5	0	0	0.428
	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hR =/\hat{R}e^{\mu CD} =$	2	0	1	0	0	1
Degree of indifferency (**E**)	0.033	0	0.0167	0	0	0.0167
$hR =/\hat{R}e^{M_{CD}} =$	2	1	0	0	1	0
Degree of indifferency (**E**)	0.033	0.0167	0	0	0.0167	0
$\hat{R}e^{\mu CD} =/\hat{R}e^{M_{CD}} =$	1	0	0	0	0	1
Degree of indifferency (**E**)	0.0167	0	0	0	0	0.0167

TABLE 12. Comparison of F-Measures considering degrees of consistency, discriminancy and indifferency.

	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hF+/\hat{hF}^{\mu CD}+$ or $hF-/\hat{hF}^{\mu CD}-$	34	29	38	23	37	28
$hF+/\hat{hF}^{\mu CD}-$ or $hF-/\hat{hF}^{\mu CD}+$	4	18	4	22	6	18
Degree of consistency (**C**)	0.895	0.617	0.905	0.511	0.860	0.609
$hF+/\hat{hF}^{MCD}-$ or $hF-/\hat{hF}^{MCD}-$	21	24	32	27	32	36
$hF+/\hat{hF}^{MCD}-$ or $hF-/\hat{hF}^{MCD}+$	23	24	11	22	12	4
Degree of consistency (**C**)	0.477	0.5	0.744	0.551	0.727	0.9
$\hat{hF}^{\mu CD}+/\hat{hF}^{MCD}-$ or $\hat{hF}^{\mu CD}-/\hat{hF}^{MCD}-$	26	43	47	46	49	25
$\hat{hF}^{\mu CD}+/\hat{hF}^{MCD}-$ or $\hat{hF}^{\mu CD}-/\hat{hF}^{MCD}+$	16	12	9	8	4	12
Degree of consistency (**C**)	0.619	0.782	0.839	0.852	0.924	0.676
	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hF =/\hat{hF}^{\mu CD} \neq$	8	11	15	10	12	6
$hF \neq/\hat{hF}^{\mu CD} =$	11	2	2	4	3	8
Degree of discriminancy (**D**)	0.727	5.5	7.5	2.5	4	0.75
$hF =/\hat{hF}^{MCD} \neq$	8	9	16	10	14	5
$hF \neq/\hat{hF}^{MCD} =$	5	1	1	0	2	14
Degree of discriminancy (**D**)	1.6	9	16	10	7	0.357
$\hat{hF}^{\mu CD} =/\hat{hF}^{MCD} \neq$	10	2	3	5	5	8
$\hat{hF}^{\mu CD} \neq/\hat{hF}^{MCD} =$	4	3	1	1	2	15
Degree of discriminancy (**D**)	2.5	0.667	3	5	2.5	0.533
	Clus *vs.* HC4.5	Clus *vs.* BR	Clus *vs.* LP	HC4.5 *vs.* BR	HC4.5 *vs.* LP	BR *vs.* LP
$hF =/\hat{hF}^{\mu CD} =$	3	0	1	1	2	0
Degree of indifferency (**E**)	0.05	0	0.0167	0.0167	0.033	0
$hF =/\hat{hF}^{MCD} =$	3	2	0	1	0	1
Degree of indifferency (**E**)	0.05	0.033	0	0.0167	0	0.0167
$\hat{hF}^{\mu CD} =/\hat{hF}^{MCD} =$	4	0	0	0	0	0
Degree of indifferency (**E**)	0.0667	0	0	0	0	0

4.7. Summary and Discussion: Measures versus Methods

This section gives an overview of the behaviour of the measures considering the methods used in the experiments, and gives answers to the five questions asked in Section 1. Statistical tests were also carried out to verify if the differences between the methods were statistically significant, with a confidence level of 95%. The test used was the two-sided non-pairwise Wilcoxon rank-sum test (Hollander and Wolfe, 1999), with the Holm correction (Holm, 1979) for multiple comparisons. The datasets were divided using k-fold cross-validation, with $k = 5$.

Table 13 shows where the algorithms located in the rows were statistically better than the algorithms located in the columns. The numbers in the rows indicate the datasets where some statistically significant difference(s) were detected, and the black circles indicate the measures where these differences were detected. In each row of each cell, the circles refer to evaluation measures in the following order: hierarchical precision, hierarchical micro precision, hierarchical macro precision, hierarchical loss function, hierarchical recall, hierarchical micro recall, hierarchical macro recall, hierarchical f-measure, hierarchical micro f-measure and hierarchical macro f-measure.

TABLE 13. Results of Statistical Tests.

	HMC-BR		HMC-LP		Clus-HMC		HMC4.5	
HMC-BR			4	○○○●○○○○○○	4	○○○●○○○○○○	4	○○○●○○○○○○
			5	●○○○○○○○○○	5	●●●○○○○●●○	5	●●●○○○○●○●
			7	●○○○●●○●○○	10	●●○○●○○○●○	7	○●○○○○○○○○
		——	8	○○●○○○○○○○	12	●●○○●○○○○○	10	●●○○○○○○○○
			10	●●●○○○○○○●	6,8,9,11	●○○○○○○○○○	8,9,11,12	●○○○○○○○○○
			11	●○●○○○○○○○				
			12	●○○○○○○○○○				
HMC-LP	1	○●○○●●○○●○			3	○●●○○○○○○○	2	○○●○○○○○○○
	2	○○○○○●○○●○			5	●●○○○○○●●○	5	●●●○○○○○○○
	3	○○●○○●●○●●		——	10	●●○○○○○○●○	11	●○○○○○○○○○
	4	○○○○○●●○●○			12	●●○○○○○○○○	12	○●○●○○○○○○
	8	○●○○○●○○●○			1,2,4,6,8	○●○○○○○○○○	1,3,4,6,7	○●○○○○○○○○
	5,10,12	○○○○○●○○○○						
Clus-HMC	2	○○○○○●○○○○	6	○○○○○●○○○○			10,11	○○●○○○○○○○
	3,6	○○○○●●○○○○	7	○○●○●●●○○●				
	4	○○○○●●●○○○	9	○○●○●●●●●●		——		
	7,10	○○●○○●●○○●	10	○○●○○●●○○●				
	11	○○●○○●●○●●	8,11,12	○○●○●●●○●●				
	8,9,12	○○●○●●●○●●						
HMC4.5	3	○○○○○●●○●●	3,4,5	○○○○○●○○○○	5	●●○○●●○●●○		
	4	○○○○●●●○●○	6	○○○○○●●○○○	7	○○○○○●○○○○		
	1,5	○○○○○●○○○○	7	○○●○●●●●●●	10	○●○○○●○○●○		
	2,6	○○○○○●○○●○	8	○○●○○●●○●●				——
	7	○○●○●●●○○●	9,10,11,12	○○●○●●●○●●				
	8,12	○○●○●●●○●●						
	9,10,11	○○●○○●●○●●						

A first glance at Table 13 already illustrates how much measures disagree among each other. Each circle represents a different measure. The first three measures evaluate the precision of the methods. Then comes the H-Loss, followed by the three recall evaluation measures. The last three measures are the hierarchical f-measure variations. It must be observed that a black circle followed by a empty circle means that one measure considers the methods statistically different while the other considers they have an equivalent performance. For example, consider the results for Hierarchical Precision (first circle) when compared to Hierarchical Micro Precision (second circle), obtained when comparing HMC-BR with the other methods. In this case, Hierarchical Precision considered HMC-BR better than HMC-LP in five datasets (i.e., datasets 5,7,10, 11 and 12), and better than HMC4.5 in six datasets (datasets 5, 8, 9, 10, 11 and 12). The Hierarchical Micro Precision considered the HMC-BR better than HMC-LP in one dataset, and better than HMC4.5 in three datasets. Hence, looking at Table 13, it is possible to see that the Hierarchical Precision and the Hierarchical Micro Precision disagree according to the statistical tests four times when comparing HMC-BR and HMC-LP and five times when comparing HMC-BR and HMC4.5.

It is also easy to notice that global methods, in general, present better recall than local methods, specially in terms of Hierarchical Micro and Macro Recall. Clus-HMC has better Hierarchical Micro Recall than HMC-BR in all datasets, except 1 and 5, and when comparing it with HMC-LP using this same measure, Clus-HMC is better in the datasets 6 to 12. Note that the Hierarchical Recall and Hierarchical Micro Recall still disagree in many cases. The better performance of the global methods in the recall measures are a indicative that the global classifiers are better than the local ones in detecting the most specific classes of the examples. High values of hierarchical precision measures are a indicative that the classifiers are better in detecting the most general classes of the hierarchy, while high hierarchical recall values are a indicative that the classifiers are better in detecting the most specific classes of the hierarchy.

Comparing the approaches among themselves (HMC4.5 *vs* Clus-HMC and HMC-BR *vs* HMC-LP), it is possible to see that there are more statistically significant differences in the comparisons among the local methods than in the comparisons among the global methods. These results confirm the results obtained when analysing the degrees of indifferency of the measures, which showed that the global methods had similar performances in the datasets investigated.

If we look at the results obtained with the f-measure variations, we observe that more statistically significant differences were detected when the global methods outperformed the local methods, than when the local methods outperformed the global methods, and, in the majority of the cases, global methods performed better. We can

summarize the discussion in Sections 4.2 to 4.6 and the results in Table 13 answering the five questions asked in Section 1.

- Q_1**:** Does a specific evaluation measure favour a specific classification approach (global or local) when used to compare global and local based methods?
  According to our experiments, H-Loss seems to favour local methods, so it should not be used as an evaluation measure to compare the predictive performance of local and global methods, because the measure's bias would be unfairly favouring local methods. It is also worth mentioning that the Hierarchical Loss Function does not take into account very skewed class distributions, which are common in HMC problems. Additionally, according to the comparisons performed, H-loss was neither statistically consistent nor more discriminant than the other measures.
  As an example of how H-Loss can favour local methods, consider Figure 14, where bold solid circles represent classes assigned to an instance. Figure 14(a) represents the true classes of an instance, Figure 14(b) represents the classes assigned to the instance by a global classifier, while Figure 14(c) represents the classes assigned to the instance by a local classifier. As can be observed in the figure, even though H-Loss considers mistakes made at higher levels in the class hierarchy more important than mistakes made at lower levels, the calculation of the H-Loss function results in the value 2 for the local method and 4 for the global method. In this case, we can see how the measure favoured the local method, even though its prediction was worse than the prediction made by the global method. We are aware that the situation shown in the figure can be easily reverted if we consider that the global method made the prediction shown in Figure 14(c) and the local method made the prediction shown in Figure 14(b). However, analysing the errors committed by the classifiers in the datasets used in this work, we observed that the situation presented in Figure 14 occurred in the majority of the times, which is confirmed considering that the global methods obtained the best overall f-measure values and the worst H-Loss values.

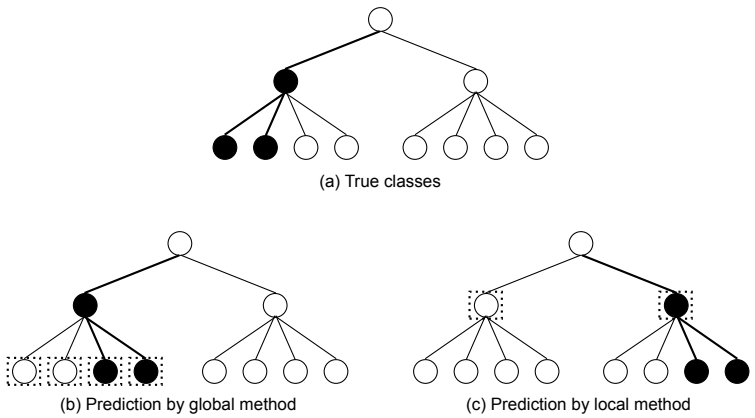


FIGURE 14. Example of how H-Loss can favour local approach when comparing local and global approaches.

- Q_2**:** Which classification approach (global or local) is better overall, considering the four aforementioned classification scenarios?
  Global methods in general obtained better classification performances when the f-measure evaluation was considered, which is an indicative that global methods are better for the HMC tasks investigated. According to the f-measure results presented in Figures 10, 11, 12 and 13, the global methods used in this study presented the best results in the majority of the variations performed in the dataset studied.

- Q_3**:** Do global/local methods are better in predicting more specific/general classes?
  Global methods, in general, achieved better results of recall than local methods, regardless of the dataset characteristics. It seems that, by the experiments and the characteristics of global methods (predictions directly in the more specific classes), the global approach is more suitable for predicting more specific classes, which means a higher recall value. Although this situation can be reversed using high threshold values, the use of thresholds between 0.4 and 0.6 seems a reasonable choice for practical use, since the outputs of these methods

are real values between 0 and 1. The higher precision values obtained by the local methods are a indicative that the local approach is better in predicting the more general classes of the hierarchy. As already mentioned, high values of hierarchical precision measures are a indicative that the classifiers are better in detecting the most general classes of the hierarchy, while high hierarchical recall values are a indicative that the classifiers are better in detecting the most specific classes of the hierarchy. In HMC classification, more specific classes are more informative and potentially more useful predictions to the user – because it allows users to design more specific biological experiments to try to confirm those computational predictions.

- Q_4**:** How different hierarchical and multi-label characteristics influence different evaluation measures and classification approaches?
  As the number of classes assigned to instances in a dataset grows, the results of global methods generally improve for all measures of precision, while the same does not always happen to local methods. The latter may be due to the error propagation inherent to local methods. As more classes are assigned to an instance, we might have more mistakes in the higher levels of the hierarchy, which are propagated downwards.
  In datasets with unbalanced hierarchies, the accuracy of local methods, measured by H-Micro Precision and Recall, increases as the unbalance becomes more evident, while this is not always true for the global methods. This may happen because with more unbalance, errors are less propagated downwards.
  Considering the multi-label variations (increase in the percentage of multi-label instances and number of classes), the experiments showed that the global methods perform better than local methods. We then recommend the use of global methods in datasets with a great number of multi-label instances and classes.
  Hierarchical Macro Precision and Recall are two of the few measures whose behaviours are consistent across most methods. Their absolute values obviously change, but the curves are always very similar.
  Very unbalanced hierarchies seem to favour the classification using local methods. The increase in the number of child nodes of each internal node, however, seems to harm the performance of local methods and favour the use of global methods.

- Q_5**:** Which evaluation measure is more suitable to use in the classification scenarios investigated?
  Considering all the experiments performed, and the comparisons among different measures, if we had to recommend a measure to evaluate any HMC method, in general, we would say the Hierarchical Precision and Recall proposed in (Kiritchenko *et al.*, 2005) effectively do their jobs. Although the Hierarchical Micro and Macro F-Measures were, in general, more discriminant than Hierarchical F-Measure, the hP and hR have the advantage of not depending on any user defined parameter, and the experiments showed that they are consistent if compared to the other measures. The micro and macro measures based on distance, for example, depend on the acceptable distance Dis_θ between the true and predicted classes, which is a subjective user parameter. Depending on the choice of Dis_θ, the values of the measures can drastically change. The Hierarchical Precision and Recall measures were also recommended by Silla and Freitas (2010), although the authors did not perform any empirical comparisons of HMC evaluation measures.

5. CONCLUSIONS AND FUTURE WORKS

This work reviewed some evaluation measures proposed in the literature to assess hierarchical multi-label classification (HMC) methods, and investigated their use for the evaluation of four different decision tree-based HMC methods, two based on the global approach and two based on the local approach. This evaluation employed 12 different real datasets, generated from the original yeast cell cycle microarray experiment (Spellman *et al.*, 1998). Four main characteristics of the datasets were varied: (i) the percentage of multi-label instances, (ii) the number of classes assigned to an instance, (iii) the unbalance of the hierarchy, and (iv) the maximum number of child nodes per internal node.

The HMC evaluation measures analysed in this work were divided into two main groups: distance-based and hierarchy-based evaluation measures. We discussed the advantages and disadvantages of each type of measure, emphasizing that most measures do not take into consideration that predictions at deeper levels of the hierarchy are more difficult and many times more useful, as they lead to more specific information than predictions at shallower levels. We presented alternative ways of dealing with this problem by weighting prediction according

to the hierarchical level. Additionally, we compared the evaluation measures investigated according to their degrees of consistency, discriminancy and indifferency.

As a result of our analysis, we recommend the Hierarchical Precision and Hierarchical Recall measures proposed by Kiritchenko *et al.* (2005) as standard hierarchical multi-label measures, as they do not depend on any user defined parameter, and when compared with the results of other measures, they are relatively consistent across different HMC methods. The use of Precision-Recall curves (Davis and Goadrich, 2006) is also recommended because many different threshold values can be used, leading to a threshold-independent evaluation measure. We also recommend the use of global classification methods, since they presented the best classification performances in the majority of the datasets variations investigated in this work.

As future works, we plan to generate artificial datasets with more hierarchical and multi-label variations including also Gene Ontology structured hierarchies, and also add more hierarchical multi-label methods to the experiments. It is also interesting to develop new HMC methods to generate classification rules. The use of global methods has already shown to be a good alternative to the HMC task (Clare and King, 2003; Vens *et al.*, 2008; Alves *et al.*, 2010; Otero *et al.*, 2010), because they can generate simpler sets of rules and simpler final classifiers.

Acknowledgements

The authors would like to thank the Brazilian research councils CAPES, CNPq, FAPESP and Fapemig for their financial support, and the reviewers for their very constructive suggestions.

REFERENCES

Alves, R., Delgado, M., and Freitas, A. (2008). Multi-label hierarchical classification of protein functions with artificial immune systems. In *III Brazilian Symposium on Bioinformatics*, volume 5167 of *Lecture Notes in Bioinformatics*, pages 1–12, Berlin, Heidelberg. Springer-Verlag.

Alves, R., Delgado, M., and Freitas, A. (2010). Knowledge discovery with artificial immune systems for hierarchical multi-label classification of protein functions. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, volume 1, pages 2097–2104, Barcelona. IEEE.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, **25**(1), 25–29.

Barutcuoglu, Z., Schapire, R. E., and Troyanskaya, O. G. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, **22**(7), 830–836.

Bi, W. and Kwok, J. (2011). Multi-Label Classification on Tree- and DAG-Structured Hierarchies. In L. Getoor and T. Scheffer, editors, *International Conference on Machine Learning*, ICML'11, pages 17–24, New York, NY, USA. ACM.

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.

Blockeel, H., Bruynooghe, M., Dzeroski, S., Ramon, J., and Struyf, J. (2002). Hierarchical multi-classification. In *Proceedings of the first Workshop on Multi-Relational Data Mining*, pages 21–35.

Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, **37**(9), 1757–1771.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Brucker, F., Benites, F., and Sapozhnikova, E. (2011). An empirical comparison of flat and hierarchical performance measures for multi-label classification with hierarchy extraction. In *Proceedings of the 15th international conference on Knowledge-based and intelligent information and engineering systems - Volume Part I*, KES'11, pages 579–589, Berlin, Heidelberg. Springer-Verlag.

Carvalho, A. and Freitas, A. (2009). *A tutorial on multi-label classification techniques*, volume 5 of *Studies in Computational Intelligence 205*, pages 177–195. Springer.

Ceci, M. and Malerba, D. (2007). Classifying web documents in a hierarchy of categories: a comprehensive study. *Journal of Intelligent Information Systems*, **28**, 37–78.

Cerri, R. and Carvalho, A. C. P. L. F. (2010). New top-down methods using svms for hierarchical multilabel

classification problems. In *Proceedings of the International Joint Conference on Neural Networks*, pages 3064–3071, Barcelona. IEEE.

Cerri, R. and Carvalho, A. C. P. L. F. (2011). Hierarchical Multilabel Protein Function Prediction Using Local Neural Networks. In *Proceedings of the Brazilian Symposium on Bioinformatics*, volume 6832, pages 10–17, Brasília-DF, Brazil.

Cerri, R., Carvalho, A. C. P. L. F., and Freitas, A. A. (2011). Adapting non-hierarchical multilabel classification methods for hierarchical multilabel classification. *Intelligent Data Analysis*, **15**(6), 861–887.

Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. (2006). Incremental algorithms for hierarchical classification. *Machine Learning*, **7**, 31–54.

Clare, A. (2003). *Machine Learning and Data Mining for Yeast Functional Genomics*. Ph.D. thesis, University of Wales.

Clare, A. and King, R. D. (2003). Predicting gene function in saccharomyces cerevisiae. *Bioinformatics*, **19**, 42–49.

Cohen, W. W. (1995). Fast effective rule induction. In *In Proceedings of the 12th International Conference on Machine Learning*, pages 115–123.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the International Conference on Machine Learning*, pages 233–240.

Dekel, O., Keshet, J., and Singer, Y. (2004). Large margin hierarchical classification. pages 209–216.

Eisner, R., Poulin, B., Szafron, D., Lu, P., and Greiner, R. (2005). Improving protein function prediction using the hierarchical structure of the gene ontology. In *Proceedings of the Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–10. IEEE.

Freitas, A. A. and Carvalho, A. C. P. F. L. (2007). *A Tutorial on hierarchical classification with applications in bioinformatics.*, volume 1, chapter VII, pages 175–208. Idea Group. Research and Trends in Data Mining Technologies and Applications.

Holden, N. and Freitas, A. (2006). Hierarchical classification of g-protein-coupled receptors with a pso/aco algorithm. In *Proceedings of the Swarm Intelligence Symposium*, pages 77–84. IEEE.

Hollander, M. and Wolfe, D. A. (1999). *Nonparametric Statistical Methods*. Wiley-Interscience, 2 edition.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, **6**, 65–70.

Hornik, K., Buchta, C., and Zeileis, A. (2009). Open-source machine learning: R meets Weka. *Computational Statistics*, **24**(2), 225–232.

Huang, J. and Ling, C. X. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Trans. on Knowl. and Data Eng.*, **17**(3), 299–310.

Ipeirotis, P. G., Gravano, L., and Sahami, M. (2001). Probe, count, and classify: categorizing hidden web databases. *SIGMOD Conference*, **30**, 67–78.

Kiritchenko, S., Matwin, S., and Famili, A. F. (2004). Hierarchical text categorization as a tool of associating genes with gene ontology codes. In *Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics*, pages 30–34, Pisa, Italy.

Kiritchenko, S., Matwin, S., and Famili, A. (2005). Functional annotation of genes using hierarchical text categorization. In *Proceedings of the ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*.

Kiritchenko, S., Matwin, S., Nock, R., and Famili, A. (2006). Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence*, volume 4013 of *Lecture Notes in Computer Science*, pages 395–406. Springer Berlin Heidelberg.

Lord, P., Stevens, R., Brass, A., and Goble, C. (2003). Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation. *Bioinformatics*, **19**(10), 1275–1283.

Mayne, A. and Perry, R. (2009). Hierarchically classifying documents with multiple labels. In *Proceedings of the Symposium on Computational Intelligence and Data Mining*, pages 133–139. IEEE.

Obozinski, G., Lanckriet, G., Grant, C., and Jordan, M.I. Noble, W. (2008). Consistent probabilistic outputs for protein function prediction. *Genome Biology*, **9**(SUPPL. 1).

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*, **2**(3), 165–181.

Pugelj, M. and Džeroski, S. (2011). Predicting structured outputs k-nearest neighbours method. In *Proceedings of the 14th international conference on Discovery science*, pages 262–276, Berlin, Heidelberg. Springer-Verlag.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, **7**, 1601–1626.

Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M., and Mewes, H. W. (2004). The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucl. Acids Res.*, **32**(18), 5539–5545.

Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., and Dzeroski, S. (2010). Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, **11**(1), 2.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**(1), 1–47.

Silla, C. and Freitas, A. (2010). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, **22**, 31–72.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, **45**(4), 427–437.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. *Molecular Biology of the Cell*, **9**(12), 3273–3297.

Struyf, J., Blockeel, H., and Clare, A. (2005). Hierarchical multi-classification with predictive clustering trees in functional genomics. In *Workshop on Computational Methods in Bioinformatics at the 12th Portuguese Conference on Artificial Intelligence*, volume 3808 of *LNAI*, pages 272–283. Springer Berlin / Heidelberg.

Sun, A. and Lim, E.-P. (2001). Hierarchical text classification and evaluation. In *Proceedings of the Fourth International Conference on Data Mining*, pages 521–528. IEEE.

Tsoumakas, G. and Vlahavas, I. (2007). Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Machine Learning*, pages 406–417, Warsaw, Poland.

Tsoumakas, G., Katakis, I., and Vlahavas, I. P. (2010). Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer, 2nd edition.

Valentini, G. (2009). True path rule hierarchical ensembles. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, volume 5519 of *Lecture Notes in Bioinformatics*, pages 232–241, Berlin, Heidelberg. Springer-Verlag.

Valentini, G. (2011). True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **8**(3), 832–847.

Valentini, G. and Cesa-Bianchi, N. (2008). HCGene: a software tool to support the hierarchical classification of genes. *Bioinformatics*, **24**(5), 729–731.

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

Wang, K., Zhou, S., and Liew, S. C. (1999). Building hierarchical classifiers using class proximity. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 363–374, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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