A multi-label classification approach via hierarchical multi-label classification

Mauri Ferrandin (maurisones@gmail.com)
UFSC: Universidade Federal de Santa Catarina https://orcid.org/0000-0002-4248-1207
Ricardo Cerri
Federal University of Sao Carlos: Universidade Federal de Sao Carlos

Research Article

Keywords: Multi-label learning, Multi-label classification, Problem transformation methods, Hierarchical multi-label classification

Posted Date: August 24th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1793069/v1

A multi-label classification approach via hierarchical multi-label classification

Mauri Ferrandin[1*] and Ricardo Cerri[2†]

[1*]Department of Control, Automation and Computer Science, Federal University of Santa Catarina, João Pessoa, 2750, Blumenau, 89036-256, SC, Brazil.
[2]Department of Computer Science, Federal University of São Carlos, São Carlos, Rodovia Washington Luís, São Carlos, 13565-905, SP, Brazil.

*Corresponding author(s). E-mail(s): mauri.ferrandin@ufsc.br;
Contributing authors: cerri@ufscar.br;
†These authors contributed equally to this work.

Abstract

Multi-label classification (MLC) is a very explored field in recent years. The most common approaches that deal with MLC problems are classified into two groups: (i) problem transformation which aims to adapt the multi-label data, making the use of traditional binary or multiclass classification algorithms feasible, and (ii) algorithm adaptation which focuses on modifying algorithms used into binary or multiclass classification, enabling them to make multi-label predictions. Several approaches have been proposed aiming to explore the relationships among the labels, with some of them through the transformation of a flat multi-label label space into a hierarchical multi-label label space, creating a tree-structured label taxonomy and inducing a hierarchical multi-label classifier to solve the classification problem. This paper presents a novel method in which a label hierarchy structured as a directed acyclic graph (DAG) is created from the multi-label label space, taking into account the label co-occurrences using the notion of closed frequent labelset. With this, it is possible to solve an MLC task as if it was a hierarchical multi-label classification (HMC) task. Global and local HMC approaches were tested with the obtained label hierarchies and compared with the approaches using tree-structured label hierarchies showing very competitive results. The main advantage of

the proposed approach is better exploration and representation of the relationships between labels through the use of DAG-structured taxonomies, improving the results. Experimental results over 32 multi-label datasets from different domains showed that the proposed approach is better than related approaches in most of the multi-label evaluation measures. Moreover, we found that both tree and in specially DAG-structured label hierarchies combined with a local hierarchical classifier are more suitable to deal with imbalanced multi-label datasets.

Keywords: Multi-label learning, Multi-label classification, Problem transformation methods, Hierarchical multi-label classification

1 Introduction

Multi-label classification (MLC) has been the focus of many research investigations in recent years. In an MLC task, different from binary or multiclass classifications tasks, one instance can be associated with more than one label simultaneously. Several real-world applications can be modeled as an MLC problem (Zhang and Zhou, 2014), such as in bioinformatics (Zhou et al, 2020), where proteins can perform several functions; text categorization (Wang et al, 2020), where documents belong to multiple categories at the same time; and music classification (Sanden and Zhang, 2011), where songs belong to multiple genres simultaneously. Currently there are two main groups of methods that deal with MLC tasks: problem transformation (PT) and algorithm adaptation (AA).

In the PT methods, the algorithms tackle a multi-label learning problem by transforming it into other well-established learning scenarios (Zhang and Zhou, 2014). Representative algorithms include binary relevance (BR) (Boutell et al, 2004b) and classifier chains (CC) (Read et al, 2011), which transform an MLC task into a binary classification task, and calibrated label ranking (CR) (Fürnkranz et al, 2008), which transforms an MLC task into a label ranking task. The AA methods tackle MLC problems by adapting popular learning techniques, or building completely new ones, to deal directly with multi-label data (Zhang and Zhou, 2014). Some common AA algorithms are ML-kNN (Zhang and Zhou, 2007), BP-MLL (Zhang and Zhou, 2006), ML-DT (Clare and King, 2001), IBRL (Cheng and Hüllermeier, 2009) and PCTs (Blockeel et al, 1998).

According to Madjarov et al (2012), the scenario of MLC methods is complex. The PT methods achieve the best performance in some datasets, whereas the AA methods perform better on other datasets. Moreover, the best performance might be obtained in between the spectrum of the AA and PT methods by transforming the original MLC problem. Nikolosky et al. (Nikoloski et al, 2017), argued that a crucial step in developing methods to output decomposition for MLC is the creation of subspaces. More specifically, the goal is to find a

dependency structure and consider jointly the labels that are interdependent. Typically, these dependencies are represented as hierarchies of labels.

Hierarchical multi-label classification (HMC) tasks are a distinguished group of MLC tasks in which there are hierarchical relationships between the labels. These hierarchical relationships are formally represented through a tree or a directed acyclic graph (DAG) structure (Silla and Freitas, 2011). Similar to MLC tasks, a HMC task can be addressed by different classification strategies: (i) flat, which ignores relationships between labels and defines a classifier for each label in the leaf nodes of the hierarchy; (ii) local, which creates a classifier for each label in the hierarchy and groups the classifiers into three standard subgroups considering how they use this local information and how they build the classifiers around (a local classifier per node, a local classifier per parent node, and a local classifier per level) (Silla and Freitas, 2011); and (iii) global, which defines a single classifier by taking into account all labels and their hierarchical relationships. Clus (Vens et al, 2008) is one of the most used software framework to deal with HMC tasks, is composed of multiple algorithms based on predictive clustering trees (PCT) (Blockeel et al, 1998) to induce classifiers that take into account the hierarchical relationships between labels.

This paper presents an approach named as F2H (flat to hierarchical) to transform a MLC task into a HMC task. The proposal uses the original flat (non hierarchical) multi-label label space to construct a DAG-structured hierarchy with a set of metalabels representing subsets of the original labels. The relationships between metalabels are defined based on the label co-occurrences in the original flat label space using the notions of closed frequent labelset (CFL). Once the label hierarchy is defined, a HMC classifier (local or global) is induced and used to predict the metalabels probabilities for a given test instance. Lastly, the probabilities of the metalabels are mapped to the original labels and used to obtain the final predictions. Similar approaches are proposed in the literature (Madjarov et al, 2015)(Madjarov et al, 2016)(Nikoloski et al, 2017)(Papanikolaou et al, 2018)(Madjarov et al, 2019), which transform a flat label space into a hierarchical one, and used MLC or HMC methods to deal with MLC tasks. These approaches are presented in the following sections.

The remainder of this paper is organized as follows. Section 2 presents the most related approaches that use data-driven algorithms to define hierarchies from the original flat label space and transform an MLC task into a HMC task. Section 3 outlines the notion of CFL and presents the F2H approach proposed in this paper. Section 4 outlines the design of the experimental study, by describing the experimental methodology and setup, the hyperparameter definitions, the evaluation measures, and the statistical analysis of the obtained results. Section 5 presents and discusses the obtained results from different viewpoints. Finally, Section 6 concludes and presents the main outcomes of the study, and also some directions to future exploration on the research field.

2 Related Works

In this section, related works are presented. To our knowledge, there are four studies in the literature that propose and evaluate the transformation of an MLC into a HMC problem. This section presents an overview of these studies.

The Hierarchy Of Multi-label classifiER (HOMER) (Tsoumakas et al, 2008) is a strategy to transform a flat MLC task into a set of minor MLC tasks by using a clustering algorithm to divide the labels into groups and organizing the classifiers in a hierarchical structure. Using a clustering algorithm, the set of original labels is divided into k groups and each group is represented by a metalabel (μ_i). The division is recursively applied until the cluster contains only a single label. Once the hierarchical organization of the labels (and metalabels) is obtained, a multi-label classifier is induced to each metalabel. Testing instances are processed by the model in a top-down direction, selecting for the next level of the hierarchy only the classifiers representing metalabels that contains labels also present in their parent classifiers which had a positive prediction in the upper level. This step is repeated down-tree until find leaf nodes (which contains individual labels).

The main issues in the process of creating the hierarchical organization of the labels in HOMER are how to distribute the set of labels to the k child nodes and, how to obtain the k value which produces the best classification results. To address the clustering problem, authors developed a new balanced clustering algorithm, called balanced k-means, which extends the well-known k-means algorithm with an explicit constraint on the size of each cluster. The selection of the best k for a given classification task needs to be user defined or optimized using training and validation datasets. The performance obtained by HOMER using balanced k-means (with $k = 2, ..., 8$) was compared against that of the original k-means (with no cluster size restriction) and also against that of a random label distribution in the clusters.

A more extensive study about HOMER was presented by Papanikolaou et al (2018), who explored some key issues of the method and proposed some improvements. They conducted experiments to verify the following: (i) how frequent and rare labels are influenced by the total number of nodes in a HOMER model, (ii) the role of hyperparameters k (number of cluster child nodes) and $nmax$ (maximum number of labels in every leaf node) with respect to performance, (iii) the use of three different clustering algorithms, and (iv) the performance of HOMER on two large-scale corpora and compared against the respective chosen baselines. The results pointed in favor to selecting a small number of clusters (two or three) as a safe default choice for configuring a HOMER model, but they emphasize that, due to the great influence of hyperparameter k on the results, it is not possible to make this general recommendation.

Madjarov et al (2015) compared different clustering algorithms for constructing label hierarchies in MLC. They used the label space information to construct the label hierarchies by using four different clustering algorithms

(balanced k-means, agglomerative clustering with single and complete linkage, and PCTs). The resulting hierarchies were structured as trees, having the original labels as leaf nodes. The remaining nodes are a set of metalabels that represent the relationships between the labels according to the clustering approach used. Once the hierarchies are obtained, a global HMC algorithm is used to induce a classifier and to make predictions to new instances.

In their experimental evaluation, Madjarov et al (2015) used 11 multi-label datasets. The results revealed that the data-derived label hierarchies used in conjunction with global HMC methods greatly improved the performances of the MLC methods. Additionally, they compared the use of binary hierarchies (in which a node of the label hierarchy could have a maximum of two child nodes), with multi-branch hierarchies (in which the number of child nodes could be more than two), and the results pointed that the multi-branch approach is much more suitable for the global HMC approaches as compared to the binary hierarchies.

Extended research was done by Madjarov et al (2016) in which two main modifications were included. First, in the creation of the HMC model, they evaluated different approaches using four different types of single predictive models that correspond to binary classification, hierarchical single-label classification, MLC and HMC. The first two approaches construct local predictive models, whereas the last two approaches construct global models. Second, the influence of the use of data-derived label hierarchies on ensemble approaches for HMC was evaluated and analyzed in particular. They used local and global variations of random forests (RF) for MLC and HMC. To define the label hierarchy the balanced k-means was used. The results revealed that the use of the data-derived label hierarchy can significantly improve the performance of single predictive models in MLC as compared to the use of a flat labelset, whereas this is not preserved for the ensemble models.

Nikoloski et al (2017) proposed a new method to define the label hierarchy from a flat multi-label problem taking into account the relevance of the features. The algorithm was also based on the approach presented in Madjarov et al (2015), but changing the label hierarchy definition process. Instead of using the original label space consisting of label co-occurrences, they calculated the feature importance/ranking scores of the features for each individual label by using the GENIE3 method (Huynh-Thu et al, 2010). A matrix containing the importance of each feature for each label is then defined and used as input to clustering methods (the same ones used in Madjarov et al (2015)) to obtain a label hierarchy.

The experimental evaluation used 8 multi-label datasets with 13 different measures and, according to Nikoloski et al (2017), there is no clear winner across all evaluation measures and datasets, and the results pointed that, generally, structuring the output space consisting of feature rankings for each label yields better predictive performance compared to structuring the output space consisting of label co-occurrences considering most of the evaluation measures in almost all datasets. Also, they claim that predictive performance

is improved when comparing the use of RF of PCTs against single PCTs for a large majority of the cases.

Madjarov et al (2019) proposed the construction of label hierarchies in an MLC task in the context of web genre prediction for text classification. In the research, authors investigated the use of different ways to define a hierarchy of labels to be used by a local and global version of the PCT based Clus (Vens et al, 2008) hierarchical multi-label classifier. The methods used to define the label hierarchy included the balanced k-means, PCTs, clustering with single and complete linkage, random and manual (defined by a domain context expert).

The results showed that exploiting a hierarchy of web genres achieved best predictive results across the two used datasets for all methods used to define the label hierarchies. Moreover, they found that data-driven hierarchies construction is at least as good as expert-constructed hierarchies with the added value that the hierarchy construction is automatic and fast. Authors also investigated the use of bagging and random forest ensembles models and concluded they have a superior performance than single tree models.

In this section, related works using the strategy to create a label hierarchy for transforming an MLC task in a HMC task were presented. The approaches presented by Tsoumakas et al (2008), Papanikolaou et al (2018) and Madjarov et al (2016), which used only the label space in the label hierarchy definition and were evaluated over datasets frequently used in the literature, will be compared to the method proposed in this paper.

3 Proposed Method

In this section the proposed method named as F2H (Flat to Hierarchical) is presented. The notion of closed frequent labelset (CFL) is presented in Section 3.1, and the method is detailed in Section 3.2.

3.1 Closed Frequent Labelset (CFL) and Label Hierarchy Definition

In this section, the notion of CFL is introduced. The idea of CFL is an adaptation from the closed itemset framework introduced by Pasquier et al (1998), who proposed an algorithm called Close, to address the problem of association rule mining from dense datasets (Pasquier et al, 1999). The proposed framework was based on the closure operator of the Galois connection used in formal concept analysis (FCA) (Ganter, 1984). Formally, the set of CFLs from a multi-label dataset is obtained as follows:

Given,

- a set of all labelsets $L = \{l_1, l_2, ..., l_n\}$ defined by $L = 2^{\mathcal{L}}$ where $\mathcal{L}$ is the set of all labels;

- a label space $Y = \{\mathcal{Y}_1, \mathcal{Y}_2, ..., \mathcal{Y}_m\}$ from a multi-label dataset E with m instances;
- a labelset incidence function denoted by $I(l_i) = \{\mathcal{Y} \in Y \mid l_i \subseteq \mathcal{Y}\}$ which returns the set of instances in which all labels from the labelset l_i are contained;
- a labelset support function representing the proportion of instances containing the labelset, denoted by $s(l_i) = \mid I(l_i) \mid /m$.

Find: a set $M = \{\mu_1, \mu_2, ..., \mu_n\}$ containing all CFLs such that

- $M \subseteq L$;
- for each $\mu_i \in M$, there is no $l_j \subset \mu_i$ in which $I(\mu_i) = I(l_j)$; and
- foreach $\mu_i \in M$, $s(\mu_i) \geqslant \tau_{CFL}$, where τ_{CFL} is a user defined minimum labelset support threshold used to prune the hierarchy to avoid the exponential behavior.

The set of all CFLs organized by inclusion order could be represented as a DAG that will be used as the label hierarchy allowing to treat a flat MLC task as a HMC task. Figure 1 shows the selection of CFLs to a hypothetical multi-label dataset with $\mathcal{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. For the sake of comprehensibility, the representations of a set are simplified in the picture by avoiding the use of $\{\}$. The figure shows a DAG with all possible labelsets (all nodes), the subset of labelsets that are CFLs (nodes with gray background), and the subset of selected CFLs given a minimal support threshold (nodes with highlighted rectangle outline).

Taking a look, for example, at the node l_5 which contains the label λ_4, it is not a CFL because there exists another node (l_{11}) in which λ_4 is contained and has the same instances (given by the I function). A similar case occurs with the nodes l_8 and l_{14}. In spite of being CFLs, some nodes will not be selected to the final label hierarchy due to the minimum support threshold τ_{CFL} defined as 0.4 in this example. This situation occurs with the nodes l_{12} and l_{15}, and the infimum l_{16}.

After the selection of CFLs from all candidate labelsets, the hierarchical label representation is obtained by keeping only the selected nodes and edges from the DAG as shown in Figure 2. In this example, the set of selected CFLs is denoted as $M = \{\mu_1, \mu_2, ..., \mu_n\}$ with $n = 9$. Hereinafter, we will refer to M as metalabels.

3.2 F2H (Flat to Hierarchical) Approach

The F2H algorithm is presented in Algorithm 1. Essentially, it is very similar to that by Madjarov et al (2015), presented in Section 2, but with one additional hyperparameter τ_{CFL} that represents the minimum labelset support threshold and also two different steps: the label hierarchy definition (line 3) and the conversion of HMC to MLC predictions (line 16). These steps are detailed in the next subsections.

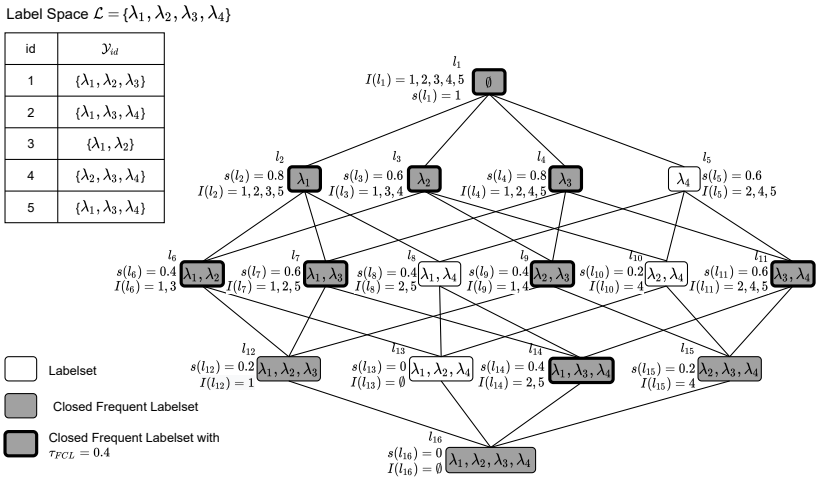


Fig. 1: DAG of all labelsets from a sample dataset with CFL selection and τ_{FCL} demonstration.

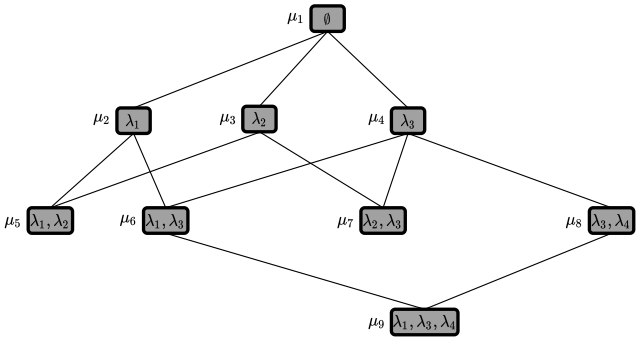


Fig. 2: DAG with a hierarchical representation of the selected CFLs (metal-abels).

3.2.1 Defining the label hierarchy

Algorithm 2 presents the steps to define the hierarchical representation of a label space as a DAG, enabling the conversion from a flat to a hierarchical label space representation. To find the set of all CFLs given a minimum support threshold (line 3), a well-known algorithm called PCBO (Krajca and Vychodil, 2009) was used. Also, to find all edges by considering the inclusion order of the CFLs (line 6), the algorithm called FindEdges (Nourine and Raynaud, 1999) was used.

Both algorithms mentioned here are from the context FCA techniques. As stated in the problem definition, finding all CFLs from a multi-label dataset can lead to an exponential behavior of the algorithm ($M = 2^{\mathcal{L}}$) considering

Algorithm 1 - F2H

procedure F2H(E^{Train}, E^{Test}, τ_{CFL}) **returns** performance
    $W^{train} \leftarrow$ ExtractLabelSet(E^{Train});
    $W_H^{train} \leftarrow$ DefineDAGHierarchy(W^{train}, τ_{CFL});

    // transform multi-label dataset to hierarchical multi-label one
    $E_H^{train} \leftarrow$ MLCToHMCTrainDataset(E^{train}, W_H^{train});

    //solve transformed hierarchical multi-label problem
    //by using approach for HMC
    HMCModel $\leftarrow$ HMCMethod(E_H^{train});

    //generate HMC predictions
    $P_H \leftarrow$ HMCModel(E^{test});

    // Convert HMC predictions P_H to MLC predictions
    $P \leftarrow$ ConvertHMCtoMLCPredictions(P_H, W_H^{train});
    return EvaluatePredictions(P, E^{test});
end procedure

the processing time in the worst case. Nevertheless, any algorithm used to find frequent itemsets could be adopted, the PCBO algorithm was chosen due to its efficiency in pruning the search space to avoid the generation of all candidate labelsets and also due to its minimum support functionality definition. The steps for the label hierarchy definition in a hypothetical multi-label dataset are presented and detailed in Section 3.1. Algorithm 2 returns a DAG with a set of metalabels that are a hierarchical representation of the original flat label space.

Algorithm 2 - F2H - DefineDAGHierarchy

procedure DefineDAGHierarchy(W^{train}, τ_{CFL}) **returns** W_H^{train}
    // find M, the set of CFLs in W^{train} with $s(\mu_i) \geqslant \tau_{CFL}$
    $M \leftarrow$ findClosedLabelSets(W^{train}, τ_{CFL})

    // find Ed, the set all edges in M
    $Ed \leftarrow$ findEdges(M)

    // create a DAG with the nodes (M) and Edges (Ed)
    $W_H^{train} \leftarrow$ createDAG(M,Ed);
    return W_H^{train};
end procedure

3.2.2 Transforming the datasets from MLC to HMC

After obtaining the DAG structured label hierarchy W_H^{train}, the label space of the training dataset needs to be modified to allow the use of a HMC classifier (line 6 of Algorithm 1). The label associations of the original flat label space are discarded and each instance is associated with a set of metalabels from the label hierarchy W_H^{train}. For each training instance $(X_i, \mathcal{Y}_i)$, in the training dataset, the conversion can be formally represented as follows:

Given a set of metalabels in the label hierarchy denoted by $M = \{\mu_1, \mu_2, ..., \mu_n\}$ and a training instance $(X_i, \mathcal{Y}_i)$ where $X_i \in \mathcal{X}$ (with $\mathcal{X}$ being the attribute space) and $\mathcal{Y}_i \subseteq \mathcal{L}$ (with $\mathcal{L}$ being the label space), a new training instance $(X_i, \mathcal{Y}'_i)$ is created where $\mathcal{Y}'_i \subseteq M$ and, foreach $\mu_j \in \mathcal{Y}'_i$, $\mu_j \subseteq \mathcal{Y}_i$ and there is no $\mu_k \in \mathcal{Y}'_i$ such that $\mu_j \supset \mu_k$.

As can be seen, the training dataset conversion is done by keeping the feature space unchanged and replacing the label space by a set of metalabels from the label hierarchy.

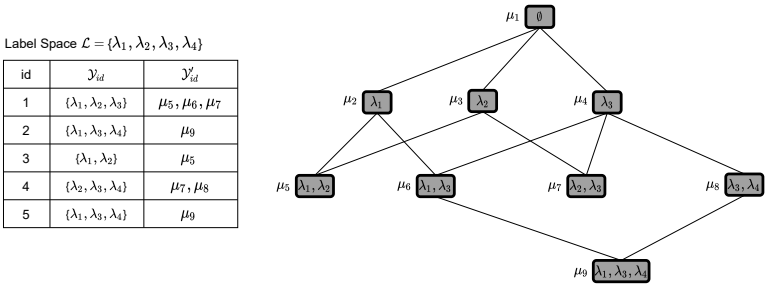


Fig. 3: Dataset transformation.

In Figure 3 the label conversions of some instances are shown. The label hierarchy is the same as that represented in Figure 1 and the label space of a set of instances is presented in the table within Figure 3. The set of labels of the first instance $\mathcal{Y}_1 = \{\lambda_1, \lambda_2, \lambda_3\}$ results in the set of hierarchical metalabels $\mathcal{Y}'_1 = \{\mu_5, \mu_6\, \mu_7\}$ (only the CFLs μ_5, μ_6 and μ_7 hold the last stated condition). Here, it is important to mention that in a HMC problem an instance associated with a given metalabel μ_i from the label hierarchy also belongs to all the parent metalabels of μ_i due to the hierarchical constraint (i.e., if the instance belongs to μ_5, it implicitly belongs also to μ_1, μ_2 and μ_3).

For the second instance $\mathcal{Y}_2 = \{\lambda_1, \lambda_3, \lambda_4\}$, the set of hierarchical labels is $\mathcal{Y}'_2 = \{\mu_9\}$. As can be seen, in this case, a metalabel containing all labels from $\mathcal{Y}_2$ was found. Due to the CFL properties, this situation will be true for all training instances if we define the minimum CFL support threshold as $\tau_{FCL} = 0$.

3.2.3 Training the hierarchical classifier and making hierarchical predictions

Once the hierarchical training dataset E_H^{train} is defined, a HMC classifier is induced using the dataset. Even though any HMC classifier could be used in this step, the PCT based Clus (Vens et al, 2008) was choosen due to its interpretability, capability of induce models using different approaches: flat MLC; local HMC (ClusHSC) and global HMC (ClusHMC) enabling a more fair comparison between them; and the ability of predict a probability score for all the nodes in the hierarchy what allows different ways to explore the results.

After the training step, predictions for the instances from a test dataset are obtained. For each test instance, the classifier outputs a probability score of each metalabel in the label hierarchy. Formally, for a given test instance X_i, the hierarchical classifier outputs a set $p_{Hi} = \{(\mu_1, p_{\mu_1}), (\mu_2, p_{\mu_2}), ..., (\mu_n, p_{\mu_n})\}$, where $\mu_j \in M$, $1 \leqslant j \leqslant n$ and $p_{\mu_j} \in [0, 1]$. The p_{μ_j} value is the probability of the instance X_i to belong to the metalabel μ_j according to the induced classifier. The set of predictions for all m test instances is the set $P_H = \{p_{H1}, p_{H2}, ..., p_{Hm}\}$, represented in line 13 of Algorithm 1.

Here, it is important to notice that a metalabel represents a set of original flat labels, so it is necessary to convert the obtained metalabel-based predictions assigned to test instances to the original labels as presented bellow.

3.2.4 Converting the obtained predictions from HMC to MLC

In this step, the hierarchical multi-label probability scores given to each metalabel are converted into original flat multi-label probability scores. Formally, for a given training instance X_i, the conversion is achieved by the following: Given a set of flat labels $\mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ and a set of predicted probabilities to X_i denoted as $p_{Hi} = \{(\mu_1, p_{\mu_1}), (\mu_2, p_{\mu_2}), ..., (\mu_n, p_{\mu_n})\}$ with every metalabel μ_j being a set of original flat labels, find a set $p_i = \{(\lambda_1, p_{\lambda_1}), (\lambda_2, p_{\lambda_2}), ..., (\lambda_q, p_{\lambda_q})\}$ where $\lambda_k \in \mathcal{L}$ and $1 \leqslant k \leqslant q$ and $p_{\lambda_k} = \arg\max_{1 \leqslant j \leqslant n, \lambda_k \subseteq \mu_j} p_{\mu_j}$.

The result of this step is a set of pairs $(\lambda_1, p_{\lambda_1})$ containing a flat label and its probability. The probability p_{λ_k} is the final probability of the instance X_i to belong to the label λ_k with respect to the induced model. The set of multi-label predictions for all m test instances is the set $P = \{p_1, p_2, ..., p_m\}$ represented in line 16 of Algorithm 1.

Figure 4 presents the prediction conversion from hierarchical metalabels to original flat labels for a given test instance X_6. Each metalabel has the prediction given by the HMC classifier. The probability scores of a given original flat label λ_i is equal to the probability score of the metalabel containing λ_i and with the highest probability score. To exemplify this, considering the hypothetical HMC predictions to a test instance X_6, the metalabel μ_2 was selected to define the score of λ_1 because it has the highest probability (0.8) among all other metalabels containing λ_1.

As can be seen in the example of Figure 4, a PCT naturally produces higher scores to labels closer to the root node of the hierarchy. In the PCT outputs, the score of each node in the hierarchy is proportional to the number of training instances it contains. Thus, more distant nodes of the root node will score less than closer nodes.

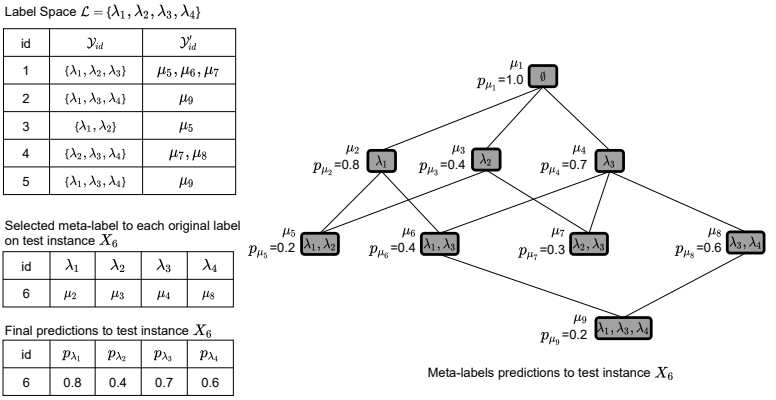


Fig. 4: Transformation of the probabilities from HMC metalabels to MLC original labels.

3.2.5 Multi-label prediction thresholding

Once the probability of each label is obtained for all test instances, a thresholding method is necessary to convert the probability value into a bipartition discrete representation (relevant or irrelevant). According to Ioannou et al (2010), the need for this conversion depends on the context the classification results are used (in some of them, just the probabilities are enough; in others, on the contrary, bipartitions are more suitable), and several of the state-of-the-art MLC algorithms of the literature (Tsoumakas et al, 2011a), (Read et al, 2008), (Read et al, 2011), actually output a score vector primarily and employ one (sometimes simple) thresholding method to be able to output bipartitions.

The thresholding method chosen for the proposed algorithms uses a OneThreshold strategy that instead of using a fixed user-defined value as threshold, the value of t was defined using a calibration method that decidedly produces better results rather than simply using an arbitrary threshold like 0.5 (Fan and Lin, 2007). The method used to calibrate t for a given dataset E is presented in Equation 1, where $\hat{y_{ij}}$ is the probability score for instance, i and label j, m is the number of test instances, q is the number of labels and $LCard(E^{train})$ is a function that gives the label cardinality of the training dataset. This calibration method was proposed by Read et al (2011) in the CC method. The main idea is to choose a t value such that the label cardinality of

the predictions are as close as possible to the label cardinality of the training dataset.

$$t = \underset{0 \leq t \leq 1}{\arg\min} \left(LCard(E^{train}) - \left(\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{q} 1_{\hat{y}_{ij} \geq t} \right) \right) \quad (1)$$

Finally, the thresholding strategy used here can be used with any classification measure to assess the classifier performance. Here, it is important to notice that the proposed approach for thresholding the F2H results uses only the information of the label cardinality from the training set and is not optimizing a specific MLC measure.

3.3 F2H and Related Works - Main Differences

There are three main differences between F2H approach and related approaches. The fist one is how the labels are grouped to create a label hierarchy. The related approaches mostly use hierarchical clustering algorithms to verify the label correlations and choose which labels will be present in a node of the hierarchy. The F2H approach uses the notion of CFL which is based on label co-occurrences and adapted from a method frequently used to rule mining and to find implications in data. The second difference is how the label hierarchy is represented. In the related approaches, the label hierarchy is represented as a tree in which there is a set of metalabels and the leaves are the original labels. After the induction of a HMC classifier over the tree, it must make mandatory leaf-node predictions to test instances. On the other hand, the F2H approach uses a DAG to represent the label hierarchy, and all nodes in the DAG are metalabels that contain a set of original flat labels. Even though, in some cases, a metalabel contains just one of the original labels, the classifier is allowed to predict any metalabel at any level of the label hierarchy for a given test instance. The last one is their experimental evaluation. Related works presented restricted experimental evaluations with a smaller number of datasets and using the hold-one-out method in some cases. In this work, the experiments were performed using 32 datasets well known in the MLC literature and using 10-fold cross-validation.

This section presented the proposed MLC method F2H, detailing the steps to obtain the final multi-label predictions by transforming the MLC task into a HMC one. The notion of CFL and the main differences between F2H and related approaches are also presented.

4 Experimental Evaluation

In this section the experimental evaluation is presented describing the data, the performance measures, the baseline algorithms, the results and other additional information.

4.1 Materials and Methods

4.1.1 Datasets

To evaluate the proposed approach, 32 popular multi-label datasets were selected from different domains. The datasets were obtained from the Cometa repository (Charte et al, 2018), an exhaustive collection of MLC datasets, integrated with a set of software tools. The datasets are presented in Table 1.

Table 1: Datasets used in experiments (m: instances; d: attributes; $|\mathcal{L}|$: labels; $|\text{Uniq}(L)|$: unique label combinations; MaxF: occurrences of the most common labelset; Card: cardinality; Dens: density; and MeanIR: mean imbalance ratio (Charte et al, 2013).

Name	Domain	m	d	$\lvert\mathcal{L}\rvert$	$\lfloor\rfloor$	$\lvert\text{Uniq}(L)\rvert$	MaxF	Card	Dens	MeanIR	Ref.
bibtex	Text	7395	1995	159	2856	2199	471	24.02	0.02	12.49	Katakis et al (2008)
birds	Audio	645	279	19	133	73	294	1.01	0.05	5.40	Briggs et al (2012)
cal500	Audio	502	242	174	502	502	1	260.44	0.15	20.57	Turnbull et al (2008)
corel5k	Image	5000	873	374	3175	2523	55	3.52	0.01	189.56	Duygulu et al (2002)
emotions	Audio	593	78	6	27	4	81	18.69	0.31	1.47	Wieczorkowska et al (2006)
enron	Text	1702	1054	53	753	573	163	33.78	0.06	73.95	Klimt and Yang (2004)
EukaryoteGO	Biology	7766	12711	22	112	37	1580	11.46	0.05	45.01	Xu et al (2016)
EukaryotePseAAC	Biology	7766	462	22	112	37	1580	11.46	0.05	45.01	Xu et al (2016)
flags	Other	194	26	7	54	24	27	33.92	0.48	2.25	Goncalves et al (2013)
foodtruck	Other	407	33	12	116	74	115	22.90	0.19	7.09	Rivolli et al (2017)
genbase	Biology	662	1213	27	32	10	170	12.52	0.05	37.31	Diplaris et al (2005)
GnegativeGO	Biology	1392	1725	8	19	5	533	1.05	0.13	18.44	Xu et al (2016)
GnegativePseAAC	Biology	1392	448	8	19	5	533	1.05	0.13	18.44	Xu et al (2016)
GpositiveGO	Biology	519	916	4	7	2	206	10.08	0.25	3.86	Xu et al (2016)
GpositivePseAAC	Biology	519	444	4	7	2	206	10.08	0.25	3.86	Xu et al (2016)
HumanGO	Biology	3106	9858	14	85	28	718	11.85	0.08	15.28	Xu et al (2016)
HumanPseAAC	Biology	3106	454	14	85	28	718	11.85	0.08	15.28	Xu et al (2016)
langlog	Text	1460	1079	75	304	189	207	11.80	0.02	39.26	Read (2010)
medical	Text	978	1494	45	94	33	155	12.45	0.03	89.50	Crammer et al (2007)
ng20	Text	19300	1026	20	55	17	997	10.29	0.05	1.00	Lang (1995)
ohsumed	Text	13929	1025	23	1147	578	1175	16.63	0.07	7.86	Joachims (1998)
PlantGO	Biology	978	3103	12	32	8	277	10.79	0.09	6.69	Xu et al (2016)
PlantPseAAC	Biology	978	452	12	32	8	277	10.79	0.09	6.69	Xu et al (2016)
reutersk500	Text	6000	603	103	811	513	381	14.62	0.01	51.98	Read (2010)
scene	Image	2407	300	6	15	3	405	1.07	0.18	1.25	Boutell et al (2004a)
slashdot	Text	3782	1101	22	156	56	525	11.81	0.05	17.69	Read et al (2011)
stackex_chess	Text	1675	812	227	1078	890	48	24.11	0.01	85.78	Charte et al (2015)
tmc2007_500	Text	28596	522	22	1172	408	2484	22.20	0.10	15.15	Katakis et al (2008)
VirusGO	Biology	207	755	6	17	6	56	12.17	0.20	4.04	Xu et al (2016)
VirusPseAAC	Biology	207	446	6	17	6	56	12.17	0.20	4.04	Xu et al (2016)
yeast	Biology	2417	117	14	198	77	237	42.37	0.30	7.19	Elisseeff and Weston (2001)
Yelp	Image	10806	676	5	32	0	2120	16.38	0.33	2.87	Sajnani et al (2013)

Each dataset was split into 10 folds for the experiments. To minimize the occurrence of imbalanced data or the absence of positive samples to given labels in the generated folds, the split was done using an iterative stratification strategy proposed specifically for multi-label data (Sechidis et al, 2011). This strategy calculates the number of instances in each fold by examining iteratively each instance to distribute properly the instances in the generated folds.

4.1.2 F2H versions and baseline methods

Table 2 presents the main characteristics of the classifiers used in the experiments. The proposed F2H is represented by two versions: F2H-L and F2H-G.

Both use a label hierarchy defined as a DAG; the first one uses a local HMC approach, whereas the second one uses a global approach.

Table 2: Configuration of the used approaches compared in the experiments.

Approach	Label Hierarchy	Base Classifier	Framework	Hierarchy Definition	Reference
F2H-L	DAG	Local PCT per edge	ClusHSC	CFL	Proposed
F2H-G	DAG	Global PCT	ClusHMC	CFL	Proposed
Flat	None	Global Multi-target PCT	Clus	None	Blockeel et al (1998)
T-L	Tree	Local PCT per edge	ClusHSC	Balanced k-means	Madjarov et al (2016)
T-G	Tree	Global PCT	ClusHMC	Balanced k-means	Madjarov et al (2016)
HO (HOMER)	Tree	Decision Tree (J48)	Mulan	Balanced k-means	Tsoumakas et al (2008)

4.1.3 Multi-label performance measures

There exist several different metrics to evaluate multi-label classifiers taking into account different aspects (Zhang and Zhou, 2014), (Gibaja and Ventura, 2014). To evaluate the F2H and baseline methods, we used six example-based evaluation measures (F1 score, precision, recall, hamming-loss, subset accuracy, and accuracy), six label-based evaluation measures (macro-F1, macro-precision, macro-recall, micro-F1, micro-precision, and micro-recall), four ranking-based measures (one-error, average-precision, coverage, and ranking-loss), and two label-problem measures (MLP and WLP). The equations for all the used measures are presented in Appendix A.

In the experimentation, all classifiers were set up to produce probability scores for each label on each testing instance. However, the step of obtaining the example and label-based measures requires predictions stating that a given label is present or not (binary 1/0 predictions) in a given instance. To obtain the binary predictions, a threshold was applied to the probabilities following the same rules defined for F2H in Section 3.2.5.

The selected measures aim to evaluate classifiers using multiple and contrasting measures. Moreover, to make a fair comparison, the threshold strategy are the same as those used by Madjarov et al (2016).

4.1.4 Algorithms setup and execution environment

In this section, the algorithms hyperparameters set up are detailed, and the information about the execution environment is also presented.

F2H[1] versions have naturally two hyperparameters: (i) τ_{CFL} - is the minimal support value used to prune CFLs in the label hierarchy. In the final run it was set to 0 (zero) in most datasets except cal500, where it was set to 0.3, and langlog, ng20, reutersk500 and corel5k, where it was set to 0.1. The values of τ_{CFL} were empirically chosen through experimentation trying to run all datasets using 0 (zero), but in some datasets, like cal500, when setting $\tau_{CFL} = 0$, the number of CFLs obtained was more than 2 million, making it computationally unfeasible. For those datasets, the τ_{CFL} was

[1]The algorithm source code implemented as a R package will be further available at: https://github.com/maurisones/F2H

incrementally modified until to reach a value that was possible to finish the experiment in a feasible time. (ii) Nonetheless, as the hierarchical classification is performed by using PCTs throughout the ClusHMC framework, there are several hyperparameters that must be set in this step. To keep it simple, most of PCT hyperparameters were kept default as recommended in the Clus distribution[2]; only the hyperparameters WParam and OptimizeErrorMeasure had their values adjusted throughout experimentation to produce better final F2H results. WParam was set to 0.8 and OptimizeErrorMeasure was set to WeightedAverageAUPRC.

The Clus configuration to other approaches (Flat, T-G and T-L) was also used with the proposed default values like described in the baseline methods. The balanced k-means was used with fixed $k = 3$ (Madjarov et al (2015) suggests that this value is the best in most datasets) and $iterations = 100$ in all cases. In the special case of HO, its execution was performed using Mulan (Tsoumakas et al, 2011b), with default configuration and J48 as base classifier.

All experiments used 10-fold cross-validation and the results were aggregated using averages. The running environment was composed of a cluster environment with several nodes, with each one containing 28-core Intel(R) Xeon(R) CPU E5−2660 2.00 GHz with 128 GB of RAM. Most of the executed methods had no support to multi-threading or parallel execution were ran without concerns about this issue and with exclusive resources of the respective running node. Analysis of the executions times is presented in Section 4.3.

4.1.5 Statistical analysis

To verify whether the overall differences in performance across the different approaches on each specific measure are statistically significant, the corrected Friedman test was used, followed by Nemenyi post-hoc test, as recommended by Demšar (2006). The results from the Nemenyi post-hoc test are presented in Section 4.2 with average rank diagrams, also known as critical diagrams. A critical diagram contains an enumerated axis on which the average ranks of the algorithms are drawn. The algorithms are represented on the axis considering the order of best ranking. Better algorithms are represented in left most side of the diagram. The lines for the average ranks of the algorithms that do not differ significantly (at a significance level of α) are connected with a horizontal line.

4.2 Results

In this section the results of the proposed experiments are presented. Tables 3 - 8 present the results obtained by the classifiers. In the tables, best values are highlighted in bold and in addition to the averages we also present the the ranking sum that consists of the sum of the positions of each classifier in the ranking of each dataset for the respective measure.

[2] https://dtai.cs.kuleuven.be/clus/

Table 3: Label-based: Macro-F1, Macro-Precision and Macro-Recall results.

	Macro-F1						Macro-Precision						Macro-Recall					
	F2H-L	F2H-G	T-G	T-L	Flat	HO	F2H-L	F2H-G	T-G	T-L	Flat	HO	F2H-L	F2H-G	T-G	T-L	Flat	HO
bibtex	**0.312**	0.282	0.271	0.301	0.006	0.256	**0.321**	0.299	0.295	0.315	0.006	0.284	**0.332**	0.304	0.299	0.320	0.013	0.252
birds	0.281	0.255	0.151	**0.289**	0.002	0.284	0.284	0.271	0.148	**0.300**	0.001	0.298	**0.337**	0.298	0.193	0.322	0.053	0.321
cal500	**0.230**	0.115	0.125	0.159	0.036	0.170	0.150	0.091	0.098	**0.166**	0.030	0.155	**0.952**	0.237	0.238	0.168	0.047	0.220
corel5k	**0.032**	0.028	0.025	0.031	0.000	0.023	0.034	0.030	0.030	0.034	0.000	**0.035**	**0.036**	0.033	0.030	0.034	0.003	0.020
emotions	0.590	0.583	0.561	0.591	**0.594**	0.579	0.589	**0.625**	0.596	0.601	0.623	0.582	**0.603**	0.587	0.564	0.600	0.591	0.590
enron	**0.180**	0.150	0.130	0.173	0.042	0.177	0.193	0.175	0.148	**0.194**	0.042	0.184	**0.190**	0.153	0.138	0.173	0.046	0.180
EukaryoteGO	**0.663**	0.598	0.556	0.662	0.352	0.637	**0.683**	0.643	0.591	0.682	0.391	0.619	0.702	0.606	0.578	0.701	0.360	**0.703**
EukaryotePseAAC	**0.118**	0.089	0.082	0.103	0.002	0.102	**0.122**	0.117	0.093	0.103	0.008	0.099	**0.126**	0.098	0.106	0.115	0.044	0.124
flags	0.641	0.651	0.642	0.649	0.612	**0.655**	0.667	**0.695**	0.674	0.665	0.626	0.626	0.655	0.647	0.650	0.662	0.653	**0.711**
foodtruck	0.209	0.196	0.172	0.223	0.070	**0.242**	0.208	0.180	0.140	0.227	0.061	**0.248**	0.227	0.232	0.231	0.241	0.083	**0.252**
genbase	**0.653**	0.627	0.592	0.641	0.467	0.651	0.652	0.624	0.590	0.641	0.472	**0.659**	**0.660**	0.639	0.600	0.647	0.466	0.649
GnegativeGO	**0.931**	0.916	0.918	0.925	0.694	0.893	**0.929**	0.919	0.922	0.924	0.711	0.888	**0.940**	0.924	0.924	0.933	0.685	0.905
GnegativePseAAC	**0.373**	0.300	0.286	0.316	0.197	0.310	**0.347**	0.326	0.298	0.306	0.235	0.298	**0.427**	0.309	0.295	0.346	0.211	0.359
GpositiveGO	0.886	0.870	0.888	**0.889**	0.883	0.877	0.874	0.879	0.887	**0.891**	0.885	0.870	**0.921**	0.885	0.911	0.906	0.905	0.907
GpositivePseAAC	**0.465**	0.439	0.417	0.424	0.450	0.411	0.432	0.436	0.418	0.398	**0.450**	0.391	**0.528**	0.457	0.434	0.474	0.456	0.455
HumanGO	0.632	0.588	0.580	**0.646**	0.414	0.624	0.638	0.638	0.626	**0.652**	0.465	0.614	0.677	0.596	0.598	**0.688**	0.417	0.657
HumanPseAAC	**0.153**	0.134	0.120	0.148	0.003	0.134	**0.154**	0.151	0.127	0.148	0.002	0.130	**0.179**	0.146	0.143	0.174	0.071	0.153
langlog	0.061	0.049	0.036	**0.065**	0.000	0.047	0.061	0.050	0.041	**0.065**	0.000	0.046	**0.078**	0.062	0.048	0.076	0.013	0.054
medical	**0.345**	0.300	0.267	0.321	0.026	0.295	**0.342**	0.304	0.273	0.320	0.022	0.300	**0.365**	0.313	0.283	0.342	0.044	0.303
ng20	0.588	**0.603**	0.574	0.568	0.360	0.579	0.588	0.605	0.581	0.568	**0.643**	0.569	0.622	**0.622**	0.602	0.601	0.310	0.620
ohsumed	**0.407**	0.386	0.374	0.403	0.152	0.357	0.427	0.447	**0.470**	0.423	0.297	0.367	**0.418**	0.368	0.345	0.412	0.137	0.363
PlantGO	0.668	0.677	**0.678**	0.668	0.402	0.621	0.681	0.755	**0.759**	0.692	0.422	0.624	**0.722**	0.670	0.671	0.711	0.433	0.671
PlantPseAAC	**0.162**	0.147	0.087	0.161	0.009	0.155	0.168	0.146	0.075	**0.172**	0.005	0.143	0.193	0.177	0.126	0.195	0.083	**0.196**
reutersk500	0.233	0.196	0.188	**0.239**	0.000	0.209	**0.248**	0.215	0.219	0.247	0.000	0.240	0.244	0.205	0.201	**0.258**	0.010	0.204
scene	0.608	0.603	**0.615**	0.599	0.613	0.584	0.598	0.603	0.616	0.585	**0.640**	0.555	**0.650**	0.615	0.625	0.644	0.600	0.635
slashdot	**0.358**	0.291	0.293	0.326	0.048	0.243	**0.375**	0.312	0.334	0.331	0.100	0.315	**0.382**	0.295	0.289	0.350	0.081	0.221
stackex_chess	**0.113**	0.053	0.046	0.092	0.000	0.068	**0.108**	0.050	0.045	0.087	0.000	0.072	**0.137**	0.068	0.058	0.110	0.004	0.073
tmc2007_500	0.568	0.474	0.436	0.538	0.264	**0.603**	0.594	0.520	0.575	0.599	0.449	**0.621**	0.555	0.451	0.397	0.507	0.221	**0.592**
VirusGO	0.817	0.805	0.789	0.831	0.757	**0.860**	0.805	0.833	0.827	0.831	0.826	**0.843**	0.886	0.813	0.788	0.860	0.767	**0.909**
VirusPseAAC	0.235	0.305	0.166	0.248	0.235	**0.357**	0.238	0.283	0.146	0.231	0.243	**0.337**	0.322	0.378	0.229	0.338	0.300	**0.439**
yeast	0.395	0.389	0.352	0.397	0.301	**0.407**	0.399	0.438	0.432	0.400	**0.491**	0.393	0.397	0.393	0.373	0.399	0.280	**0.428**
Yelp	0.696	0.662	0.650	**0.698**	0.614	0.687	0.687	0.666	0.655	0.688	**0.696**	0.668	0.709	0.662	0.651	0.711	0.566	**0.717**
Average	**0.416**	0.390	0.368	0.407	0.258	0.400	**0.416**	0.408	0.390	0.413	0.295	0.400	**0.466**	0.406	0.386	0.429	0.271	0.425
Rank. Sum.	**61**	116	141	74	176	104	**82**	100	128	85	156	121	**53**	121	147	78	183	90

Table 4: Label-based: Micro-F1, Micro-Precision and Micro-Recall results.

	Micro-F1						Micro-Precision						Micro-Recall					
	F2H-L	F2H-G	T-G	T-L	Flat	HOM	F2H-L	F2H-G	T-G	T-L	Flat	HOM	F2H-L	F2H-G	T-G	T-L	Flat	HOM
bibtex	**0.402**	0.391	0.396	0.400	0.088	0.345	**0.391**	0.382	0.388	0.386	0.150	0.349	0.413	0.400	0.406	**0.415**	0.063	0.342
birds	**0.361**	0.311	0.267	0.324	0.022	0.349	**0.298**	0.258	0.222	0.267	0.022	0.294	**0.458**	0.393	0.335	0.411	0.021	0.431
cal500	0.260	0.411	**0.442**	0.378	0.303	0.398	0.150	0.397	0.432	0.378	**0.633**	0.347	**1.000**	0.489	0.515	0.378	0.199	0.468
corel5k	0.224	0.226	**0.229**	0.222	0.007	0.167	0.224	0.226	**0.227**	0.221	0.015	0.206	0.224	0.226	**0.230**	0.222	0.004	0.140
emotions	0.600	0.609	0.585	0.602	**0.617**	0.588	0.590	0.610	0.587	0.593	**0.625**	0.580	0.611	0.609	0.584	**0.613**	0.611	0.598
enron	0.551	0.563	0.557	**0.569**	0.383	0.526	0.550	0.559	0.556	**0.569**	0.566	0.522	0.552	0.569	0.560	**0.570**	0.290	0.530
EukaryoteGO	0.793	0.795	0.788	0.790	0.711	**0.795**	0.772	**0.778**	0.769	0.767	0.692	0.757	0.815	0.813	0.809	0.815	0.731	**0.837**
EukaryotePseAAC	0.275	**0.365**	0.361	0.252	0.004	0.253	0.248	**0.336**	0.330	0.224	0.005	0.225	0.309	**0.401**	0.399	0.287	0.004	0.289
flags	0.747	0.748	**0.755**	0.732	0.755	0.747	0.739	0.753	**0.756**	0.728	0.729	0.698	0.758	0.744	0.755	0.736	0.784	**0.807**
foodtruck	0.484	0.531	**0.543**	0.511	0.441	0.508	0.486	0.533	0.537	0.509	**0.727**	0.508	0.483	0.532	**0.555**	0.514	0.317	0.510
genbase	0.976	0.967	0.953	0.969	0.931	**0.979**	0.980	0.968	0.956	0.969	0.984	**0.993**	**0.972**	0.966	0.950	0.970	0.884	0.966
GnegativeGO	0.955	0.949	0.951	0.951	0.937	**0.956**	0.951	0.946	0.951	0.948	**0.958**	0.952	0.960	0.953	0.950	0.954	0.916	**0.961**
GnegativePseAAC	0.564	**0.590**	0.587	0.552	0.541	0.524	0.511	**0.585**	0.582	0.510	0.553	0.480	**0.630**	0.595	0.593	0.602	0.529	0.576
GpositiveGO	0.935	0.934	0.941	**0.943**	0.935	0.936	0.922	0.935	**0.943**	0.933	0.938	0.926	0.950	0.933	0.939	**0.954**	0.931	0.947
GpositivePseAAC	0.558	0.569	0.567	0.544	**0.580**	0.534	0.501	0.556	0.556	0.495	**0.580**	0.492	**0.629**	0.584	0.580	0.604	0.579	0.587
HumanGO	0.744	0.739	0.730	**0.746**	0.691	0.744	0.722	0.721	0.711	**0.725**	0.675	0.719	0.768	0.759	0.751	0.769	0.708	**0.770**
HumanPseAAC	0.270	**0.368**	0.357	0.265	0.023	0.266	0.246	**0.350**	0.330	0.238	0.025	0.239	0.301	0.389	**0.389**	0.299	0.021	0.301
langlog	0.172	0.159	0.151	**0.179**	0.012	0.130	0.153	0.139	0.137	**0.154**	0.013	0.113	0.197	0.186	0.170	**0.213**	0.011	0.154
medical	**0.781**	0.757	0.741	0.770	0.334	0.762	0.756	0.742	0.726	0.750	0.375	**0.759**	**0.807**	0.775	0.758	0.792	0.301	0.766
ng20	0.546	**0.584**	0.545	0.526	0.313	0.563	0.488	**0.550**	0.498	0.468	0.316	0.515	0.622	**0.622**	0.601	0.600	0.310	0.620
ohsumed	**0.522**	0.517	0.513	0.519	0.264	0.461	0.500	0.511	**0.512**	0.498	0.349	0.447	**0.547**	0.523	0.514	0.542	0.212	0.475
PlantGO	0.766	0.784	**0.788**	0.758	0.715	0.753	0.736	0.769	**0.777**	0.733	0.710	0.724	**0.799**	0.799	0.799	0.785	0.720	0.786
PlantPseAAC	0.236	0.270	**0.306**	0.226	0.055	0.224	0.210	0.253	**0.305**	0.199	0.057	0.194	0.270	0.291	**0.309**	0.262	0.053	0.266
reutersk500	0.424	**0.426**	0.401	0.417	0.002	0.414	0.398	0.404	0.376	0.389	0.003	**0.415**	**0.453**	0.450	0.429	0.448	0.002	0.414
scene	0.587	0.591	0.603	0.582	**0.606**	0.575	0.545	0.577	0.591	0.538	**0.626**	0.532	**0.637**	0.605	0.617	0.634	0.587	0.626
slashdot	**0.510**	0.446	0.460	0.478	0.072	0.388	**0.474**	0.426	0.447	0.443	0.078	0.400	**0.552**	0.469	0.474	0.519	0.066	0.378
stackex_chess	**0.354**	0.270	0.261	0.327	0.006	0.277	**0.345**	0.257	0.257	0.318	0.010	0.293	**0.366**	0.286	0.266	0.337	0.004	0.263
tmc2007_500	0.661	0.636	0.632	0.660	0.552	**0.702**	0.659	0.628	0.633	0.656	0.671	**0.694**	0.664	0.643	0.632	0.664	0.469	**0.710**
VirusGO	0.865	0.886	0.884	0.888	0.844	**0.912**	0.830	0.868	**0.880**	0.874	0.847	0.869	0.908	0.906	0.890	0.906	0.842	**0.960**
VirusPseAAC	0.341	0.401	0.385	0.348	0.387	**0.426**	0.325	0.380	0.397	0.324	**0.420**	0.393	0.361	0.427	0.385	0.377	0.361	**0.468**
yeast	0.568	0.621	**0.624**	0.567	0.585	0.587	0.567	0.621	0.624	0.565	**0.696**	0.562	0.569	0.621	**0.624**	0.570	0.507	0.615
Yelp	0.741	0.709	0.696	0.741	0.670	**0.746**	0.722	0.696	0.681	0.721	0.705	**0.731**	0.762	0.723	0.712	**0.763**	0.639	0.760
Average	0.549	**0.562**	0.558	0.548	0.410	0.542	0.525	**0.549**	0.548	0.528	0.453	0.523	**0.600**	0.579	0.573	0.573	0.388	0.566
Rank. Sum.	95	**89**	104	106	165	113	106	95	**94**	120	128	129	**73**	100	116	92	183	108

Table 5: Example-based: F1, Precision and Recall results.

	F1						Precison						Recall					
	F2H-L	**F2H-G**	**T-G**	**T-L**	**Flat**	**HOM**	**F2H-L**	**F2H-G**	**T-G**	**T-L**	**Flat**	**HOM**	**F2H-L**	**F2H-G**	**T-G**	**T-L**	**Flat**	**HOM**
bibtex	**0.389**	0.381	0.373	0.384	0.073	0.343	0.409	**0.413**	0.392	0.404	0.150	0.382	**0.441**	0.420	0.424	0.439	0.051	0.373
birds	**0.211**	0.170	0.150	0.185	0.015	0.210	0.219	0.178	0.163	0.197	0.022	**0.224**	**0.237**	0.193	0.168	0.204	0.012	0.227
cal500	0.259	0.407	**0.439**	0.373	0.308	0.394	0.150	0.401	0.432	0.381	**0.633**	0.350	**1.000**	0.493	0.520	0.381	0.207	0.472
corel5k	0.208	0.212	**0.217**	0.206	0.007	0.154	**0.225**	0.224	0.222	0.222	0.015	0.211	0.224	0.226	**0.228**	0.223	0.004	0.139
emotions	0.561	0.574	0.550	0.567	**0.582**	0.556	0.592	0.601	0.580	0.600	**0.615**	0.595	0.599	0.608	0.582	**0.609**	0.607	0.594
enron	0.551	0.561	0.558	**0.569**	0.392	0.530	0.586	0.579	0.578	**0.601**	0.542	0.565	0.575	0.584	0.584	**0.593**	0.325	0.555
EukaryotoGO	0.800	0.802	0.794	0.798	0.708	**0.809**	0.795	0.798	0.788	0.792	0.700	**0.799**	0.831	0.831	0.827	0.831	0.740	**0.851**
EukaryotePseAAC	0.256	**0.334**	0.330	0.232	0.005	0.237	0.243	**0.316**	0.307	0.217	0.005	0.223	0.311	0.389	**0.390**	0.289	0.004	0.289
flags	0.719	0.717	0.734	0.699	**0.738**	0.724	0.729	0.734	**0.741**	0.717	0.719	0.692	0.736	0.717	0.747	0.710	0.780	**0.793**
foodtruck	0.491	0.526	**0.534**	0.507	0.498	0.517	0.537	0.575	0.557	0.541	**0.727**	0.568	0.573	0.617	**0.640**	0.598	0.429	0.607
genbase	**0.983**	0.980	0.974	0.981	0.953	0.983	0.989	0.986	0.980	0.985	0.985	**0.993**	**0.984**	0.982	0.977	0.982	0.938	0.979
GnegativeGO	0.963	0.956	0.957	0.958	0.943	**0.964**	0.965	0.959	0.962	0.960	0.958	**0.966**	0.970	0.963	0.961	0.964	0.936	**0.971**
GnegativePseAAC	0.579	**0.595**	0.593	0.565	0.545	0.541	0.557	**0.599**	0.598	0.548	0.553	0.528	**0.639**	0.604	0.602	0.612	0.541	0.584
GpositiveGO	0.940	0.936	0.942	**0.949**	0.936	0.940	0.936	0.938	0.944	**0.946**	0.938	0.937	0.951	0.936	0.942	**0.957**	0.935	0.948
GpositivePseAAC	0.566	0.571	0.570	0.548	**0.580**	0.540	0.536	0.564	0.564	0.522	**0.582**	0.519	**0.632**	0.587	0.583	0.607	0.580	0.588
HumanGO	0.756	0.751	0.740	0.759	0.696	**0.762**	0.754	0.749	0.736	0.754	0.690	**0.761**	0.797	0.788	0.780	0.800	0.735	**0.800**
HumanPseAAC	0.255	**0.352**	0.330	0.247	0.021	0.248	0.244	**0.346**	0.313	0.233	0.025	0.236	0.306	**0.391**	0.385	0.307	0.020	0.304
langlog	**0.152**	0.146	0.117	0.145	0.008	0.113	**0.149**	0.147	0.110	0.139	0.013	0.108	0.188	0.177	0.155	**0.193**	0.006	0.144
medical	**0.785**	0.765	0.747	0.780	0.328	0.769	**0.783**	0.769	0.752	0.781	0.375	0.782	**0.819**	0.795	0.780	0.810	0.305	0.783
ng20	0.550	**0.591**	0.552	0.531	0.311	0.572	0.521	**0.576**	0.529	0.502	0.312	0.552	0.623	**0.624**	0.603	0.602	0.312	0.623
ohsumed	0.503	**0.511**	0.501	0.501	0.276	0.450	0.513	**0.537**	0.526	0.514	0.345	0.473	**0.574**	0.560	0.549	0.571	0.249	0.505
PlantGO	0.780	0.792	**0.796**	0.774	0.718	0.770	0.772	0.791	**0.797**	0.770	0.718	0.764	0.812	0.810	**0.812**	0.799	0.732	0.798
PlantPseAAC	0.224	0.266	**0.305**	0.209	0.054	0.210	0.210	0.260	**0.307**	0.189	0.057	0.189	0.271	0.295	**0.314**	0.264	0.053	0.271
reutersk500	0.430	**0.448**	0.415	0.431	0.002	0.433	0.417	0.438	0.392	0.421	0.003	**0.444**	0.507	**0.510**	0.488	0.501	0.002	0.462
scene	0.587	0.597	0.607	0.584	**0.610**	0.577	0.569	0.598	0.608	0.565	**0.627**	0.558	**0.644**	0.616	0.627	0.643	0.603	0.635
slashdot	**0.508**	0.457	0.467	0.478	0.071	0.389	**0.493**	0.457	0.473	0.465	0.078	0.400	**0.570**	0.492	0.496	0.539	0.068	0.399
stackex_chess	**0.332**	0.259	0.236	0.309	0.005	0.261	**0.373**	0.305	0.263	0.345	0.010	0.314	**0.374**	0.294	0.274	0.345	0.003	0.275
tmc2007_500	0.648	0.625	0.620	0.648	0.531	**0.698**	0.681	0.653	0.646	0.678	0.661	**0.729**	0.695	0.674	0.669	0.699	0.505	**0.742**
VirusGO	0.872	0.900	0.896	0.902	0.843	**0.920**	0.862	0.903	**0.906**	0.906	0.855	0.904	0.910	0.928	0.914	0.924	0.853	**0.960**
VirusPseAAC	0.322	0.373	0.369	0.326	0.384	**0.419**	0.313	0.376	0.391	0.314	**0.419**	0.407	0.371	0.423	0.383	0.387	0.376	**0.485**
yeast	0.540	0.602	**0.607**	0.540	0.556	0.566	0.571	0.629	0.626	0.580	**0.703**	0.592	0.573	0.622	**0.629**	0.572	0.504	0.615
Yelp	0.710	0.678	0.660	0.710	0.631	**0.717**	0.736	0.708	0.690	0.736	0.695	**0.745**	0.742	0.707	0.693	**0.742**	0.627	0.742
Average	0.539	**0.553**	0.549	0.538	0.410	0.536	0.538	**0.561**	0.554	0.542	0.453	0.541	**0.604**	0.585	0.582	0.579	0.400	0.573
Rank. Sum.	98	**87**	105	112	163	107	103	**86**	109	116	146	112	**82**	94	114	91	184	107

Table 6: Example-based: Hamming-loss, Subset-accuracy and Accuracy results.

	Hamming-Loss						Subset-Accuracy						Accuracy					
	F2H-L	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**	**F2H-L**	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**	**F2H-L**	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**
bibtex	**0.019**	0.019	0.019	0.019	0.020	0.020	0.116	**0.140**	0.124	0.117	0.004	0.116	0.310	**0.313**	0.302	0.305	0.051	0.275
birds	0.086	0.093	0.098	0.092	0.104	**0.085**	0.060	0.045	0.022	0.046	0.005	**0.070**	0.170	0.136	0.114	0.147	0.012	**0.171**
cal500	0.850	0.240	0.230	0.186	**0.137**	0.212	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.150	0.261	**0.289**	0.234	0.186	0.250
corel5k	0.015	0.015	0.015	0.015	**0.012**	0.013	0.003	0.003	**0.004**	0.002	0.000	0.002	0.136	0.140	**0.143**	0.134	0.004	0.101
emotions	0.253	0.244	0.258	0.252	**0.236**	0.261	0.185	0.248	0.226	0.196	**0.284**	0.196	0.468	0.491	0.468	0.473	**0.506**	0.462
enron	0.057	0.056	0.057	**0.055**	0.059	0.061	0.102	**0.120**	0.103	0.112	0.078	0.108	0.431	0.445	0.438	**0.451**	0.304	0.413
EukaryotoGO	0.022	**0.022**	0.023	0.023	0.031	0.022	0.695	**0.698**	0.690	0.691	0.609	0.687	0.774	0.775	0.768	0.771	0.683	**0.778**
EukaryotePseAAC	0.085	**0.073**	0.074	0.089	0.097	0.089	0.124	**0.181**	0.173	0.114	0.004	0.125	0.220	**0.295**	0.290	0.200	0.004	0.206
flags	0.247	0.242	**0.236**	0.261	0.246	0.265	0.171	**0.258**	0.222	0.155	0.207	0.164	0.605	0.612	0.624	0.584	**0.629**	0.611
foodtruck	0.196	0.180	0.178	0.188	**0.153**	0.189	0.119	0.113	0.099	0.104	**0.282**	0.135	0.385	0.414	0.416	0.396	**0.429**	0.409
genbase	0.002	0.003	0.004	0.003	0.006	**0.002**	0.949	0.930	0.932	0.941	0.887	**0.958**	0.976	0.971	0.965	0.973	0.937	**0.977**
GnegativeGO	0.012	0.013	0.013	0.013	0.016	**0.011**	0.927	0.920	0.925	0.923	0.913	**0.927**	0.954	0.947	0.949	0.949	0.936	**0.955**
GnegativePseAAC	0.127	**0.108**	0.109	0.128	0.117	0.137	0.454	**0.545**	0.542	0.457	0.529	0.449	0.546	**0.583**	0.581	0.537	0.541	0.517
GpositiveGO	0.033	0.033	0.030	**0.029**	0.033	0.033	0.917	0.929	**0.935**	0.927	0.931	0.921	0.935	0.934	0.940	**0.943**	0.935	0.935
GpositivePseAAC	0.252	0.223	0.223	0.255	**0.212**	0.257	0.443	0.538	0.542	0.435	**0.573**	0.449	0.534	0.563	0.563	0.520	**0.578**	0.516
HumanGO	0.045	0.045	0.047	**0.044**	0.054	0.045	0.606	0.611	0.591	**0.619**	0.565	0.617	0.718	0.716	0.702	0.723	0.663	**0.725**
HumanPseAAC	0.137	**0.113**	0.119	0.140	0.153	0.140	0.124	**0.213**	0.170	0.117	0.015	0.124	0.219	**0.316**	0.289	0.211	0.020	0.214
langlog	0.030	0.031	0.030	0.031	**0.029**	0.032	0.066	**0.068**	0.031	0.047	0.001	0.053	**0.126**	0.123	0.092	0.116	0.006	0.095
medical	**0.013**	0.014	0.015	0.013	0.033	0.013	0.645	0.623	0.590	0.652	0.239	**0.657**	**0.750**	0.729	0.707	0.748	0.305	0.741
ng20	0.053	**0.046**	0.052	0.056	0.070	0.050	0.424	**0.519**	0.453	0.411	0.306	0.480	0.517	**0.572**	0.527	0.498	0.310	0.548
ohsumed	0.072	0.071	**0.071**	0.073	0.086	0.080	0.190	**0.200**	0.182	0.193	0.166	0.172	0.420	**0.427**	0.416	0.419	0.246	0.374
PlantGO	0.044	0.040	**0.039**	0.045	0.052	0.046	0.686	0.716	**0.726**	0.693	0.662	0.690	0.756	0.773	**0.778**	0.753	0.704	0.749
PlantPseAAC	0.156	0.140	**0.125**	0.162	0.164	0.166	0.124	0.188	**0.259**	0.112	0.048	0.118	0.197	0.245	**0.293**	0.182	0.053	0.184
reutersk500	0.017	0.017	0.018	0.018	0.024	**0.017**	0.227	0.268	0.225	0.246	0.001	**0.293**	0.373	**0.400**	0.364	0.380	0.002	0.395
scene	0.160	0.150	0.145	0.163	**0.137**	0.165	0.444	0.514	0.524	0.443	**0.574**	0.439	0.550	0.576	0.586	0.547	**0.601**	0.542
slashdot	**0.057**	0.063	0.060	0.061	0.092	0.064	**0.336**	0.324	0.334	0.316	0.058	0.306	**0.462**	0.422	0.432	0.435	0.068	0.368
stackex_chess	**0.014**	0.016	0.016	0.015	0.015	0.015	0.045	0.028	0.019	**0.046**	0.000	0.030	**0.247**	0.186	0.166	0.228	0.003	0.190
tmc2007_500	0.069	0.074	0.074	0.069	0.077	**0.061**	0.252	0.256	0.230	0.245	0.201	**0.353**	0.551	0.532	0.522	0.549	0.443	**0.614**
VirusGO	0.057	0.048	0.048	0.047	0.063	**0.038**	0.774	0.781	0.781	0.800	0.749	**0.827**	0.848	0.870	0.867	0.876	0.819	**0.897**
VirusPseAAC	0.282	0.259	0.244	0.287	**0.232**	0.256	0.169	0.168	0.232	0.160	**0.286**	0.245	0.283	0.320	0.334	0.283	0.359	**0.372**
yeast	0.262	0.230	0.228	0.263	**0.217**	0.262	0.060	**0.149**	0.118	0.073	0.114	0.074	0.420	**0.490**	0.489	0.421	0.443	0.445
Yelp	0.174	0.194	0.204	0.174	0.206	**0.170**	0.421	0.385	0.358	0.421	0.358	**0.462**	0.642	0.608	0.587	0.642	0.564	**0.656**
Average	0.120	0.094	**0.093**	0.100	0.096	0.100	0.337	**0.364**	0.355	0.335	0.300	0.348	0.485	**0.503**	0.497	0.484	0.380	0.485
Rank. Sum.	122	123	**124**	108	90	105	114	**79**	101	117	148	98	105	**85**	104	115	157	106

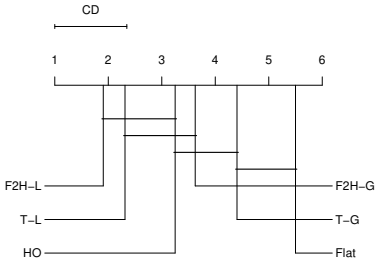
Table 7: Ranking-based: one-error, Average-precision and coverage results.

	One-Error						Average-precision						Coverage					
	F2H-L	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**	**F2H-L**	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**	**F2H-L**	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**
bibtex	0.596	0.594	0.612	0.597	0.850	**0.591**	**0.377**	0.370	0.364	0.373	0.091	0.348	72.85	72.86	**71.86**	72.40	96.26	76.87
birds	0.767	0.814	0.835	0.808	0.978	**0.767**	0.479	0.425	0.389	0.436	0.237	**0.483**	7.81	8.08	8.61	8.09	9.91	**7.77**
cal500	0.904	0.407	0.422	0.447	**0.366**	0.630	0.250	0.341	0.381	0.343	**0.386**	0.288	151.78	151.71	151.45	**151.05**	151.68	157.43
corel5k	**0.733**	0.754	0.776	0.755	0.985	0.785	**0.209**	0.206	0.200	0.203	0.107	0.147	**165.28**	166.07	165.71	166.67	173.74	262.40
emotions	0.440	0.433	0.443	0.428	**0.393**	0.408	0.683	0.706	0.691	0.692	**0.717**	0.689	2.52	**2.25**	2.33	2.44	2.27	2.57
enron	0.338	0.304	**0.291**	0.319	0.389	0.493	0.544	0.563	0.562	**0.565**	0.413	0.500	23.91	23.75	24.64	**23.09**	29.42	28.37
EukaryoteGO	0.205	**0.201**	0.210	0.209	0.301	0.201	0.829	0.828	0.824	0.827	0.750	**0.848**	2.14	2.25	2.23	2.12	3.26	**1.66**
EukaryotePseAAC	0.757	**0.691**	0.701	0.791	0.995	0.779	0.350	**0.418**	0.416	0.328	0.142	0.342	8.17	7.10	**6.99**	8.32	11.09	8.43
flags	**0.205**	**0.205**	**0.205**	0.210	0.211	0.299	**0.811**	0.809	0.810	0.801	0.801	0.786	3.80	3.76	**3.75**	3.81	3.78	3.87
foodtruck	0.310	0.277	**0.273**	0.309	**0.273**	0.503	0.683	0.716	**0.720**	0.698	0.663	0.591	4.81	4.53	**4.37**	4.65	5.91	5.19
genbase	0.008	0.014	0.015	0.011	0.015	**0.005**	0.984	0.979	0.977	0.982	0.946	**0.985**	0.81	0.90	0.89	0.80	1.81	**0.57**
GnegativeGO	0.039	0.043	0.043	0.044	0.042	**0.033**	0.973	0.969	0.970	0.969	0.959	**0.977**	0.15	0.18	0.17	0.18	0.31	**0.14**
GnegativePseAAC	0.447	**0.402**	0.404	0.454	0.447	0.471	0.699	**0.716**	0.713	0.688	0.688	0.644	**1.36**	1.39	1.44	1.46	1.55	2.01
GpositiveGO	0.062	0.062	0.056	**0.054**	0.062	0.062	0.960	0.958	0.962	**0.966**	0.959	0.961	0.14	0.15	0.14	**0.12**	0.15	0.13
GpositivePseAAC	0.470	0.439	0.439	0.487	**0.420**	0.493	0.698	0.717	0.718	0.687	**0.723**	0.684	0.96	0.93	**0.93**	1.00	0.94	1.00
HumanGO	0.256	0.262	0.272	0.257	0.322	**0.238**	0.806	0.800	0.791	**0.808**	0.754	0.806	1.68	1.75	1.86	**1.68**	2.20	2.12
HumanPseAAC	0.766	**0.664**	0.694	0.775	0.975	0.758	0.386	**0.465**	0.450	0.384	0.226	0.381	5.51	4.71	**4.68**	5.51	7.30	5.66
langlog	**0.865**	0.866	0.913	0.875	0.987	0.895	**0.227**	0.221	0.191	0.222	0.087	0.187	30.30	31.51	31.83	**29.57**	37.85	34.66
medical	0.218	**0.209**	0.235	0.217	0.625	0.219	**0.801**	0.787	0.768	0.797	0.346	0.781	**5.47**	7.15	7.63	5.98	22.02	6.42
ng20	0.488	**0.423**	0.471	0.505	0.688	0.437	0.613	**0.650**	0.617	0.597	0.405	0.647	4.01	3.95	4.20	4.21	7.18	**3.79**
ohsumed	0.456	0.451	**0.450**	0.461	0.655	0.517	**0.563**	0.561	0.560	0.560	0.359	0.510	**8.21**	8.37	8.47	8.26	12.07	9.34
PlantGO	0.228	**0.207**	0.207	0.231	0.279	0.234	0.823	**0.835**	0.835	0.816	0.778	0.819	1.29	1.25	**1.24**	1.41	1.72	1.38
PlantPseAAC	0.795	0.744	**0.697**	0.815	0.943	0.803	0.380	0.409	**0.436**	0.372	0.265	0.355	4.20	**4.17**	4.20	4.18	4.90	4.95
reutersk500	0.604	0.574	0.616	0.594	0.997	**0.547**	0.453	**0.474**	0.448	0.457	0.061	0.474	31.34	30.77	31.21	30.87	60.09	**30.48**
scene	0.415	0.394	0.382	0.421	**0.373**	0.440	0.700	0.709	**0.716**	0.698	0.715	0.696	1.45	1.46	1.42	1.44	1.47	**1.38**
slashdot	**0.499**	0.535	0.521	0.532	0.922	0.592	**0.561**	0.513	0.520	0.532	0.156	0.476	**6.46**	7.46	7.30	6.84	12.33	6.98
stackex_chess	**0.618**	0.690	0.710	0.640	0.990	0.685	**0.307**	0.237	0.226	0.285	0.027	0.240	**123.48**	132.77	134.86	127.41	160.77	132.37
tmc2007_500	0.326	0.334	0.360	0.343	0.355	**0.272**	0.687	0.669	0.651	0.677	0.579	**0.735**	**6.03**	6.51	6.86	6.25	8.68	6.15
VirusGO	0.129	**0.081**	0.091	0.083	0.133	0.106	0.907	**0.935**	0.923	0.933	0.891	0.934	0.63	0.55	0.62	0.55	0.73	**0.45**
VirusPseAAC	0.721	0.656	0.607	0.710	**0.581**	0.599	0.485	0.509	0.527	0.487	0.542	**0.572**	2.51	2.49	2.48	2.52	2.45	**2.13**
yeast	0.463	0.467	0.475	0.456	**0.332**	0.447	0.628	0.674	0.673	0.631	**0.707**	0.634	7.42	6.54	**6.42**	7.57	6.45	7.90
Yelp	**0.206**	0.243	0.277	0.207	0.285	0.230	**0.892**	0.870	0.848	0.892	0.839	0.868	1.27	1.36	1.42	**1.27**	1.47	1.41
Average	0.456	**0.426**	0.433	0.446	0.545	0.462	0.608	**0.618**	0.614	0.607	0.499	0.597	**22.145**	22.494	22.607	22.272	27.106	26.278
Rank. Sum.	123	**138**	110	107	78	111	94	**88**	106	104	160	120	**131**	124	125	128	57	107

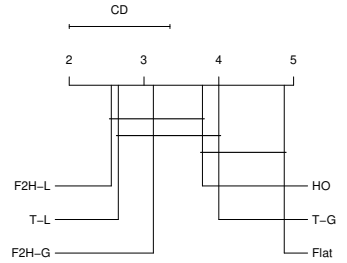
Table 8: Ranking-based: ranking-loss, MLP and WLP results.

	Ranking-Loss						MLP						WLP					
	F2H-L	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**	**F2H-L**	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**	**F2H-L**	**F2H-G**	**T-G**	**T-L**	**Flat**	**HO**
bibtex	0.292	0.298	0.291	**0.287**	0.445	0.317	0.016	0.043	0.139	0.020	0.987	0.007	0.201	0.273	0.315	0.237	0.987	0.236
birds	0.163	0.178	0.192	0.172	0.241	**0.162**	0.205	0.211	0.442	0.126	0.947	0.142	0.416	0.432	0.668	0.421	0.947	0.405
cal500	0.372	0.296	**0.267**	0.289	0.295	0.362	0.000	0.710	0.706	0.262	0.953	0.260	0.048	0.721	0.715	0.559	0.953	0.544
corel5k	**0.226**	0.227	0.229	0.226	0.247	0.404	0.719	0.856	0.868	0.780	0.997	0.731	0.872	0.888	0.903	0.874	0.997	0.892
emotions	0.293	0.263	0.279	0.287	**0.260**	0.301	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
enron	0.193	0.190	0.199	**0.185**	0.289	0.249	0.338	0.543	0.660	0.404	0.925	0.251	0.592	0.658	0.723	0.621	0.925	0.596
EukaryoteGO	0.082	0.086	0.087	0.082	0.135	**0.063**	0.064	0.141	0.200	0.064	0.495	0.036	0.082	0.186	0.241	0.082	0.514	0.077
EukaryotePseAAC	0.357	0.311	**0.306**	0.365	0.496	0.372	0.118	0.650	0.686	0.064	0.945	0.000	0.473	0.714	0.741	0.505	0.945	0.473
flags	0.215	0.218	**0.211**	0.227	0.223	0.259	0.071	0.043	0.100	0.029	0.143	0.029	0.100	0.057	0.100	0.071	0.143	0.057
foodtruck	0.228	0.204	**0.199**	0.213	0.293	0.270	0.208	0.433	0.700	0.233	0.917	0.125	0.533	0.625	0.717	0.558	0.917	0.492
genbase	0.011	0.013	0.013	0.010	0.033	**0.008**	0.330	0.330	0.347	0.306	0.519	0.337	0.333	0.356	0.384	0.347	0.519	0.337
GnegativeGO	0.013	0.017	0.015	0.017	0.031	**0.012**	0.025	0.025	0.025	0.025	0.250	0.063	0.025	0.025	0.025	0.025	0.250	0.063
GnegativePseAAC	**0.185**	0.188	0.194	0.197	0.212	0.277	0.088	0.325	0.375	0.088	0.600	0.075	0.213	0.350	0.388	0.275	0.613	0.263
GpositiveGO	0.044	0.048	0.044	**0.035**	0.047	0.041	0.000	0.000	0.000	0.000	0.000	0.000	0.025	0.025	0.025	0.025	0.025	0.025
GpositivePseAAC	0.320	0.308	**0.307**	0.332	0.309	0.331	0.025	0.225	0.250	0.050	0.100	0.050	0.125	0.225	0.250	0.200	0.225	0.225
HumanGO	0.098	0.104	0.112	**0.098**	0.139	0.127	0.043	0.093	0.100	0.043	0.300	0.043	0.093	0.121	0.143	0.093	0.336	0.107
HumanPseAAC	0.383	**0.327**	0.328	0.381	0.514	0.398	0.100	0.493	0.536	0.029	0.929	0.007	0.364	0.586	0.600	0.350	0.929	0.386
langlog	0.296	0.309	0.314	**0.288**	0.387	0.350	0.617	0.649	0.716	0.508	0.987	0.413	0.812	0.825	0.880	0.813	0.987	0.837
medical	**0.093**	0.123	0.131	0.102	0.435	0.112	0.536	0.567	0.613	0.536	0.956	0.548	0.596	0.640	0.667	0.609	0.956	0.652
ng20	0.207	0.203	0.217	0.217	0.372	**0.195**	0.000	0.000	0.000	0.000	0.140	0.000	0.000	0.000	0.000	0.000	0.140	0.000
ohsumed	**0.256**	0.258	0.260	0.258	0.438	0.300	0.000	0.074	0.130	0.009	0.583	0.000	0.043	0.135	0.174	0.087	0.583	0.026
PlantGO	0.103	**0.098**	0.098	0.112	0.142	0.111	0.058	0.058	0.067	0.058	0.417	0.075	0.083	0.075	0.083	0.075	0.417	0.125
PlantPseAAC	0.365	**0.361**	0.363	0.364	0.428	0.436	0.125	0.383	0.750	0.042	0.917	0.008	0.442	0.525	0.758	0.400	0.917	0.375
reutersk500	0.243	0.243	0.249	0.239	0.537	**0.229**	0.414	0.465	0.471	0.327	0.990	0.340	0.499	0.554	0.564	0.492	0.990	0.532
scene	0.268	0.269	0.260	0.266	0.270	**0.255**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
slashdot	**0.270**	0.316	0.306	0.287	0.553	0.297	0.223	0.245	0.273	0.227	0.836	0.241	0.264	0.314	0.341	0.286	0.836	0.405
stackex_chess	**0.342**	0.387	0.396	0.358	0.545	0.373	0.613	0.682	0.767	0.655	0.996	0.631	0.792	0.871	0.889	0.816	0.996	0.852
tmc2007_500	0.143	0.163	0.174	0.151	0.245	**0.143**	0.000	0.000	0.018	0.000	0.336	0.000	0.000	0.000	0.023	0.000	0.341	0.000
VirusGO	0.076	0.055	0.065	0.057	0.089	**0.045**	0.017	0.050	0.083	0.050	0.083	0.033	0.033	0.083	0.083	0.067	0.083	0.033
VirusPseAAC	0.455	0.436	0.435	0.459	0.429	**0.375**	0.083	0.183	0.617	0.033	0.333	0.033	0.417	0.350	0.683	0.400	0.500	0.267
yeast	0.271	0.222	0.218	0.275	**0.207**	0.282	0.071	0.100	0.286	0.014	0.257	0.029	0.093	0.164	0.286	0.079	0.271	0.107
Yelp	0.105	0.126	0.142	**0.105**	0.153	0.141	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Average	0.221	**0.217**	0.218	0.221	0.300	0.241	0.165	0.277	0.352	0.161	0.575	0.145	0.276	0.348	0.399	0.302	0.588	0.303
Rank. Sum.	**132**	124	119	130	58	109	129	88	59	129	36	**139**	**140**	93	61	129	35	124

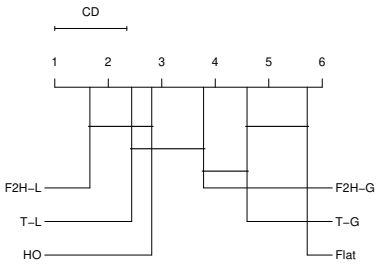
Figures 5, 6 and 7 show the critical diagrams for each individual measure and their respective p-values obtained from the corrected Friedman test with the Nemenyi post-hoc with significance level of $\alpha = 0.05$.



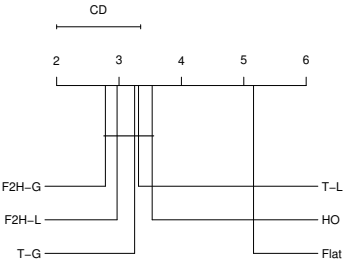
(a) macro-F1 ($p = 5.55 \times 10^{-16}$)



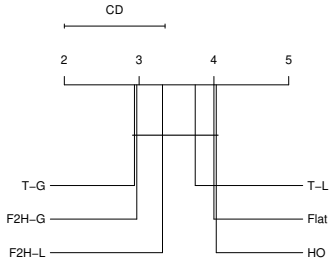
(b) macro-precision ($p = 8.96 \times 10^{-7}$)



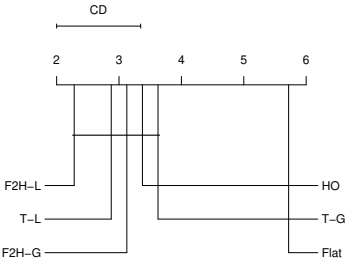
(c) macro-recall ($p = 0.00$)



(d) micro-F1 ($p = 3.30 \times 10^{-6}$)



(e) micro-precision ($p = 4.69 \times 10^{-2}$)

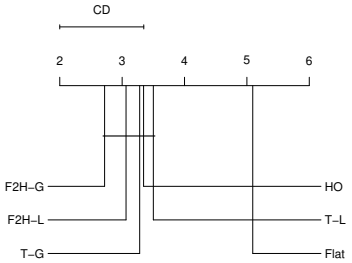


(f) micro-recall ($p = 2.05 \times 10^{-12}$)

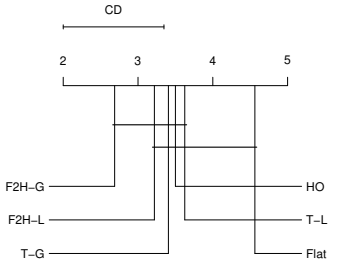
Fig. 5: Critical diagrams for individual measures (Part 1).

4.3 Execution Times

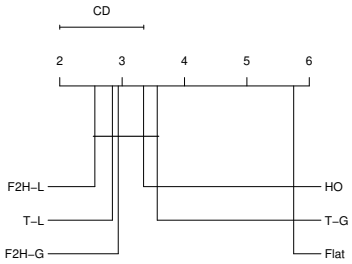
With regard to execution times, Table 9 presents the average time taken by each classifier to run the classification in all 32 datasets. As mentioned earlier, most of the executed classifiers had no support to multithreading or parallel



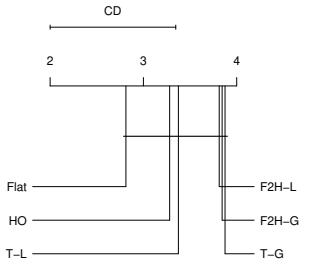
(a) F1 ($p = 8.49 \times 10^{-6}$)



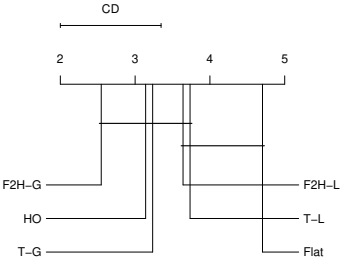
(b) precision ($p = 3.95 \times 10^{-3}$)



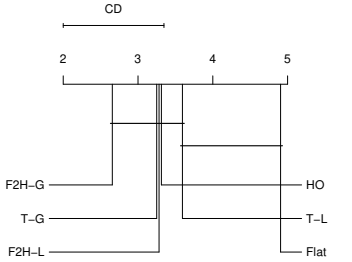
(c) recall ($p = 6.20 \times 10^{-12}$)



(d) hamming-loss ($p = 1.47 \times 10^{-1}$)



(e) subset-accuracy ($p = 2.12 \times 10^{-4}$)

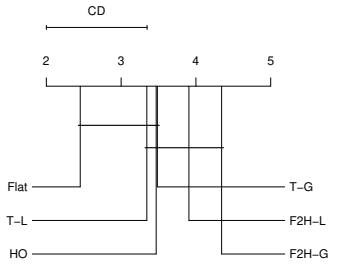


(f) accuracy ($p = 8.92 \times 10^{-5}$)

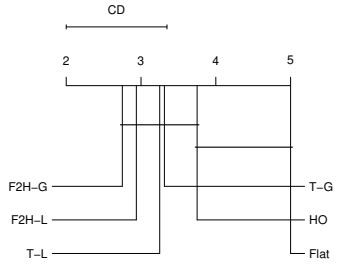
Fig. 6: Critical diagrams for individual measures (Part 2).

execution, and average times presented here are simply the time taken by the classifier from its start until the end.

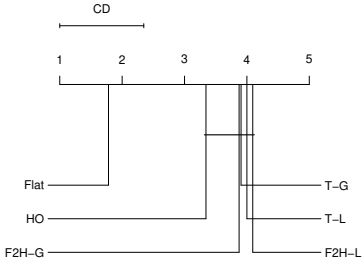
In Table 10, the time taken by each F2H main steps is accounted along with the number of CFLs (which is equivalent to the number of nodes) and the number of edges in the label hierarchy. The obtained values show that most of time consumption of F2H is regarded to the HMC or the HSC step (ClusHMC and ClusHSC respectively). This is explained by the fact that the Clus step does the biggest part of the job and also by the absence of parallel support in the Clus implementations. Moreover, the step of finding the CFLs is not representative in the final times; besides the best time execution of the



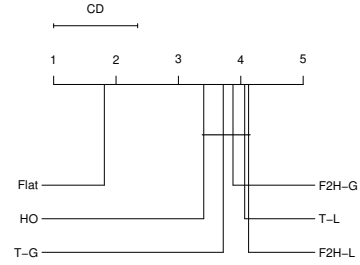
(a) one-error ($p = 2.62 \times 10^{-3}$)



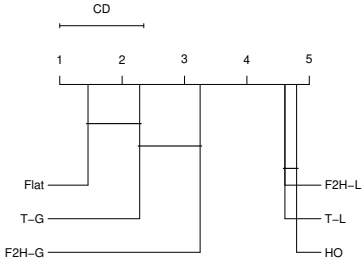
(b) average-precision ($p = 1.42 \times 10^{-5}$)



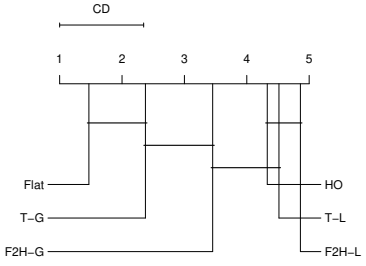
(c) coverage ($p = 1.17 \times 10^{-6}$)



(d) ranking-loss ($p = 2.07 \times 10^{-6}$)



(e) mlp ($p = 0.00$)



(f) wlp ($p = 3.33 \times 10^{-15}$)

Fig. 7: Critical diagrams for individual measures (Part 3).

PCBO algorithm and the use of the threshold τ_{CFL} in some datasets, this is also explained by the fact that, even though this step could have an exponential behavior in the worst case, it rarely happens in most of multi-label datasets used in the experiments. Finally, in datasets with large number of CFLs, the execution time was higher (i.e., cal500 and bibtex); this is also true for datasets with large number of instances and/or attributes (i.e., EukaryoteGO and ng20).These findings are explained considering that, the bigger the number of instances, attributes, and nodes in the label hierarchy for a given dataset, the bigger the search space in the classification task. Finally, there is

Table 9: Average time (in seconds) taken by each classifier in the experiments.

Dataset	Flat	T-G	T-L	F2H-G	F2H-L	HO
bibtex	6204.5	**1460.2**	18253.4	3112.8	100178.8	6586.5
birds	67.2	9.2	33.2	45.7	7755.5	**3.6**
cal500	225.2	10.4	108.0	5238.2	14503.0	**7.3**
corel5k	5755.0	**329.4**	6993.8	1067.2	20438.7	359.6
emotions	10.8	3.0	10.9	12.4	28.8	**2.8**
enron	188.1	76.7	999.6	182.4	6573.6	**40.6**
EukaryoteGO	4863.2	**2320.9**	11505.1	2903.8	16484.6	2377.6
EukaryotePseAAC	2130.6	**277.4**	2136.9	535.5	7095.6	294.7
flags	**0.7**	1.2	2.5	8.1	11.3	2.5
foodtruck	**0.9**	1.4	3.7	11.1	24.0	2.8
genbase	43.7	27.6	97.0	46.1	107.0	**5.8**
GnegativeGO	40.6	38.6	101.2	58.8	146.6	**6.1**
GnegativePseAAC	102.9	26.6	127.7	45.3	198.9	**5.0**
GpositiveGO	3.8	4.4	10.0	14.8	17.4	**3.3**
GpositivePseAAC	20.5	7.4	28.1	18.5	38.8	**3.2**
HumanGO	885.4	515.9	1874.3	736.9	3373.5	**130.1**
HumanPseAAC	562.7	85.7	578.9	170.5	1538.6	**22.8**
langlog	180.6	38.3	218.4	81.4	473.4	**21.3**
medical	113.8	52.8	419.8	79.5	617.3	**7.8**
ng20	2704.3	**1272.4**	7524.7	1741.1	15735.8	6296.8
ohsumed	1937.5	**805.1**	6373.9	1578.0	22193.1	9233.6
PlantGO	59.1	47.6	140.7	58.4	206.8	**8.1**
PlantPseAAC	117.5	17.4	109.8	33.9	234.1	**5.0**
reutersk500	955.8	181.1	1251.6	439.3	3812.7	**35.6**
scene	259.0	58.9	267.3	94.7	367.0	**11.6**
slashdot	530.8	184.1	1252.4	301.2	3217.2	**88.6**
stackex_chess	487.9	41.5	332.3	233.2	1613.6	**17.7**
tmc2007_500	1784.7	**1679.0**	20983.3	3192.4	103382.3	15146.6
VirusGO	**1.7**	2.2	5.0	11.2	11.1	3.0
VirusPseAAC	10.3	3.1	11.2	9.7	18.3	**2.9**
yeast	252.3	27.0	194.8	433.6	1755.3	**6.8**
Yelp	602.2	**513.3**	3010.2	711.2	7717.8	3493.1
Average	972.0	**316.3**	2655.0	725.2	10621.0	1382.3

a clear disadvantage of the F2H-L approach in execution times, and the reason is the use of the local hierarchical multi-label classifier ClusHSC that creates a classifier per edge of the label hierarchy; on the other hand, in the F2H-G, the global multi-label classifier ClusHMC induces just a single model.

5 Discussion

In an overall analysis spotting which approach was the winner on each measure (Tables 3-8), it is possible to verify that the F2H approaches performed better in 15 of the 18 measures used in experiments, with F2H-L winning in 8 measures and F2H-G in 7. The F2H approaches only did not obtained the best performances on micro-precision, Hamming-loss and MLP. This shows an advantage of the approaches using a label hierarchy structured as a DAG, but taking into account the statistical analysis presented in the critical diagrams (Figures 5-7), it is possible to verify that in all the mentioned cases, the difference to other classifiers is within the critical distance.

Another relevant point is the performance of the Flat classifier that presented the worst performance in all measures in the direct comparison, and also when analyzing the critical distance in the critical diagrams. This is an indication that when comparing PCT approaches that use a data-driven hierarchy of labels with approaches that only use flat label space, there is a clear advantage for the first group. Considering that the Flat approach used in this work is a global classifier, the results indicate that, although it induces a single classifier with all labels, the approach is having difficulties to explore the correlations between them. Moreover, is possible to claim (and reinforce the

Table 10: F2H: average 10-fold time(in seconds) taken by each F2H main step and the label hierarchy details.

	F2H Steps execution time in seconds								Hierarchy	
Dataset	**Find CFLs**	**Define Hierarchy**	**MLC to HMC**	**ClusHSC F2H-L**	**ClusHMC F2H-G**	**HMC to MLC**	**Total F2H-L**	**Total F2H-G**	*CFLs*	*Edges*
bibtex	1.6	32.6	187.1	97446.25	2084.0	807.5	100178.8	3112.8	5124.2	14460.7
birds	0.1	0.6	2.2	130.4	32.6	10.3	7755.5	45.7	137.9	295.3
cal500	3.9	1063.9	37.9	3287.5	3819.8	312.8	14503.0	5238.2	28225.0	120153.6
corel5k	0.3	12.2	27.7	19678.9	307.4	719.6	20438.7	1067.2	650.4	1175.3
emotions	0.1	0.6	0.6	23.2	6.9	4.2	28.8	12.4	28.8	55.6
enron	0.3	2.5	18.4	6493.8	102.6	58.6	6573.6	182.4	1336.8	3829.7
EukaryoteGO	0.4	1.1	838.7	15547.3	1966.4	97.2	16484.6	2903.8	106.4	202.9
EukaryotePseAAC	0.1	0.7	28.1	6986.4	426.3	80.4	7095.6	535.5	107.0	204.2
flags	0.1	0.5	0.2	8.4	5.2	2.2	11.3	8.1	74.8	203.3
foodtruck	0.1	0.5	0.4	17.4	4.6	5.5	24.0	11.1	212.1	620.6
genbase	0.1	0.8	9.4	85.4	24.4	11.4	107.0	46.1	36.4	45.0
GnegativeGO	0.1	0.4	17.5	120.4	32.5	8.4	146.6	58.8	20.0	29.0
GnegativePseAAC	0.0	0.4	5.1	184.6	31.0	8.8	198.9	45.3	20.0	29.0
GpositiveGO	0.1	0.3	3.6	10.2	7.7	3.1	17.4	14.8	8.5	10.0
GpositivePseAAC	0.0	0.3	2.0	30.9	10.6	5.6	38.8	18.5	8.6	10.2
HumanGO	0.3	1.5	252.1	3086.7	450.1	33.0	3373.5	736.9	83.3	168.8
HumanPseAAC	0.1	1.3	11.7	1498.1	130.0	27.4	1538.6	170.5	83.2	168.9
langlog	0.1	0.4	11.9	421.6	29.6	39.4	473.4	81.4	36.8	37.4
medical	0.1	0.5	11.2	583.7	45.9	21.8	617.3	79.5	90.7	138.0
ng20	0.1	1.0	140.0	15398.1	1403.4	196.7	15735.8	1741.1	26.0	30.0
ohsumed	0.6	8.3	168.1	21778.8	1163.7	237.2	22193.1	1578.0	1188.5	3652.7
PlantGO	0.1	0.4	22.7	176.3	27.9	7.3	206.8	58.4	32.1	49.2
PlantPseAAC	0.0	0.4	3.6	220.7	20.4	9.4	234.1	33.9	32.2	49.4
reutersk500	0.1	5.6	24.7	3547.4	174.0	234.8	3812.7	439.3	154.9	238.6
scene	0.1	0.8	7.4	345.2	72.9	13.5	367.0	94.7	16.4	24.6
slashdot	0.1	1.0	33.0	3128.2	212.2	54.9	3217.2	301.2	146.8	294.1
stackex_chess	0.4	3.2	12.6	1423.5	43.0	173.9	1613.6	233.2	1307.3	2800.7
tmc2007_500	1.8	23.6	276.8	102512.1	2322.2	567.9	103382.3	3192.4	1663.6	5974.8
VirusGO	0.2	0.9	1.4	5.7	5.8	2.9	11.1	11.2	18.0	28.3
VirusPseAAC	0.1	0.4	0.9	14.4	5.8	2.5	18.3	9.7	18.0	28.5
yeast	0.2	7.6	6.7	1703.7	382.0	37.2	1755.3	433.6	618.2	2149.3
Yelp	0.1	0.8	59.1	7612.1	605.5	45.7	7717.8	711.2	32.0	80.0
Average	**0.4**	**36.7**	**69.5**	**9797.1**	**498.6**	**120.0**	**10621.0**	**725.2**	**1301.4**	**4913.7**

conclusions of Madjarov et al (2019)) that the use of the label hierarchies created from a flat MLC label space can help to improve the classifiers results.

Analyzing the results obtained in the label-based macro-measures (Table 3 and Figures 5a, 5b and 5c) we can observe a superiority of the hierarchical approaches using local classifiers, especially the F2H-L, T-L and HO. An initial analysis shows that the local classifiers obtained better result in datasets with more imbalance in the label distribution (MeanIR greater than 10 according to Table 1), while the global classifiers (F2H-G, T-G and Flat) performed worst in imbalanced datasets. Macro-measures give the same weight to all classes in the evaluation, so this suggests that local classifiers achieve better results in imbalanced datasets. Moreover, there is a clear indication that when evaluating imbalanced datasets with macro-measures, the choice of a local or a global approach have more influence in results than choosing between a tree or a DAG-structured label hierarchy.

In the label-based micro-measures (Table 4 and Figures 5d, 5e and 5f), when comparing results of tree against DAG-structured approaches, there exists a little advantage to last ones which performed better on micro-F1 and micro-recall while the tree-structured performed better (almost tied) in the micro-precision. When compared the performance of local and global approaches, there is no winner, the global F2H-G and T-G performed better in micro-F1 and micro-precision where the local F2H-L did better on micro-recall.

Taking into account the six instance-based measures presented in Tables 5 and 6, there exists a predominance of the DAG-structured approaches which

performed best than tree-structured ones on five measures, showing that first ones have some advantage on this group of measures. Moreover, in this group of measures we found that global approaches performed better than local ones in almost five measures, the only exception was in recall where the local approaches performed better.

Another interesting finding when looking to recall measures (macro-recall, micro-recall and recall) is the performance of the local classifiers, considering the critical diagrams in the Figures 5c, 5f and 6c, the local F2H-L followed by the local T-L performed better in all scenarios. The general explanation for this finding is that as local approaches aforementioned define one classifier per edge in the label hierarchy they tend to have a higher sensitivity in the predictions.

When taking into account the ranking measures in Table 7 and Table 8 (for ranking-loss), and critical diagrams in Figures 7a-7d), there exists a predominance of the DAG-structured approach over tree-structured ones, with highlights to F2H ones (local and global) that performed better in all four measures. Confronting the local and global approaches, it is not possible to determine which was the best.

The two label problem measures MLP and WLP (Table 8 and critical diagrams in Figures 7e and 7f) show some interesting findings and are the measures in which we can see larger statistical support on critical diagrams. The MLP results reveals the local approaches performing better than global ones in a similar way to what happened in the label-based macro-measures. The MLP measures the proportion of labels never predicted by the classifiers and is possible to notice that there is a significant difference between the values of the local and global classifiers (i.e., for the HumanPseAAC that has 14 labels, the F2H-L never predicted 10% of labels, F2H-G 49.3%, T-G 53.6%, T-L 2.9%, Flat 92.9% and HO 0.7%). These results shows that local classifiers tend to predict a greater number of labels in relation to global ones and that greater coverage leads to better results on macro-measures and the recall ones where the local approaches performed better as stated before.

The WLP represents the case where a label might be predicted for some instances, but these predictions are always wrong and can be seen as a generalization of MLP. The results show a predominance of the local approaches in a similar way they appeared in the MLP and that is reasonable because WLP can be seen as an extension of MLP. It is also possible to notice that the HO approach that presented the best MLP was not the best on WLP. This suggests that among the investigated classifiers, the HO presents the more sensitivity leading lower precision and higher recall in the predictions. HO performed better than the global approaches on macro-recall and its behavior on WLP reinforces this conclusion.

Lastly, regarding execution times, the HO approach presented the best execution times in most of the datasets, especially those with a moderate number of instances; on the contrary, in datasets with a great number of instances, T-G

that uses a tree-structured label hierarchy and induces a unique global hierarchical multi-label classifier was faster. Moreover, the data presented in Table 9 and Table 10 give additional support to some general conclusions. When comparing the execution times of the DAG and tree-structured approaches, the last ones were faster, T-L outperformed F2H-L and T-G outperformed F2H-G. Finally, when comparing the execution times of F2H-L and F2H-G, the latter has a big advantage, and this is due to the fact that F2H-L creates one local classifier to each edge in the DAG-structured label hierarchy while the F2H-G approach creates just a single global classifier.

6 Conclusions and Future Works

In this paper, a new problem transformation method named F2H (Flat to Hierarchical) was proposed. It allows obtaining the predictions to a multi-label classification task through the use of a hierarchical multi-label classifier (HMC). The notion of closed frequent labelset (CFL), also presented, allows the definition of a label hierarchy structured as a directed acyclic graph to be used by the hierarchical multi-label classifier. Moreover, two different hierarchical multi-label strategies were proposed and tested within F2H: F2H-L that uses a local approach creating a classifier per node of the label hierarchy; and F2H-G that creates a single global classifier.

Regarding the closer related approaches, this paper presented a new method to create data-driven label hierarchies from flat label spaces, and proposed the use of DAG-structured label hierarchies to improve the representation of the label correlations. Also, an extensive experimental evaluation was performed in terms of the number of datasets used in the experiments, using 10-fold cross-validation.

The experiments using 32 popular multi-label classification datasets and 18 multi-label performance measures, showed, in a general assumption, that F2H using a DAG structured label hierarchy can produce competitive results when compared to related works using tree-structured ones. This suggests that DAG-represented label hierarchies through the notion of CFL are more efficient in capturing label correlations than tree-structured label hierarchies defined by balanced k-means.

Looking at each evaluation measure individually, and taking into account the statistical support in the comparisons, we can state that DAG-structured approaches presented very competitive results when compared to tree-structured ones. Also, results showed that the local approaches can achieve better performances when evaluated with macro-measures. Moreover, the local approaches are also more suitable to deal with label imbalance, while the global ones perform better on datasets with more balanced label distributions. F2H-L and the other local approaches using a hierarchical classifier tend to have more time consumption in spite of a better performance.

As future research, we suggest defining a method that dynamically calibrates the hierarchy depth and also the CFL minimum support threshold,

allowing the selection of only the most representative labelsets to compose the label hierarchy. The use of other clustering methods like k-means and self organizing maps to create a DAG-structured hierarchy that best represents the label space and its correlations can also be studied. We believe that improving the quality of the label hierarchy can lead to better results. Moreover, other HMC algorithms could be used and tested in the classification step, such as other approaches to obtain label hierarchies, like a label weighted selection that uses a measure to establish the relevance of each label. The application of F2H in other classification scenarios like large and/or sparse datasets, or even to adapt it to deal with HMC tasks (transform a HMC task in a new, more efficient HMC task) can also be investigated. Lastly, ensemble versions of F2H could be developed and evaluated against other commonly used ensemble classifiers to investigate how the F2H approaches behave in this context.

Declarations

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Author Contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Mauri Ferrandin and Ricardo Cerri. The first draft of the manuscript was written by Mauri Ferrandin and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data Availability

The datasets and algorithms used in the current study will be available in the GitHub repository https://github.com/maurisones/F2H after the publication approval.

Appendix A Multi-label Performance Measures

The measures and descriptions used in this work, and here represented, are based on Zhang and Zhou (2014). Sixteen different metrics were chosen and are grouped in three categories: 6 label-based measures (macro-F1, macro-precision, macro-recall, micro-F1, micro-precision and micro-recall; 6

example-based measures (F1, precision, recall, hamming-loss, subset-accuracy and accuracy); and 4 ranking-based measures (one-error, average-precision, coverage and ranking-loss). The chosen metrics are presented in Equations A1 to A16.

The label-based macro and micro metrics are computed taking into account the true positives (TP), false positives (FP) and false negatives (FN) for each single-label and using two different aggregation approaches: The macro approach computes one metric for each label and then average the values over all the categories, whereas the micro approach considers predictions of all instances together (aggregating the TP, FP and FN values of all labels) and then calculates the measure across all at last (Gibaja and Ventura, 2014). Formally we define, for a given label λ_j, $TP_j = \mid \{X_i \mid y_j \in \mathcal{Y} \wedge y_j \in h_{MLC}(X_i), 1 \leqslant i \leqslant m\} \mid$, $FP_j = \mid \{X_i \mid y_j \notin \mathcal{Y} \wedge y_j \in h_{MLC}(X_i), 1 \leqslant i \leqslant m\} \mid$ and $FN_j = \mid \{X_i \mid y_j \in \mathcal{Y} \wedge y_j \notin h_{MLC}(X_i), 1 \leqslant i \leqslant m\} \mid$.

The main difference between the label-based micro and macro metrics is that the former gives an equal weight for each individual label, whereas the latter first aggregates the TP, FP and FN for all labels and then computes the average. Macro-precision, macro-recall and macro-F1 are presented in Equations A1, A2 and A3, respectively, whereas micro-precision, micro-recall and micro-F1 are presented in Equations A4, A5 and A6,respectively.

$$\uparrow Macro\text{-}Precision = \frac{1}{q}\sum_{j=1}^{q}\frac{TP_j}{TP_j + FP_j} \tag{A1}$$

$$\uparrow Macro\text{-}Recall = \frac{1}{q}\sum_{j=1}^{q}\frac{TP_j}{TP_j + FN_j} \tag{A2}$$

$$\uparrow Macro\text{-}F1 = \frac{2 \times Macro\text{-}Precision \times Macro\text{-}Recall}{Macro\text{-}Precision + Macro\text{-}Recall} \tag{A3}$$

$$\uparrow Micro\text{-}Precision = \frac{\sum_{j=1}^{q} TP_j}{\sum_{j=1}^{q} TP_j + \sum_{j=1}^{q} FP_j} \tag{A4}$$

$$\uparrow Micro\text{-}Recall = \frac{\sum_{j=1}^{q} TP_j}{\sum_{j=1}^{q} TP_j + \sum_{j=1}^{q} FN_j} \tag{A5}$$

$$\uparrow Micro\text{-}F1 = \frac{2 \times Micro\text{-}Precision \times Micro\text{-}Recall}{Micro\text{-}Precision + Micro\text{-}Recall} \tag{A6}$$

The example-based metrics are calculated for each test instance and then averaged across the test set. Precision (Equation A7) calculates the fraction of predicted relevant labels that are relevant, whereas Recall (Equation A8) calculates the fraction predicted true relevant labels. The harmonic mean of precision and recall is calculated by $F1$ (Equation A9). Equation A10 presents the hamming-loss which evaluates the fraction of misclassified instance-label pairs, i.e. a relevant label is missed or an irrelevant is predicted (Here, Δ stands for the symmetric difference between two sets). The subset-accuracy presented in Equation A11 evaluates the fraction of correctly classified examples, i.e.

the predicted label set is identical to the ground-truth label set. Finally, the Equation A12 presents the accuracy that measures the proportion of correct predicted labels in relation to all ground-truth and predicted labels.

$$\uparrow Precision = \frac{1}{m}\sum_{i=1}^{m}\frac{|\ Y_i \cap \hat{Y}_i\ |}{\hat{Y}_i} \tag{A7}$$

$$\uparrow Recall = \frac{1}{m}\sum_{i=1}^{m}\frac{|\ Y_i \cap \hat{Y}_i\ |}{Y_i} \tag{A8}$$

$$\uparrow F1 = \frac{1}{m}\sum_{i=1}^{m}\frac{2\ |\ Y_i \cap \hat{Y}_i\ |}{|\ \hat{Y}_i\ | + |\ Y_i\ |} \tag{A9}$$

$$\downarrow hamming\text{-}loss = \frac{1}{m}\sum_{i=1}^{m}|\ \hat{Y}_i \Delta Y_i\ | \tag{A10}$$

$$\uparrow subset\text{-}accuracy = \frac{1}{m}\sum_{i=1}^{m}|\ \hat{Y}_i = Y_i\ | \tag{A11}$$

$$\uparrow accuracy = \frac{1}{m}\sum_{i=1}^{m}\frac{|\ Y_i \cap \hat{Y}_i\ |}{|\ Y_i \cup \hat{Y}_i\ |} \tag{A12}$$

The ranking-base measures evaluates the results of classifiers that predicts the labels as a ranking. The one-error (Equation A13) evaluates the fraction of examples whose top-ranked label is not in the relevant label set. The average-precision (Equation A14) evaluates the average fraction of relevant labels ranked higher than a particular label $y \in Y_i$. The coverage (Equation A15) evaluates how many steps are needed, on average, to move down the ranked label list so as to cover all the relevant labels of the example. The ranking-loss (Equation A16) evaluates the fraction of reversely ordered label pairs, i.e. an irrelevant label is ranked higher than a relevant label.

$$\downarrow one\text{-}error = \frac{1}{m}\sum_{i=1}^{m}[\![\arg\max_{y\in\mathcal{Y}} f(x_i, y) \notin \mathcal{Y}_i]\!] \tag{A13}$$

$$\uparrow average\text{-}precision = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|\ \mathcal{Y}_i\ |}\sum_{y\in Y_i}\frac{|\ \{y' \mid rank_f(x_i, y') \leqslant rank_f(x_i, y), y' \in Y_i\}\ |}{rank_f(x_i, y)} \tag{A14}$$

$$\downarrow coverage = \frac{1}{m}\sum_{i=1}^{m}\max_{y\in Y_i} rank_f(xi, y) - 1 \tag{A15}$$

$$\downarrow ranking\text{-}loss = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|\ \mathcal{Y}_i\ ||\ \hat{\mathcal{Y}}_i\ |}|\ \{(y', y") \mid f(x_i, y') \leqslant f(x_i, y"), (y', y") \in Y_i \times \hat{Y}_i)\}\ | \tag{A16}$$

The Missing Label Prediction (MLP) measure (Rivolli et al, 2018) (Equation A17) indicates the proportion of labels that are never predicted by a strategy.

$$\downarrow MLP = \frac{1}{q}\sum_{i=1}^{q} I(TP_i + FP_i == 0) \tag{A17}$$

The Wrong Label Prediction (WLP) measure (Rivolli et al, 2018) (Equation A18), which can be seen as a generalization or relaxation of MLP, represents the case where a label might be predicted for some instances, but these predictions are always wrong.

$$\downarrow WLP = \frac{1}{q}\sum_{i=1}^{q} I(TP_i == 0) \tag{A18}$$

In the presented measures, the symbol $\downarrow$ indicates that lesser values for the associated measure are better, whereas the symbol $\uparrow$ indicates that bigger values are better.

References

Blockeel H, Raedt LD, Ramon J (1998) Top-down induction of clustering trees. In: Proceedings of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML '98, p 55–63

Boutell M, Luo J, Shen X, et al (2004a) Learning multi-label scene classification. Pattern Recognition 37(9):1757–1771

Boutell MR, Luo J, Shen X, et al (2004b) Learning multi-label scene classification. Pattern Recognition 37(9):1757–1771. https://doi.org/10.1016/j.patcog.2004.03.009

Briggs F, Lakshminarayanan B, Neal L, et al (2012) Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. The Journal of the Acoustical Society of America 131(6):4640–4650

Charte F, Rivera A, del Jesus MJ, et al (2013) A first approach to deal with imbalance in multi-label datasets. In: Pan JS, Polycarpou MM, Woźniak M, et al (eds) Hybrid Artificial Intelligent Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 150–160

Charte F, Rivera AJ, del Jesus MJ, et al (2015) Quinta: A question tagging assistant to improve the answering ratio in electronic forums. In: EUROCON 2015 - International Conference on Computer as a Tool (EUROCON), IEEE, pp 1–6

Charte F, Rivera AJ, Charte D, et al (2018) Tips, guidelines and tools for managing multi-label datasets: The mldr.datasets r package and the cometa data repository. Neurocomputing https://doi.org/https://doi.org/10.1016/j.neucom.2018.02.011

Cheng W, Hüllermeier E (2009) Combining instance-based learning and logistic regression for multilabel classification. Machine Learning 76(2-3):211–225. https://doi.org/10.1007/s10994-009-5127-5

Clare A, King RD (2001) Knowledge discovery in multi-label phenotype data. In: Principles of Data Mining and Knowledge Discovery. Springer Berlin Heidelberg

Crammer K, Dredze M, Ganchev K, et al (2007) Automatic code assignment to medical text. In: Proc. Workshop on Biological, Translational, and Clinical Language Processing, Prague, Czech Republic, BioNLP07, pp 129–136

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7

Diplaris S, Tsoumakas G, Mitkas P, et al (2005) Protein classification with multiple algorithms. In: Proc. 10th Panhellenic Conference on Informatics, Volos, Greece, PCI05, pp 448–456

Duygulu P, Barnard K, de Freitas J, et al (2002) Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Computer Vision, ECCV 2002, LNCS, vol 2353. p 97–112

Elisseeff A, Weston J (2001) A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems, pp 681–687

Fan R, Lin C (2007) A study on threshold selection for multi-label classification. Department of Computer Science, National Taiwan University pp 1–23. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.66.1611{&}rep=rep1{&}type=pdf

Fürnkranz J, Hüllermeier E, Loza Mencía E, et al (2008) Multilabel classification via calibrated label ranking. Machine Learning 73(2):133–153. https://doi.org/10.1007/s10994-008-5064-8

Ganter B (1984) Two basic algorithms in concept analysis. FB4–Preprint 831, TH Darmstadt

Gibaja E, Ventura S (2014) Multi-label learning: A review of the state of the art and ongoing research. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 4(6):411–444. https://doi.org/10.1002/widm.1139

Goncalves EC, Plastino A, Freitas AA (2013) A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on, pp 469–476

Huynh-Thu VA, Irrthum A, Wehenkel L, et al (2010) Inferring regulatory networks from expression data using tree-based methods. PLOS ONE 5(9):1–10. https://doi.org/10.1371/journal.pone.0012776, URL https://doi.org/10.1371/journal.pone.0012776

Ioannou M, Sakkas G, Tsoumakas G, et al (2010) Obtaining Bipartitions from Score Vectors for Multi-Label Classification. In: 2010 22nd IEEE International Conference on Tools with Artificial Intelligence, vol 1. IEEE, pp 409–416, https://doi.org/10.1109/ICTAI.2010.65, URL http://ieeexplore.ieee.org/document/5670068/

Joachims T (1998) Text categorization with suport vector machines: Learning with many relevant features. In: Proc. 10th European Conference on Machine Learning, pp 137–142

Katakis I, Tsoumakas G, Vlahavas I (2008) Multilabel text classification for automated tag suggestion. In: Proc. ECML PKDD08 Discovery Challenge, Antwerp, Belgium, pp 75–83

Klimt B, Yang Y (2004) The enron corpus: A new dataset for email classification research. In: Proc. ECML04, Pisa, Italy. p 217–226

Krajca P, Vychodil V (2009) Distributed algorithm for computing formal concepts using map-reduce framework. In: Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII. Springer-Verlag, Berlin, Heidelberg, IDA '09, pp 333–344

Lang K (1995) Newsweeder: Learning to filter netnews. In: Proc. 12th International Conference on Machine Learning, pp 331–339

Madjarov G, Kocev D, Gjorgjevikj D, et al (2012) An extensive experimental comparison of methods for multi-label learning. Pattern Recognition 45(9):3084–3104. https://doi.org/10.1016/j.patcog.2012.03.004

Madjarov G, Dimitrovski I, Gjorgjevikj D, et al (2015) Evaluation of different data-derived label hierarchies in multi-label classification. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science) 8983:19–37. https://doi.org/10.1007/978-3-319-17876-9_2

Madjarov G, Gjorgjevikj D, Dimitrovski I, et al (2016) The use of data-derived label hierarchies in multi-label classification. Journal of Intelligent Information Systems 47(1):57–90. https://doi.org/10.1007/s10844-016-0405-8

Madjarov G, Vidulin V, Dimitrovski I, et al (2019) Web genre classification with methods for structured output prediction. Information Sciences 503:551–573. https://doi.org/10.1016/j.ins.2019.07.009, URL https://doi.org/10.1016/j.ins.2019.07.009

Nikoloski S, Kocev D, Dzeroski S (2017) Structuring the output space in multi-label classification by using feature ranking. In: Appice A, Loglisci C, Manco G, et al (eds) New Frontiers in Mining Complex Patterns - 6th International Workshop, NFMCP 2017, Held in Conjunction with ECML-PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Revised Selected Papers, Lecture Notes in Computer Science, vol 10785. Springer, pp 151–166, https://doi.org/10.1007/978-3-319-78680-3_11, URL https://doi.org/10.1007/978-3-319-78680-3_11

Nourine L, Raynaud O (1999) A fast algorithm for building lattices. Information Processing Letters https://doi.org/10.1016/s0020-0190(99)00108-8

Papanikolaou Y, Tsoumakas G, Katakis I (2018) Hierarchical partitioning of the output space in multi-label data. Data and Knowledge Engineering 116(0):42–60. https://doi.org/10.1016/j.datak.2018.05.003, https://arxiv.org/abs/arXiv:1612.06083

Pasquier N, Bastide Y, Taouil R, et al (1998) Pruning closed itemset lattices for association rules. Actes de la conférence BDA sur les Bases de Données Avancées (October):177–196. URL http://www.informatik.uni-trier.de/{~}ley/db/conf/bda/bda98.html

Pasquier N, Bastide Y, Taouil R, et al (1999) Efficient mining of association rules using closed itemset lattices. Information Systems 24(1):25–46. https://doi.org/10.1016/S0306-4379(99)00003-4

Read J (2010) Scalable multi-label classification. PhD thesis, University of Waikato

Read J, Pfahringer B, Holmes G (2008) Multi-label classification using ensembles of pruned sets. Proceedings - IEEE International Conference on Data Mining, ICDM pp 995–1000. https://doi.org/10.1109/ICDM.2008.74

Read J, Pfahringer B, Holmes G, et al (2011) Classifier chains for multi-label classification. Machine Learning 85(3):333–359. https://doi.org/10.1007/s10994-011-5256-5

Rivolli A, Parker LC, de Carvalho AC (2017) Food truck recommendation using multi-label classification. In: Portuguese Conference on Artificial Intelligence, Springer, pp 585–596, https://doi.org/10.1007/978-3-319-65340-2_48

Rivolli A, Soares C, de Carvalho AC (2018) Enhancing multilabel classification for food truck recommendation. Expert Systems 35(4):1–19. https://doi.org/10.1111/exsy.12304

Sajnani H, Saini V, Kumar K, et al (2013) The yelp dataset challenge - multilabel classification of yelp reviews into relevant categories. URL https://www.ics.uci.edu/~vpsaini/

Sanden C, Zhang JZ (2011) Enhancing multi-label music genre classification through ensemble techniques. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp 705–714, https://doi.org/10.1145/2009916.2010011

Sechidis K, Tsoumakas G, Vlahavas I (2011) On the stratification of multi-label data. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6913 LNAI

Silla CN, Freitas Aa (2011) A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery 22(1-2):31–72. https://doi.org/10.1007/s10618-010-0175-9

Tsoumakas G, Katakis I, Vlahavas I (2008) Effective and efficient multilabel classification in domains with large number of labels. Proc ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08) pp 30–44. URL http://lpis.csd.auth.gr/publications/tsoumakas-mmd08.pdf

Tsoumakas G, Katakis I, Vlahavas I (2011a) Random {\textit{k}}-labelsets for multi-label classification. IEEE Transactions on Knowledge and Data Engineering 23(7):1079–1089

Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, et al (2011b) Mulan: A java library for multi-label learning. Journal of Machine Learning Research 12:2411–2414

Turnbull D, Barrington L, Torres D, et al (2008) Semantic annotation and retrieval of music and sound effects. Audio, Speech, and Language Processing, IEEE Transactions on 16(2):467–476

Vens C, Struyf J, Schietgat L, et al (2008) Decision trees for hierarchical multi-label classification. Machine Learning 73(2):185–214. https://doi.org/10.1007/s10994-008-5077-3

Wang T, Liu L, Liu N, et al (2020) A multi-label text classification method via dynamic semantic representation model and deep neural network. Appl Intell

Wieczorkowska A, Synak P, Ra's Z (2006) Multi-label classification of emotions in music. In: Intelligent Information Processing and Web Mining, vol 35. chap 30, p 307–315

Xu J, Liu J, Yin J, et al (2016) A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously. Knowledge-Based Systems 98:172–184

Zhang ML, Zhou ZH (2006) Multilabel neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18

Zhang ML, Zhou ZH (2007) Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition 40(7):2038 – 2048. https://doi.org/https://doi.org/10.1016/j.patcog.2006.12.019

Zhang ML, Zhou ZH (2014) A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering 26(8):1819–1837. https://doi.org/10.1109/TKDE.2013.39

Zhou JP, Chen L, Guo ZH, et al (2020) Iatc-nrakel: An efficient multi-label classifier for recognizing anatomical therapeutic chemical classes of drugs. Bioinformatics 36