

# The use of data-derived label hierarchies in multi-label classification

Gjorgji Madjarov<sup>1</sup> · Dejan Gjorgjevikj<sup>1</sup> ·  
Ivica Dimitrovski<sup>1</sup> · Sašo Džeroski<sup>2</sup>

Received: 31 July 2015 / Revised: 22 March 2016 / Accepted: 29 March 2016 /  
Published online: 18 April 2016  
© Springer Science+Business Media New York 2016

**Abstract** Instead of traditional (multi-class) learning approaches that assume label independency, multi-label learning approaches must deal with the existing label dependencies and relations. Many approaches try to model these dependencies in the process of learning and integrate them in the final predictive model, without making a clear difference between the learning process and the process of modeling the label dependencies. Also, the label relations incorporated in the learned model are not directly visible and can not be (re)used in conjunction with other learning approaches. In this paper, we investigate the use of label hierarchies in multi-label classification, constructed in a data-driven manner. We first consider flat label sets and construct label hierarchies from the label sets that appear in the annotations of the training data by using a hierarchical clustering approach. The obtained hierarchies are then used in conjunction with hierarchical multi-label classification (HMC) approaches (two local model approaches for HMC, based on SVMs and PCTs, and two global model approaches, based on PCTs for HMC and ensembles thereof). The experimental results reveal that the use of the data-derived label hierarchy can significantly improve the performance of single predictive models in multi-label classification as compared to the use of a flat label set, while this is not preserved for the ensemble models.

---

✉ Gjorgji Madjarov  
gjorgji.madjarov@finki.ukim.mk  
Dejan Gjorgjevikj  
dejan.gjorgjevikj@finki.ukim.mk  
Ivica Dimitrovski  
ivica.dimitrovski@finki.ukim.mk  
Sašo Džeroski  
saso.dzeroski@ijs.si

<sup>1</sup> Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Rugjer Boshkovikj 16, 1000 Skopje, Macedonia

<sup>2</sup> Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

**Keywords** Multi-label · Hierarchical · Classification · Ranking · Learning

## 1 Introduction

Multi-label learning is concerned with learning from examples, where each example is associated with multiple labels. For instance, a document can belong to multiple categories in text categorization, a gene may be associated with multiple functions in bioinformatics, an image may belong to multiple semantic categories in image classification etc. Instead of traditional learning approaches that assume label independency and learn independent mapping functions between the input space and the corresponding labels from the output space, multi-label learning approaches should deal with the existing label dependencies and relations.

In recent years, many different approaches have been developed for solving multi-label problems (Zhang and Zhou 2014; Gibaja and Ventura 2015). Tsoumakas and Katakis (2007) summarize them into two main categories: a) algorithm adaptation methods, and b) problem transformation methods. Algorithm adaptation methods extend specific learning algorithms to handle multi-label data directly. Problem transformation methods, on the other hand, transform the multi-label classification problem into one or more single-label classification problems (where each example is associated with a single label). Madjarov et al. (2012) extend the categorization of multi-label methods with a third group of methods, namely, ensemble methods. This group of methods consists of methods that use ensembles to make multi-label predictions and their base classifiers belong to either problem transformation or algorithm adaptation methods. Methods that belong to this group are RAKEL (Tsoumakas and Vlahavas 2007), ensembles of classifier chains (ECC) (Read et al. 2009), ensembles of Predictive Clustering Trees (Kocev et al. 2007; Kocev 2011), ensembles of multi-label C4.5 trees (Clare and King 2001), Variable Pairwise Constraint projection for Multi-label ensembles (Li et al. 2013), ensembles of Fading Random Trees (Kong and Yu 2011), etc.

In addition, Madjarov et al. (2012) presented an extensive experimental evaluation of the most popular methods for multi-label learning using a wide range of evaluation measures on a variety of datasets. The results reveal that the best performing methods over all evaluation measures are the ensemble method Random Forests of Predictive Clustering Trees for Multi-target Classification (RF-PCTs for MTC) (Kocev 2011) and the single predictive model Hierarchy Of Multi-label classifierS (HOMER) (Tsoumakas et al. 2008), followed by Binary Relevance (BR) (Tsoumakas and Katakis 2007), also a single predictive model.

BR decomposes the flat multi-label output space into  $n$  single-label output spaces (where  $n$  is the total number of labels in the original multi-label learning problem). It builds one classifier for each transformed output space, using all the examples labeled with the label from that space as positive examples and all remaining examples as negative examples. On the other hand, HOMER structures the flat multi-label output space into a tree-shaped hierarchy first, and after that utilizes BR approach for solving the classification problems defined in each node of the hierarchy. We believe that the better predictive performance of HOMER as compared to BR comes as a result of the transformed output space that HOMER is using while learning and during prediction.

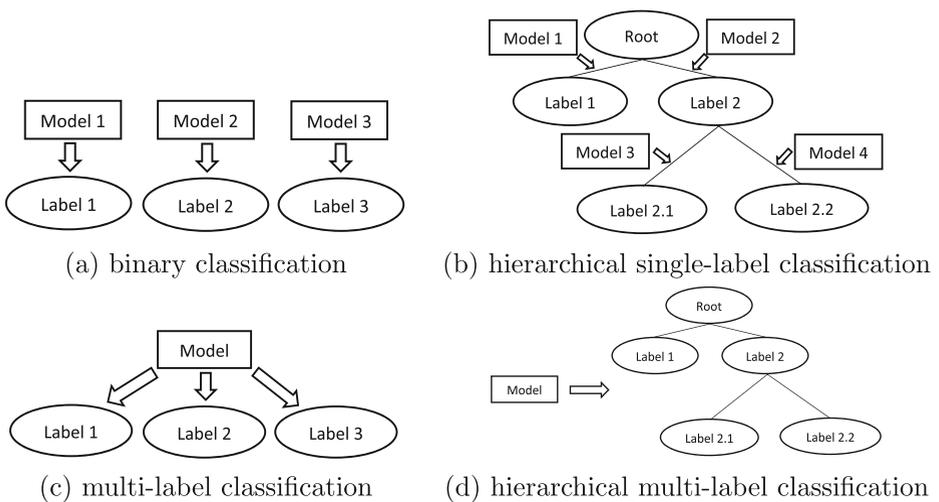
As a consequence of this believe, Madjarov et al. (2015) investigated and evaluated the utility of four different data-derived label hierarchies in the context of predictive clustering trees for HMC in a global setting (Silla Carlos and Freitas 2011). The experimental results show that the use of data-derived hierarchies results in improved predictive performance and that more balanced hierarchies offer better representation of the label relationships.

Multi-branch hierarchy (defined by balanced  $k$ -means clustering) outperforms binary hierarchies (defined by agglomerative clustering with single and complete linkage and PCTs) on datasets with higher number of labels used in the experiments.

This paper extends the aforementioned study on evaluation of different data-derived label hierarchies in multi-label classification (Madjarov et al. 2015). We explore the use of data-derived label hierarchies, constructed by using clustering approaches from the label sets that appear in the annotations of the training examples. Firstly, we investigate the utility of the data-derived hierarchies in the context of single predictive models. Four different types of single predictive models (Fig. 1) were constructed that correspond to: binary classification, hierarchical single-label classification, multi-label classification and hierarchical multi-label classification. The first two approaches construct (an architecture of) local predictive models, while the last two approaches construct global models. Secondly, we evaluate and analyze the influence that the use of data-derived label hierarchies has on ensemble approaches for HMC (in particular Random Forest ensemble method for MLC and HMC in a local and a global setting). Finally, we investigate whether the conclusions from the investigation on single models carry over to the ensemble models and whether it is more beneficial to extract and use data-derived label hierarchies or just use the original, flat organization of the labels in multi-label classification problems.

In order to answer to all of these questions we have compared:

1. The best performing problem transformation methods for MLC: the BR method (Tsoumakas and Katakis 2007) and the HOMER method (Tsoumakas et al. 2008) utilizing SVMs as a base classifiers,



**Fig. 1** Schematic representation of the four different modeling tasks we consider to investigate how exploitation of label hierarchy affects the performance. Single label classification **a**, builds a separate model for each label, while hierarchical single label classification **b**, builds a separate model for each edge of the artificially generated label hierarchy (each model is trained by using only data that is relevant to that edge). Both models build *local* classifier. Multi-label classification **c** and hierarchical multi-label classification **d** build one *global* model which considers all the classes at once: the former approach **(c)** directly solves the flat multi-label classification task, while the latter approach **(d)** exploits information about the artificially generated label hierarchy

2. Three different approaches based on PCTs (Blockeel et al. 1998), one for solving classical MLC problems (Kocev 2011) and two for solving HMC problems by using a local and a global setting (Vens et al. 2008)), and
3. Ensembles of PCTs (Kocev 2011) for MLC (one of the best performing methods for MLC (Madjarov et al. 2012)) and ensembles of PCTs for HMC (in a local and a global setting).

The experimental evaluation is made on 11 benchmark multi-label datasets using 16 evaluation measures. The datasets come from five application domains: two from image classification, one from gene function prediction, six from text classification, one from music classification and one from video classification. The predictive performance of the methods is assessed using six example-based measures, six label-based measures and four ranking-based measures.

The remainder of this paper is organized as follows. Section 2 defines the tasks of multi-label classification, multi-label ranking and hierarchical multi-label classification. An overview of the base level algorithms for MLC and the approaches for HMC that are experimentally compared in this work are given in Section 3 and Section 4. The general use of data derived label hierarchies in multi-label classification is proposed in Section 5. Section 6 describes the multi-label datasets, the evaluation measures and the experimental setup, while Section 7 presents and discusses the experimental results. Finally, the conclusions and directions for further work are presented in Section 8.

## 2 The tasks of multi-label and hierarchical multi-label classification

In this section, we first define the task of multi-label learning and then the task of hierarchical multi-label learning.

### 2.1 The task of multi-label classification (MLC)

Multi-label learning is concerned with learning from examples, where each example is associated with multiple labels. These multiple labels belong to a predefined set of labels. We can distinguish two types of tasks: multi-label classification and multi-label ranking.

In the case of multi-label classification, the goal is to construct a predictive model that will provide a list of relevant labels for a given, previously unseen example. On the other hand, the task of multi-label ranking is understood as learning a model that associates with a query example both a ranking of the complete label set and a bipartite partition of this set into relevant and irrelevant labels (Brinker et al. 2006).

The task of multi-label learning is defined as follows (Kocev et al. 2013):

#### Given:

- An input space  $\mathcal{X}$  that consists of vectors of values of primitive data types (nominal or numeric), i.e.,  $\forall \mathbf{x}_i \in \mathcal{X}, \mathbf{x}_i = (\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \dots, \mathbf{x}_{i_D}), 1 \leq i \leq N$ , where  $N$  is the number of vectors in the input space and  $D$  is the size of the vector (or number of descriptive attributes),
- an output space  $\mathcal{Y}$  that is defined as  $\mathcal{Y} = 2^{\mathcal{L}}$ , i.e. the set of subsets of finite set of different labels  $\mathcal{L} = \{\lambda_1, \lambda_2, \dots, \lambda_Q\}$  ( $Q > 1$  and  $\mathcal{Y}_i \subseteq \mathcal{L}$ )

- a set of examples  $E$ , where each example is a pair of a vector and a label set from the input and output space respectively, i.e.,  $E = \{(\mathbf{x}_i, \mathcal{Y}_i) | \mathbf{x}_i \in \mathcal{X}, \mathcal{Y}_i \subseteq \mathcal{L}, 1 \leq i \leq N\}$  where  $N$  is the number of examples of  $E$  ( $N = |E| = |\mathcal{X}|$ ), and
- a quality criterion  $q$ , which rewards models with high predictive performance and low computational complexity.

If the task at hand is multi-label classification, then the goal is to

**Find:** a function  $h: \mathcal{X} \rightarrow 2^{\mathcal{L}}$  from the input space to the label power-set (which assigns a set of labels to each example) such that  $h$  maximizes  $q$ .

On the other hand, if the task is multi-label ranking, then the goal is to

**Find:** a function  $f: \mathcal{X} \times \mathcal{L} \rightarrow \mathcal{R}$ , such that  $f$  maximizes  $q$ , where  $\mathcal{R}$  gives the ranking for a given label and for a given example.  $\mathcal{R}$  is a set of values on which a total strict order exists, typically a set of real non-negative values which can be totally ordered.

An extensive bibliography of learning methods for solving multi-label learning problems can be found in Madjarov et al. (2012), Zhang and Zhou (2014), and Gibaja and Ventura (2015).

## 2.2 The task of hierarchical multi-label classification (HMC)

Hierarchical classification differs from the multi-label classification in the following: the labels are organized in a hierarchy. An example that is labeled with a given label is automatically labeled with all its ancestor-labels (this is known as the hierarchy constraint (Vens et al. 2008)). Furthermore, an example can be labeled simultaneously with multiple labels that can follow multiple paths from the root label. This task is called hierarchical multi-label classification (HMC).

Here, the output space  $\mathcal{Y}$  is defined with a label hierarchy  $(\mathcal{L}, \leq_h)$ , where  $\mathcal{L}$  is a set of labels and  $\leq_h$  is a partial order representing the ancestor-descendant relationship ( $\forall \lambda_1, \lambda_2 \in \mathcal{L} : \lambda_1 \leq_h \lambda_2$  if and only if  $\lambda_1$  is an ancestor of  $\lambda_2$ ) structured as a tree (Kocev et al. 2013). Each example from the set of examples  $E$  is a pair of a vector and a set from the input and output space respectively, where the set satisfies the hierarchy constraint, i.e.,  $E = \{(\mathbf{x}_i, \mathcal{Y}_i) | \mathbf{x}_i \in \mathcal{X}, \mathcal{Y}_i \subseteq \mathcal{L}, \lambda \in \mathcal{Y}_i \Rightarrow \forall \lambda' \leq_h \lambda : \lambda' \in \mathcal{Y}_i, 1 \leq i \leq N\}$  where  $N$  is the number of examples of  $E$  ( $N = |E|$ ). The quality criterion  $q$ , rewards models with high predictive performance and low complexity as in the task of multi-label classification.

An extensive bibliography of learning methods for hierarchical classification scattered across different application domains is given by Silla Carlos and Freitas (2011).

## 3 Algorithms for multi-label classification

In this section we give a detailed description of the algorithms for MLC that are used in the experimental comparison in this work.

### 3.1 Binary relevance

The simplest strategy in the multi-label setting is the one-against-all strategy, also referred to as the Binary Relevance (BR) method (Tsoumakas and Katakis 2007). It addresses the

multi-label learning problem by learning one classifier for each label, using all the examples labelled with that label as positive examples and all remaining examples as negative. When making a prediction, each binary classifier predicts whether its label is relevant for the given example or not, resulting in a set of relevant labels. In the ranking scenario, the labels are ordered according to the probability associated to each label by the respective binary classifier.

The most important and widely relevant advantage of BR is its low computational complexity relative to other methods. It is theoretically simple and intuitive. Its assumption of label independence makes it suited to contexts where new examples may not necessarily be relevant to any known labels or where label relationships may change over the test data. Given a constant number of examples, BR scales linearly with the size of the known label set.

### 3.2 Predictive clustering trees

Predictive Clustering Trees (PCTs) (Blockeel et al. 1998)<sup>1</sup> are a generalization of the decision tree approach (Breiman et al. 1984). They can be used for a variety of learning tasks including different types of prediction and clustering. This includes both multi-target/multi-label classification and hierarchical multi-label classification.

The PCT framework views a decision tree as a hierarchy of clusters. The top-node of a PCT corresponds to one cluster containing all data, which is recursively partitioned into smaller clusters while moving down the tree. The leaves represent the clusters at the lowest level of the hierarchy and each leaf is labeled with its cluster's prototype (prediction).

PCTs are built using greedy recursive top-down induction (TDI) algorithm, similar to that of C4.5 (Quinlan 1993) or CART (Breiman et al. 1984). The learning algorithm starts by selecting an attribute (feature) test for the root node. Based on this test, the training set is partitioned into subsets according to the test outcome. This is recursively repeated to construct the subtrees. The partitioning process stops when a stopping criterion is satisfied (e.g., the number of instances in the induced subsets is smaller than some predefined value; the length of the path from the root to the current subset exceeds some predefined value, etc.). In that case, the prediction is calculated and stored in a leaf.

One of the most important steps in the TDI algorithm is the test selection procedure. For each node, an attribute (test) is selected from the input space by using a heuristic function computed on the training examples. The goal of the heuristic is to guide the algorithm towards small trees with good predictive performance. The heuristic used in this algorithm for selecting the attribute tests in the internal nodes is the reduction in variance caused by partitioning the instances, where the variance  $Var(E)$  is defined by (1) for multi-target classification and (3) for hierarchical multi-label classification. Maximizing the variance reduction maximizes cluster homogeneity and improves predictive performance.

#### 3.2.1 PCTs for multi-target classification

Multi-target prediction (Kocev 2011) is concerned with learning from examples, where each example is associated with multiple targets. If the targets are continuous variables then the task is referred to as multi-target regression. If the targets are discrete variables, then we

---

<sup>1</sup>The PCT framework is implemented in the CLUS system, which is available at <http://www.cs.kuleuven.be/~dtai/clus>.

have a task of multi-target classification. In the multi-label scenario, each discrete target variable is a binary variable (it holds 1 if the corresponding label is relevant to the instance and 0 otherwise).

For the task of predicting of discrete targets, the variance function  $Var(E)$  is computed as the sum of the *Gini indices* (Breiman et al. 1984) of the variables from the target vector, i.e.,

$$Var(E) = \sum_{i=1}^T Gini(E, \Lambda_i) \quad (1)$$

where  $T$  is the number of target attributes and  $\Lambda_i$  is the  $i$ -th target attribute.  $Gini(E, \Lambda_i) = 1 - \sum_{j=1}^M p_j^2$  where  $p_j$  is the fraction of records in  $E$  of the  $j$ -th class in the  $\Lambda_i$  target attribute and  $M$  is the number of classes in the same target attribute. The prototype function returns a vector of probability distributions over the values for each discrete target. In the case of multi-label learning, it returns a vector of probabilities that an example is labeled with each of the labels from the original label set  $\mathcal{L}$ . An example arriving in the leaf is labeled with label  $\lambda_i$  if its corresponding probability from the vector of probabilities is above some threshold  $\tau$  (e.g. chosen by a domain expert).

### 3.2.2 Random forests of PCTs for MTC

In this subsection, we explain how PCTs for MTC are used in the context of an ensemble classifier. An ensemble is a set of (base) classifiers. A new example is classified by the ensemble by combining the predictions of the member classifiers. The predictions can be combined by taking the average (for regression tasks), the majority vote (for classification tasks) (Breiman 2001), or more complex combinations.

Averaging is applied to combine the predictions of the different trees. The leaf's prototype is the proportion of examples of different classes that belong to it. As for the single PCT, a threshold should be specified to make a prediction.

We consider the random forest ensemble learning technique that constructs different classifiers by making different bootstrap samples (Breiman 1996) of the training set on one hand, and by randomly changing the feature set during learning on the other hand. More precisely, at each node in the decision tree, a random subset of the input attributes is taken, and the best feature is selected from this subset (instead of the set of all attributes). The number of attributes that are retained is given by a function  $f$  of the total number of input attributes  $x$  (e.g.,  $f(x) = x$ ,  $f(x) = \sqrt{x}$ ,  $f(x) = \lfloor \log_2 x \rfloor + 1, \dots$ ).

## 4 The use of hierarchies in hierarchical multi-label classification

An extensive bibliography of learning methods for hierarchical classification scattered across different application domains is given by Silla Carlos and Freitas (2011). Based on the existing literature, they propose a unifying framework for hierarchical classification, including a taxonomy of hierarchical classification problems and methods, and clarify the similarities and differences between a number of types of problems and methods. They also present a conceptual comparison of these types of problems and methods at a high level of abstraction, discussing their advantages and disadvantages.

One of the dimensions along which the hierarchical classification methods differ is the way of using (exploring) the hierarchical label structure in the learning and prediction phases. Silla Carlos and Freitas (2011) reviewed three different approaches: the top-down

(or local) approach that uses local information to create a set of local classifiers; the global (or big-bang) approach, where a single classifier coping with the entire class hierarchy is learned; or the flat approach, that ignores the class relationships, typically building models that predict only the leaf nodes.

#### 4.1 Local approaches for solving HMC problems

In the task of hierarchical multi-label classification, there are three different types of local approaches that can be used (Silla Carlos and Freitas 2011): Local classifiers per level, local classifiers per node, and local classifiers per parent node. The local classifiers per level approach constructs one classifier for each level of the hierarchy. The local classifiers per node approach constructs a classifier for each node from the hierarchy, except the root. There are several policies for selecting the positive and negative examples that can be used to train the local classifiers. The local classifiers per parent node approach constructs a classifier for each non-leaf node from the hierarchy. A multi-label classifier for each parent node is learned, e.g. by transforming the MLC problem by using the binary relevance scheme and learning binary classifiers for each child node in the hierarchy.

##### 4.1.1 Architectures of PCTs and random forests of PCTs for hierarchical classification

Vens et al. (2008) investigated the performance of the local classifiers per node and per parent node approaches over a large collection of (native) hierarchical multi-label datasets from functional genomics. The conclusion of the study was that the last approach (called hierarchical single-label classification - HSC) performs better in terms of predictive performance, smaller total model size and faster induction times.

We will keep the notation HSC here, but would like to emphasize that this approach performs hierarchical multi-label classification and not hierarchical single-label classification as the term suggests.

The approach of Predictive Clustering Trees for Hierarchical Single-label Classification (PCTs for HSC) by Vens et al. (2008), constructs a decision tree classifier for each edge (connecting a label  $\lambda$  with its direct parent label  $parent(\lambda)$ )<sup>2</sup> in the hierarchy, thus creating an architecture of classifiers. The corresponding tree predicts membership to label  $\lambda$ , using the instances that belong to  $parent(\lambda)$ . The construction of this type of tree uses few instances: only instances labeled with  $parent(\lambda)$  are used for training. The instances labeled with label  $\lambda$  are positive instances, while the ones that are labeled with  $parent(\lambda)$ , but not with  $\lambda$  are negative. The resulting tree predicts the conditional probability  $P(\lambda|parent(\lambda))$ , where for the top-level labels, all training examples are used.

To make predictions for a new instance, PCTs for HSC use the product rule  $P(\lambda) = P(\lambda|parent(\lambda)) \cdot P(parent(\lambda))$  (for non top-level labels). This rule applies the trees recursively for each node (label) of the hierarchy, starting from the tree for a top-level label.

Kocev (2011) extends the approach of Vens et al. (2008) of using PCTs for HSC by applying ensembles as local classifiers at each branch, instead of single decision trees (PCTs). The method of Random Forests of PCTs for HSC (RF-PCTs for HSC) constructs a random

<sup>2</sup>We use the term  $parent(\lambda)$  for the direct parent label (the label at the previous level that is directly connected to  $\lambda$ ) and the term  $ancestor$  for all parent labels from the root of the hierarchy to the  $parent(\lambda)$  (including  $parent(\lambda)$ ).

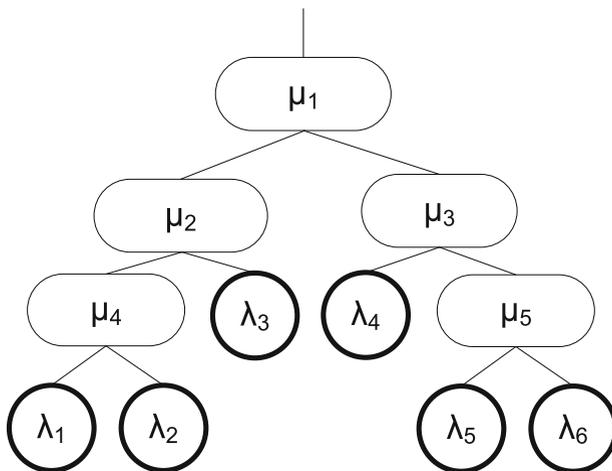
forest ensembles of PCTs for each edge (connecting a label  $\lambda$  with its direct parent label  $parent(\lambda)$ ) in the hierarchy, thus creating an architecture of ensembles. All other settings for this method are the same as for the PCTs for HSC. We will refer to the two approaches as HSC architectures of PCTs and ensembles of PCTs.

#### 4.1.2 The hierarchy architecture of SVM classifiers in HOMER

The approach taken by HOMER is to use local classifiers for solving HMC tasks. After learning a hierarchy on the flat label space by applying an unsupervised (clustering) approach to the label part of the data, HOMER constructs a hierarchical architecture of SVM classifiers that follows the hierarchy on the label space. In each internal node of the hierarchy, a BR architecture of binary SVMs is created. Since the BR architecture contains one SVM for each descendant label of the internal node, this architecture is the same as the HSC architecture, the only difference being that PCTs for binary classification are used in PCTS for HSC and SVMs for binary classification are used in HOMER.

In the hierarchy (tree) on the label space used by HOMER, the leaves represent the labels of the original MLC problem, while each internal node  $m$  contains the union of the label sets of its children (i.e., the meta-label  $\mu_m$ ). In the learning phase, a training example is considered annotated with a meta-label  $\mu_m$ , if it is annotated with at least one of the (original) labels from  $\mu_m$ . An example hierarchy of labels and classifiers produced by HOMER for a multi-label classification task with 8 labels is given in Fig. 2.

In the prediction phase, HOMER starts from the root and follows a recursive process forwarding the test instance  $\mathbf{x}$  to the child node  $m$  only if  $\mu_m$  is among the predictions of the multi-label classifier. The union of the predicted labels in the leaves of the tree are the relevant labels for an instance  $\mathbf{x}$ . For example, let's say that according to the predictions obtained by the multi-label classifiers in the nodes of the tree from Fig. 2 the labels  $\mu_1, \mu_2, \lambda_3, \mu_5$  and  $\lambda_5$  were assigned to an example  $\mathbf{x}$ . Only the leaf labels  $\lambda_3$  and  $\lambda_5$  are declared as relevant labels for that example  $\mathbf{x}$ .



**Fig. 2** An example of label hierarchy defined over the flat label space of the *emotions* dataset by using balanced  $k$ -means clustering method where  $k$  is set to 2 ( $\lambda_i$  - original label,  $\mu_i$  - artificially defined meta-label)

While the prediction phase of HOMER is similar to that of PCTs for HSC, there is an important difference. HOMER only forwards a testing instance to the child nodes that correspond to the meta-labels predicted to be relevant for the instance. As a consequence, no probability (and ranking) is produced for the original labels that belong to meta-labels predicted to be irrelevant.

## 4.2 Global approaches for solving HMC problems

Although the problem of hierarchical multi-label classification can be tackled by the local approaches, learning a single global model for all labels (in the hierarchy) can have some advantages (Kocev 2011). The total size of the global classification model is typically smaller as compared to the total size of all the local models learned by any of the local classifier approaches. Also, in the global classifier approach, a single classification model is built from the training set, taking into account the label hierarchy and relationships. During the test phase, each test example is classified using the induced model, in a process that can assign labels to a test example at potentially every level of the hierarchy.

### 4.2.1 PCTs and random forests of PCTs for hierarchical multi-label classification

To apply PCTs to the task of HMC, the example labels are represented as vectors with Boolean components. The components in the vector correspond to the labels in the hierarchy traversed in a depth-first manner. The  $k$ -th component of the vector is 1 ( $v_k = 1$ ) if the example is labeled with label  $\lambda_k$  and 0 otherwise. If  $v_k = 1$ , then  $v_j = 1$  for all  $v_j$ 's on the path from the root to  $v_k$ .

The variance of a set of examples  $E$  is defined as the average squared distance between each example's label vector  $\mathbf{v}_i$  and the mean label vector  $\bar{\mathbf{v}}$  of the set, i.e.,

$$\bar{\mathbf{v}} = \frac{1}{|E|} \sum_{i=1}^{|E|} \mathbf{v}_i \quad (2)$$

$$Var(E) = \frac{1}{|E|} \sum_{\mathbf{e}_i \in E} d(\mathbf{v}_i, \bar{\mathbf{v}})^2 \quad (3)$$

where each component of  $\bar{\mathbf{v}}$  is the proportion of examples  $\bar{v}_k$  in the leaf that are labeled with label  $\lambda_k$ .

The higher levels of the hierarchy are more important: an error at the upper levels costs more than an error at the lower levels. Considering this, a weighted Euclidean distance is used (4):

$$d(\mathbf{v}_1, \mathbf{v}_2) = \sqrt{\sum_k^{|v|} w(\lambda_k) (\mathbf{v}_{1k} - \mathbf{v}_{2k})^2} \quad (4)$$

where  $\mathbf{v}_{ik}$  is the  $k$ 'th component of the class vector  $\mathbf{v}_i$  of the instance  $\mathbf{e}_i$ , and  $w(\lambda_k)$  are the class weights. The class weights decrease with the depth of the class in the hierarchy,  $w(\lambda_k) = w_0 \cdot w(\lambda_j)$ , where  $\lambda_j$  is the parent of  $\lambda_k$ .

Each leaf in the tree stores the mean  $\bar{\mathbf{v}}$  of the vectors of the examples that are sorted into that leaf. An example arriving in the leaf can be labeled with label  $\lambda_k$  if  $\bar{v}_k$  is above some threshold  $t_k$  (that can be chosen by a domain expert).

Random forests of PCTs for HMC are considered in the same manner as the random forest of PCTs for MTC. In the case of HMC the ensemble is a set of PCTs for HMC. A new example is classified by the ensemble by combining the predictions of the member classifiers by taking the majority vote (Breiman 2001). Like in PCTs for HMC, the predictions of the random forest ensemble of PCTs for HMC satisfy the hierarchy constraint (an example that is labeled with a given label is automatically labeled with all its ancestor-labels).

### 5 The use of data derived label hierarchies in multi-label classification

In this study we investigate the use of label hierarchies, constructed in a data-driven manner in conjunction with HMC approaches (local and global) (Silla Carlos and Freitas 2011) and ensemble methods capable of solving HMC classification problems.

In particular, we derive label hierarchies considering the label sets that appear in the annotations of the training examples from the original (flat) classification problem. In this way, we first structure the label co-occurrence relationships that exist hidden in the output space of the multi-label classification problems in a hierarchy and after that, use that (data-derived) hierarchy to map the original classification problem into a hierarchical multi-label one. This artificially generated hierarchical multi-label classification problem could be considered as a separate (newly defined) hierarchical multi-label classification task and it could be tackled by global (*big-bang*) (Madjarov et al. 2015) or local approaches (Silla Carlos and Freitas 2011) for HMC.

An example hierarchy of labels generated by using the balanced *k*-means clustering method from the *emotions* multi-label classification task (used in the experimental evaluation) is given in Fig. 2. The original label space of the *emotions* dataset has six labels  $\{\lambda_1, \lambda_2, \dots, \lambda_6\}$  and each example from the dataset originally is labeled with one or more labels. Table 1 shows five examples from the *emotions* dataset with their original labels (third column - *original labels*) and the corresponding hierarchical labels (fourth column - *hierarchical labels*) obtained by using the label hierarchy from Fig. 2 ( $\mathcal{HL} = \{\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6\}$ ). Each example in the transformed, HMC dataset is actually labeled with multiple paths of the hierarchy, defined from the root to the leaves (represented by the relevant labels for the corresponding example in the original MLC dataset).

**Table 1** Five examples from the *emotions* dataset with their *original labels* and the corresponding *hierarchical labels* obtained by using the label hierarchy from Fig. 2

Example	Features (F1, F2,..., FD)	Original labels sets	Hierarchical labels set
<b>x<sub>1</sub></b>	$x_{11}, x_{12}, \dots, x_{1D}$	$\{\lambda_1\}$	$\{\mu_1, \mu_2, \mu_4, \lambda_1\}$
<b>x<sub>2</sub></b>	$x_{21}, x_{22}, \dots, x_{2D}$	$\{\lambda_3, \lambda_5\}$	$\{\mu_1, \mu_2, \mu_3, \mu_5, \lambda_3, \lambda_5\}$
<b>x<sub>3</sub></b>	$x_{31}, x_{32}, \dots, x_{3D}$	$\{\lambda_6\}$	$\{\mu_1, \mu_3, \mu_5, \lambda_6\}$
<b>x<sub>4</sub></b>	$x_{41}, x_{42}, \dots, x_{4D}$	$\{\lambda_1, \lambda_6\}$	$\{\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \lambda_1, \lambda_6\}$
<b>x<sub>5</sub></b>	$x_{51}, x_{52}, \dots, x_{5D}$	$\{\lambda_1, \lambda_2, \lambda_6\}$	$\{\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \lambda_1, \lambda_2, \lambda_6\}$

## 5.1 Generating a label hierarchy on a multi-label output space

There exist many possible ways to structure the output space of a flat MLC problem. This process is critical for the good performance of the HMC methods on the transformed problems. One can use classical hierarchical clustering algorithms, such as hierarchical agglomerative and divisive clustering, some partitioning algorithms employed at each node of the hierarchy, or graph-based clustering approaches (Madjarov et al. 2015).

When we build the hierarchy over the label space, there is only one constraint that we should take care of: the original MLC task should be defined by the leaves of the label hierarchy. In particular, the labels from the original MLC problem represent the leaves of the tree hierarchy (Fig. 2), while the labels that represent the internal nodes of the tree hierarchy are so-called meta-labels (that model the correlation among the original labels).

In this work, we employ the approach of balanced  $k$ -means proposed by Tsoumakas et al. (2008) for deriving the hierarchy on the output space of the (original) MLC problem. It showed best performance in comparison between four different approaches that was recently made by Madjarov et al. (2015). In that comparison, only global approaches for HMC were considered.

The balanced  $k$ -means clustering approach extends the well-known  $k$ -means algorithm with an explicit constraint on the size of each cluster. The algorithm consider only the label data  $\mathbf{W}_n$  of the examples at the current node  $n$  of the hierarchy.  $\mathbf{W}_n = [w_{ij}]$ , where the value of  $w_{ij}$  ( $i = 1 \dots |E_n|$  and  $j = 1 \dots |\mathcal{L}_n|$ ) is 1 if the  $i$ -th example of  $E_n$  is labeled with the label  $\lambda_j \in \mathcal{L}_n$  and 0 otherwise.  $E_n$  is the set of examples that belong to the node  $n$  and  $\mathcal{L}_n \subseteq \mathcal{L}$  is the set of labels considered in the node  $n$  (for the top level node - root node  $\mathcal{L}_n = \mathcal{L}$ ). For example, for the node of the hierarchy on Fig. 2, corresponding to the meta-label  $\mu_3$  only the label data from  $\lambda_4$ ,  $\lambda_5$  and  $\lambda_6$  is considered.

For the node  $n$ , the algorithm accepts as input the set of labels  $\mathcal{L}_n$ , the label data  $\mathbf{W}_n$  of the examples that belong to that node, the number of partitions (clusters)  $k$  and the number of iterations  $it$ . It outputs  $k$  disjoint subsets of  $\mathcal{L}_n$  with approximately equal sizes. Figure 3 shows the balanced  $k$ -means algorithm in pseudo-code.

The distance between the label  $\lambda_j$  ( $\lambda_j \in \mathcal{L}_n$ ,  $j = 1 \dots |\mathcal{L}_n|$ ) and the  $r$ -th cluster center  $c_r$  (out of  $k$ ) within the labels data  $\mathbf{W}_n$  in the  $n$ -th node is calculated by (5).

$$distance(\lambda_j, c_r, \mathbf{W}_n) = \sqrt{\sum_{i=1}^{|E_n|} (c_{ri} - w_{ij})^2} \quad (5)$$

## 5.2 Solving MLC problems by using global approaches for HMC

Figure 4 gives the pseudo-code of the general scenario for solving a MLC problem by using data-derived label hierarchies and global approaches for HMC. The scenario first defines the hierarchy, then solves the HMC problem by using a global approach for HMC. It finally transforms the HMC predictions  $P_H$  of the global model into a flat MLC representation.

$E^{train}$  and  $E^{test}$  are the training and testing examples, while  $\mathbf{W}^{train}$  is only the label part (label data) of the training set. Using the label hierarchy derived from the label data,  $\mathbf{W}^{train}$  is transformed into new hierarchically organized label data  $\mathbf{W}_H^{train}$ .  $E_H^{train}$  and  $E_H^{test}$  are the corresponding hierarchical multi-label datasets obtained by transforming the original (flat) multi-label datasets ( $E^{train}$  and  $E^{test}$ ) into hierarchical form.

$P_H$  are the predictions for the examples of the hierarchical multi-label dataset  $E_H^{test}$ , while  $P$  are the predictions for the original labels.  $P_H$  are represented as vectors of probabilities (one vector for one example), where each probability is associated to only one label

---

```

procedure BalancedkMeans( $k, \mathcal{L}_n, W_n, it$ )
1: for  $i = 1$  to  $k$  do
2:    $C_i = \{\}$ ;
3:   //Random initialization of the  $k$  clusters
4:    $c_i =$  random member of  $\mathcal{L}_n$ ;
5: end for
6: while  $it > 0$  do
7:   for all  $\lambda \in \mathcal{L}_n$  do
8:     for  $i = 1$  to  $k$  do
9:        $d_{\lambda i} =$  distance( $\lambda, c_i, W_n$ );
10:    end for
11:     $finished = false$ 
12:     $\nu = \lambda$ 
13:    while not finished do
14:       $j = \arg \max_i d_{\nu i}$ ;
15:      Insert sort( $\nu, d_\nu$ ) to sorted list  $C_j$ ;
16:      //Balance the number of labels in each cluster
17:      if  $|C_j| > \lceil |\mathcal{L}_n|/k \rceil$  then
18:        remove last element from  $C_j$ ;
19:         $d_{\nu j} = \infty$ ;
20:      else
21:         $finished = true$ 
22:      end if
23:    end while
24:  end for
25:  recalculate centers;
26:   $it = it - 1$ ;
27: end while
28: return  $C_1, \dots, C_k$ ;

```

---

**Fig. 3** Balanced  $k$ -Means

from the hierarchy (meta-label representing an internal node or original label representing a leaf). In the context of global approaches for HMC, predictions  $P$  in the original multi-label scenario can be obtained by using different approaches for transforming the hierarchical

---

```

procedure MLCToHMC( $E^{train}, E^{test}$ ) returns performance
1:  $\mathbf{W}^{train} =$  ExtractLabelSet( $E^{train}$ );
2:  $\mathbf{W}_H^{train} =$  DefineHierarchy( $\mathbf{W}^{train}$ );
3:
4: //transform multi-label dataset to hierarchical multi-label one
5:  $E_H^{train} =$  MLCToHMCTrainDataset( $E^{train}, \mathbf{W}_H^{train}$ );
6:  $E_H^{test} =$  MLCToHMCTestDataset( $E^{test}, \mathbf{W}_H^{train}$ );
7:
8: //solve transformed hierarchical multi-label problem
9: //by using local/global approach for HMC
10: HMCModel = GlobalHMCMethod( $E_H^{train}$ );
11: //HMCModel = LocalHMCMethod( $E_H^{train}$ );
12:
13: //generate HMC predictions
14:  $P_H =$  HMCModel( $E_H^{test}$ );
15:
16: //Extract predictions only for the leaves from the HMC predictions  $P_H$ 
17:  $P =$  ExtractLeavesPredictionsFromHMCPredictions( $P_H, \mathbf{W}_H^{train}, \mathbf{W}^{train}$ );
18: return EvaluatePredictions( $P$ );

```

---

**Fig. 4** The general scenario for solving flat MLC problems by using local/global approaches for HMC: Line 10 is used for the local approaches, and line 11 for the global ones

multi-label predictions  $P_H$ . In this work, we propose the simplest approach: only the probabilities for the leaves from the hierarchical predictions  $P_H$  are evaluated, while the other probabilities (for the meta-labels) are simply ignored.

### 5.3 Solving MLC problems by using local approaches for HMC

After the transformation of the original MLC problem into a HMC one, the new HMC problem can be also solved by a hierarchical multi-label local learning approach. There are different ways of using local information to create local classifiers, and although most of them are referred to as top-down in the literature, they are very different during the training phase and slightly different in the test phase.

Only line 10 of the pseudo-code in Table 4 needs to be changed (with line 11) to obtain an algorithm for solving MLC problems by using data-derived label hierarchies with local approaches for HMC. The algorithm still defines the label hierarchy first and then solves the HMC problem by using a local approach for HMC. It also extracts the predictions for the leaves of the hierarchy (that are actually the predictions for the original labels) and evaluates the performance (instead of the global one).

## 6 Experimental design

In this section, we present the experimental design used to compare the methods for flat MLC and for MLC via HMC. In particular, we compare:

- The BR method and the HOMER method (BR as a flat MLC approach and HOMER as MLC via local HMC approach, both of them using SVM architectures as base classifiers);
- PCTs for multi-target classification (as flat MLC approach), HSC architectures of PCTs for binary classification (as a MLC via local HMC approach) and PCTs for HMC (as a MLC via global HSC approach).

We also compare ensembles of PCTs in the same settings as the single trees:

- ensembles of PCTs for multi-target classification with HSC architectures of ensembles of PCTs for binary classification and
- ensembles of PCTs for HMC (the latter two based on the artificially defined label hierarchy).

We first briefly describe the benchmark multi-label datasets. We then give a short overview of the evaluation measures typically applied to assess the predictive performance of methods for multi-label learning. Next, we present the specific setup and the instantiation of the parameters for the compared multi-label learning methods. Finally, we present the procedure for statistical evaluation of the experimental results.

### 6.1 Datasets

We use eleven multi-label classification benchmark problems used in previous studies and evaluations of methods for multi-label learning. We include benchmark datasets of different size and from different application domains. Table 2 presents the basic statistics of the

**Table 2** Description of the benchmark problems in terms of application domain (*domain*), number of training (*#tr.e.*) and test (*#t.e.*) examples, number of features (*D*), total number of labels (*Q*) and label cardinality (*l<sub>c</sub>*)

	<i>domain</i>	<i>#tr.e.</i>	<i>#t.e.</i>	<i>D</i>	<i>Q</i>	<i>l<sub>c</sub></i>
emotions	multimedia	391	202	72	6	1.87
scene	multimedia	1211	1159	294	6	1.07
yeast	biology	1500	917	103	14	4.24
medical	text	645	333	1449	45	1.25
enron	text	1123	579	1001	53	3.38
corel5k	multimedia	4500	500	499	374	3.52
tmc2007	text	21519	7077	500	22	2.16
mediamill	multimedia	30993	12914	120	101	4.38
bibtex	text	4880	2515	1836	159	2.40
delicious	text	12920	3185	500	983	19.02
bookmarks	text	60000	27856	2150	208	2.03

The problems are ordered by their overall complexity roughly calculated as  $\#tr.e. \times D \times Q$

datasets. We can note that the datasets vary in size from 391 to 60000 training examples, from 202 to 27856 testing examples, from 72 to 2150 features, from 6 to 983 labels, and from 1.07 to 19.02 labels per example on average (i.e., label cardinality (Tsoumakas and Katakis 2007)). The datasets come pre-divided into training and testing parts as used by other researchers. In our experiments, we use these partitions in their original format. The training part usually comprises around 2/3 of the complete dataset, while the testing part consists of the remaining 1/3 of the dataset.

The datasets come from three domains: biology, multimedia and text categorization. From the biological domain, we have the *yeast* dataset (Elisseeff and Weston 2005). It is a widely used dataset, where genes are instances in the dataset and each gene can be associated with 14 biological functions (labels).

The datasets that belong to the multimedia domain are: *emotions* (Trohidis et al. 2008), *scene* (Boutell et al. 2004), *corel5k* (Duygulu et al. 2002) and *mediamill* (Snoek et al. 2006). The domain of text categorization is represented with 6 datasets: *medical* (Read et al. 2009), *enron* (Klimt and Yang 2004), *tmc2007* (Srivastava and Zane-Ulman 2005), *bibtex* (Katakis et al. 2008), *delicious* (Tsoumakas et al. 2008) and *bookmarks* (Katakis et al. 2008).

## 6.2 Evaluation measures

Performance evaluation for multi-label learning systems differs from that of classical single-label learning systems. In any multi-label experiment, it is essential to include multiple and contrasting measures because of the additional degrees of freedom that the multi-label setting introduces. In our experiments, we used various evaluation measures that have been suggested by Tsoumakas and Katakis (2007).

In particular, we used six *example-based* evaluation measures (*Hamming loss*, *accuracy*, *precision*, *recall*, *F<sub>1</sub> score* and *subset accuracy*) and six *label-based* evaluation measures

(*micro precision, micro recall, micro  $F_1$ , macro precision, macro recall and macro  $F_1$* ). Note that these evaluation measures require predictions stating that a given label is present or not (binary 1/0 predictions). However, most predictive models predict a numerical value for each label and the label is predicted as present if that numerical value exceeds some pre-defined threshold  $\tau$ . The performance of the predictive model thus directly depends on the selection of an appropriate value of  $\tau$ . To this end, we applied a threshold calibration method by choosing the threshold (6) that minimizes the difference in label cardinality between the training data and the predictions for the test data (Read et al. 2009).

$$\tau = \underset{\tau \in \{0.00, 0.05, \dots, 1.00\}}{\operatorname{argmin}} |LabelCard(E^{train}) - LabelCard(H_\tau(E^{test}))| \quad (6)$$

where  $E^{train}$  is the training set and a classifier  $H_\tau$  has made predictions for test set  $E^{test}$  under threshold  $\tau$ . We do not use the output space of the test set while calculating the threshold.

Also, we used four *ranking-based* evaluation measures (*one-error, coverage, ranking loss and average precision*) that compare the predicted ranking of the labels with the ground truth ranking. A detailed description of the evaluation measures is given in the Appendix A.

### 6.3 Experimental setup

The comparison of the multi-label learning methods was performed using the MULAN<sup>3</sup> library for the machine learning framework WEKA (Hall et al. 2009) and CLUS<sup>4</sup> system for predictive clustering. The MULAN library was used for BR and HOMER, and the CLUS system for the PCTs based methods. All experiments were performed on a server with an Intel Xeon processor at 2.50GHz and 64GB of RAM with the Fedora 14 operating system. In the remainder of this section, we first state the base classifiers that were used for the HOMER and the BR methods and then the parameter instantiations of all methods.

HOMER uses Support Vector Machines (SVM) as base classifiers for solving the partial binary classification problems. Binary Relevance classifier is used as a multi-label classifier in the internal nodes of the HOMER. For training the SVMs, we used the implementation from the LIBSVM library (Chang and Lin 2001). In particular, we used SVMs with a radial basis kernel.

The kernel parameter *gamma* and the penalty *C* were determined for each dataset by 10-fold cross validation using only the training sets. The values  $2^{-15}, 2^{-13}, \dots, 2^1, 2^3$  were considered for *gamma* and  $2^{-5}, 2^{-3}, \dots, 2^{13}, 2^{15}$  for the penalty *C*. For tuning the parameters we have used the *Hamming loss* evaluation measure. After determining the best parameters values on each dataset, the classifiers were trained using all available training examples and were evaluated by recognizing all test examples from the corresponding dataset.

The same settings (experimental design and setup) were used for the global BR approach. For the single PCT approaches (PCTs for multi-target classification - MTC, HSC

<sup>3</sup><http://mulan.sourceforge.net/>

<sup>4</sup><http://clus.sourceforge.net>

architecture of binary classifiers and PCTs for HMC) we used the default settings of CLUS. The ensemble methods PCTs for multi-target classification, PCTs for hierarchical single-label learning and PCTs for hierarchical multi-label learning learn 100 models as suggested by Bauer and Kohavi (1999). For the size of the feature subsets needed for construction of the base classifiers for the ensembles we selected  $f(x) = \lfloor 0.1 \cdot |x| + 1 \rfloor$  as recommended by Kocev (2011). The weight parameter  $w_0$  for all PCTs based approaches for solving HMC problems is set to 0.75.

HOMER, HSC and the HMC methods require one additional parameter to be configured: the number of clusters  $k$  for the balanced  $k$ -means clustering algorithm. For this parameter, five different values (2–6) were considered for HOMER in the cross-validation (Tsoumakas et al. 2008). After determining the best value of  $k$  on every dataset (via cross-validation on the training dataset), HOMER was trained using all available training examples and was evaluated by recognizing all test examples from the corresponding dataset. The same hierarchies (the same values for  $k$  obtained for HOMER) were used for the HSC and HMC methods. The values of the parameter  $k$  are 3 for most of the datasets, 2 for the emotions dataset, 5 for the yeast dataset, and 4 for the enron and delicious datasets.

To investigate the utility of the data-derived hierarchy we perform pairwise comparison between the methods that use the hierarchy and their counterparts that do not use the hierarchy. In each of the comparisons the performance is compared in terms of the 16 different performance measures. To assess whether the difference in performance are statistically significant, we have employed the non-parametric Wilcoxon test for statistical significance (Demšar 2006).

## 7 Results and discussion

In this section, we present the results from the experimental evaluation. Table 3 shows the values for the statistical significance level ( $p$ ) of the difference (as measured by the non-parametric Wilcoxon test for statistical significance) between the methods that use the data-defined label hierarchy and the methods that don't. In particular, the following pairs of methods are compared:

- BR and HOMER
- PCTs for multi-target classification (labeled as MTC) and hierarchical architecture of binary PCTs (labeled as HSC)
- PCTs for multi-target classification (labeled as MTC) and PCTs for hierarchical multi-label classification (labeled as HMC)
- Random forests of PCTs for multi-target classification (labeled as RFMTC) and HSC architectures of random forests of PCTs for hierarchical classification (labeled as RFHSC)
- Random forests of PCTs for multi-target classification (labeled as RFMTC) and random forests of PCTs for hierarchical multi-label classification (labeled as RFHMC)

In this experimental evaluation, we did not include the BR approach that use single PCTs and ensembles of PCTs (in particular, Random Forest of PCTs) as base classifiers. This decision was made as a result of a recent experimental evaluation (Levatić et al. 2014) in which the authors show that the predictive performance of this combination (denoted

as single-label classification) is clearly the worst for single predictive models, and only comparable to the other methods for ensembles.

The first column of the table lists the evaluation measures, while the other two columns show the values of the significance level  $p$ . The sign '>' in the 3rd, 6th, 9th, 12th and 15th columns indicates that the first method (out of the two compared methods in the corresponding pairwise comparison) is better than the second method, while the sign '<' indicates that the second method is better than the first method. The difference between the methods' performance is statistically significant if the value of the significance level ( $p$ ) is lower than 0.05. These values are shown in boldface.

The results clearly show that the methods that use the data-derived hierarchies outperform the methods that do not use those hierarchies. The methods that use the data-derived hierarchies have 31 significant wins against only 3 significant wins of the methods that do not use the hierarchies. Both, local and global approaches show similar improvements which are more pronounced on single models than ensembles.

Inspecting Table 3 (BR vs HOMER), we note that HOMER performs better than BR on example-based and label-based measures overall (8 wins vs 4 losses, 5 significant wins vs. 1 significant loss). It performs better on all recall-based measures,  $F_1$ -based measures, *accuracy* and *subset accuracy* and worse on *Hamming loss* and precision-based measures. HOMER is significantly better on *recall*, *micro recall*,  $F_1$  score, *micro  $F_1$*  and *accuracy*, and worse on *macro precision*.

Moving on to the results for comparing single PCTs that use and don't use the data-derived hierarchies (MTC vs HSC and MTC vs HMC), we can immediately see that both (local and global) single PCTs models that use the hierarchies perform consistently better than single PCTs for MTC (that don't use the hierarchies). The HSC architectures perform better on 14 evaluation measures (significantly on 10) and worse (but not significantly) on 2. PCTs for HMC beat PCTs for MTC even more convincingly, performing better on all but one evaluation measure: In terms of significant wins, the score is 13:0.

Finally, the comparisons RFMTC vs RFHSC and RFMTC vs RFHMC concern the use of the label hierarchy in the context of learning PCT ensembles (Random Forest of PCTs), both HSC architectures of binary PCT ensembles and ensembles of PCTs for HMC. As a baseline, we use ensembles (RFs) of PCTs for MTC. The HSC architecture actually performs slightly worse than RFs of PCTs for MTC (the significant wins score is 0:2). RFs of PCTs for HMC, however, perform better than RFs of PCTs for MTC (better on 12 and worse on 4 evaluation measures, significant wins score 3:0). It is obvious that the difference in the predictive performance is not emphasized as for the single models. Much larger improvements of performance are hard to achieve, given that RFs of PCTs for MTC were the best performing approach from a recent comparative study (Madjarov et al. 2012) (that did not use a label hierarchy).

The results in our research reveal that, if we have a very strong classifier and it can exploit in deep the label relationships, it will always produce good predictive performance. But, by using the label relationships (in particular, represented by data-derived hierarchies) as an additional input to the classifiers, we can improve (and not decrease) the predictive performance. In our work, this is emphasized especially for the single decision trees (which are not powerful classifiers, but very important because of their knowledge extraction and representation capabilities) and less emphasized for the ensembles and SVM-based methods.

**Table 3** The values of the significance level  $p$  from the non-parametric Wilcoxon test of statistical significance for the difference in performance along 16 performance evaluation measures

	BR vs HOMER		MTC vs HSC		MTC vs HMC		RFMTC vs RFHSC		RFMTC vs RFHMC	
	p-value	>	p-value	>	p-value	>	p-value	>	p-value	>
Example-based										
Hamming loss	0.169	>	0.929	>	0.266	>	0.016	>	0.131	>
Hamming loss	0.169	>	0.929	>	0.266	>	0.016	>	0.131	>
Accuracy	0.017	<	0.050	<	0.013	<	0.534	<	0.075	<
Precision	0.508	>	0.131	<	0.016	<	0.780	<	0.266	<
Recall	0.005	<	0.004	<	0.010	<	0.722	<	0.016	<
F1 score	0.009	<	0.021	<	0.008	<	0.722	<	0.016	<
Subset accuracy	0.721	<	0.424	<	0.130	<	0.061	<	0.197	<
Label-based										
Micro precision	0.333	>	0.534	>	0.859	>	0.182	>	0.965	>
Micro recall	0.005	<	0.003	<	0.010	<	0.563	<	0.689	<
Micro F1	0.047	<	0.016	<	0.007	<	0.477	<	0.594	<
Macro precision	0.005	>	0.026	<	0.021	<	0.040	>	0.929	>
Macro recall	0.139	<	0.004	<	0.010	<	0.213	>	0.790	>
Macro F1	0.508	<	0.008	<	0.021	<	0.075	>	0.929	>
Ranking-based										
One error	n/a	<	0.021	<	0.013	<	0.929	<	0.929	<
Coverage	n/a	<	0.182	<	0.010	<	0.091	<	0.033	<
Ranking loss	n/a	<	0.041	<	0.016	<	0.050	<	0.213	<
Average precision	n/a	<	0.021	<	0.008	<	0.657	<	0.965	<

**Table 3** (continued)

	BR vs HOMER		MTC vs HSC		MTC vs HMC		RFMTC vs RFHSC		RFMTC vs RFHMC			
	Wins	Sig. wins	Wins	Sig. wins	Wins	Sig. wins	Wins	Sig. wins	Wins	Sig. wins		
Wilcoxon test	BR	4	1	MTC	2	0	MTC	1	0	RFMTC	4	0
	HOMER	8	5	HSC	14	10	HMC	15	13	RFHSC	4	0
										RFHMC	12	3

Five pairwise comparison we considered: BR vs HOMER, MTC vs HSC, MTC vs HMC, RFMTC vs RFHSC and RFMTC vs RFHMC. The sign ' $</>$ ' in column  $x$  vs  $y$  and row  $z$ , means that  $x$  performs better than  $y$  in terms of measures  $z$ . If the corresponding  $p$ -value is shown in bold, the difference in performance is significant



**Fig. 5** Visualization of the hierarchies of the *tmc2007* and *enron* datasets and the number of examples in the nodes of the hierarchies represented by the size of the circles. Circles of the original labels are annotated, while the circles that represent the meta-labels are not annotated

In the Appendix A, we have also shown the average ranking diagrams depicting the relative performance of the 8 methods considered in the experiments: BR, HOMER, PCTs for MTC, HSC of PCTs, PCTs for HMC, RFs of PCTs for MTC, HSC of RFs of PCTs, RFs of PCTs for HMC. These indicate that HOMER performs best on recall-related measures (incl. *recall*, *micro recall* and *macro recall*), *micro F<sub>1</sub>*, *macro F<sub>1</sub>*, and *accuracy*. RFs for HMC, on the other hand, perform best on *precision* and ranking-related measures (incl. *micro precision* and *macro precision*, *Hamming loss*, *subset accuracy*, *ranking loss*, *one-error*, *coverage* and *average precision*).

A more detailed analysis of the results, looking at the performance figures of each methods for individual datasets (given in Appendix B) provides additional insight. The methods that use the data-derived hierarchy (both local and global approaches for HMC), generally show better performance on datasets with a large number of labels. Also, the analysis show that for the datasets with a large number of labels, many labels have only few examples (the labels space is sparse) and the data-derived hierarchy reduces the sparsity of the label space. We believe that this is the main reason that the balanced k-means approach works better than the hierarchical agglomerative clustering methods (work presented by Madjarov et al. 2015). In particular, balanced k-means tries to create more balanced clusters in terms of the number of labels and their co-occurrence in the partial classification problems defined in each node of the tree. That means that the number of examples in a particular node among the different labels is similar and the new (hierarchical) classification problem is more balanced in comparison to the classification problem defined by the original (flat) organization of the labels. Figures 5a, b visualize the hierarchies of the *tmc2007* and *enron* datasets and the number of examples in each node of the hierarchy. The size of the circles corresponds to the number of the examples in the training set labeled with a particular label. Circles of the original labels are annotated, while the circles that represent the meta-labels are not annotated.

## 8 Conclusions and further work

In this paper, we have investigated the use of label hierarchies, constructed in a data-driven manner, in multi-label classification. We consider flat label-sets and construct label hierarchies from the label sets that appear in the annotations of the training data by using a hierarchical clustering approaches based on balanced  $k$ -means clustering. The hierarchies are then used in conjunction with hierarchical multi-label classification approaches (in a global and a local setting) in the hope of achieving better multi-label classification.

While the use of hierarchies constructed in this manner has been proposed a few years ago, it has only been considered in conjunction with a local model approach to HMC (Tsoumakas et al. 2008). In particular, a binary relevance hierarchical MLC architecture based on SVMs has been considered and evaluated on two datasets in the light of a few performance measures. We conduct a much more thorough study, investigating the utility of the hierarchy in the context of two local and two global model approaches to HMC, on a large collection of datasets, through the prism of a large number of performance measures.

In particular, we investigate the utility of the hierarchy in the context of two hierarchical architectures that use binary relevance classifier based on SVMs and decision trees (predictive clustering trees for binary classification), and the global model approaches of PCTs for HMC and ensembles thereof. The experimental results clearly show that the use of the hierarchy results in improved performance.

The performance is improved both in the context of local approaches and the context of global approaches for HMC. For the local approaches, the performance is improved when using a binary relevance architecture with SVMs (significantly better performance along 5 measures). It results in even more obvious improvements for a binary relevance architecture with decision trees (PCTs, significantly better performance along 10 measures).

In the context of global approaches, the label hierarchy used in PCTs for HMC greatly improves the performance of PCTs for multi-target classification (as used for MLC): The results show improvement in performance on 15 of the 16 measures considered, significantly for 13 measures (and insignificantly worse on one measure). For ensembles (RFs) of PCTs, the use of the hierarchy improves performance along 12 of the 16 measures, but the difference is significant for only 3 measures. However, note that RFs of PCTs perform very well already even without the hierarchy, being the best performing method among many MLC methods considered in a recent extensive experimental comparison.

We also believe that the use of the label hierarchies can lead to better understanding of the learned predictive models, but we have to investigate this in our further work. Additionally, we plan to extend this study by comparison of hierarchies constructed by humans and hierarchies generated in a data-driven fashion. For HMC problems, we can consider the MLC task defined by the leaves of the provided label hierarchy. We can then construct label hierarchies automatically, as described above, and compare these hierarchies (and their utility) to the originally provided label hierarchy. Also, some other types of structures (such as DAG, DMOZ hierarchy and etc.) could be considered for capturing the dependencies and relations between the labels of the original multi-label classification problems.

**Acknowledgments** We would like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944).

## Appendix A: Evaluation measures

In this section, we present the measures that are used to evaluate the predictive performance of the compared methods in our experiments. In the definitions below,  $\mathcal{Y}_i$  denotes the set of true labels of example  $\mathbf{x}_i$  and  $h(\mathbf{x}_i)$  denotes the set of predicted labels for the same examples. All definitions refer to the multi-label setting.

### A.1 Example based measures

**Hamming loss** evaluates how many times an example-label pair is misclassified, i.e., label not belonging to the example is predicted or a label belonging to the example is not predicted. The smaller the value of *hamming\_loss*( $h$ ), the better the performance. The performance is perfect when *hamming\_loss*( $h$ ) = 0. This metric is defined as:

$$hamming\_loss(h) = \frac{1}{N} \sum_{i=1}^N \frac{1}{Q} |h(\mathbf{x}_i) \Delta \mathcal{Y}_i| \tag{7}$$

where  $\Delta$  stands for the symmetric difference between two sets,  $N$  is the number of examples and  $Q$  is the total number of possible class labels.

**Accuracy** for a single example  $\mathbf{x}_i$  is defined by the Jaccard similarity coefficients between the label sets  $h(\mathbf{x}_i)$  and  $\mathcal{Y}_i$ . Accuracy is micro-averaged across all examples.

$$accuracy(h) = \frac{1}{N} \sum_{i=1}^N \frac{|h(\mathbf{x}_i) \cap \mathcal{Y}_i|}{|h(\mathbf{x}_i) \cup \mathcal{Y}_i|} \tag{8}$$

**Precision** is defined as:

$$precision(h) = \frac{1}{N} \sum_{i=1}^N \frac{|h(\mathbf{x}_i) \cap \mathcal{Y}_i|}{|h(\mathbf{x}_i)|} \tag{9}$$

**Recall** is defined as:

$$recall(h) = \frac{1}{N} \sum_{i=1}^N \frac{|h(\mathbf{x}_i) \cap \mathcal{Y}_i|}{|\mathcal{Y}_i|} \tag{10}$$

**F<sub>1</sub> score** is the harmonic mean between precision and recall and is defined as:

$$F_1 = \frac{1}{N} \sum_{i=1}^N \frac{2 \times |h(\mathbf{x}_i) \cap \mathcal{Y}_i|}{|h(\mathbf{x}_i)| + |\mathcal{Y}_i|} \tag{11}$$

$F_1$  is an example based metric and its value is an average over all examples in the dataset.  $F_1$  reaches its best value at 1 and worst score at 0.

**Subset accuracy** or classification accuracy is defined as follows:

$$subset\_accuracy(h) = \frac{1}{N} \sum_{i=1}^N I(h(\mathbf{x}_i) = \mathcal{Y}_i) \tag{12}$$

where  $I(true) = 1$  and  $I(false) = 0$ . This is a very strict evaluation measure as it requires the predicted set of labels to be an exact match of the true set of labels.

### A.2 Label based measures

**Macro precision** (precision averaged across all labels) is defined as:

$$macro\_precision = \frac{1}{Q} \sum_{j=1}^Q \frac{tp_j}{tp_j + fp_j} \tag{13}$$

where  $tp_j, fp_j$  are the number of true positives and false positives for the label  $\lambda_j$  considered as a binary class.

**Macro recall** (recall averaged across all labels) is defined as:

$$macro\_recall = \frac{1}{Q} \sum_{j=1}^Q \frac{tp_j}{tp_j + fn_j} \tag{14}$$

where  $tp_j, fp_j$  are defined as for the macro precision and  $fn_j$  is the number of false negatives for the label  $\lambda_j$  considered as a binary class.

**Macro  $F_1$**  is the harmonic mean between precision and recall, where the average is calculated per label and then averaged across all labels. If  $p_j$  and  $r_j$  are the precision and recall for all  $\lambda_j \in h(\mathbf{x}_i)$  from  $\lambda_j \in \mathcal{Y}_i$ , the macro  $F_1$  is

$$macro\_F_1 = \frac{1}{Q} \sum_{j=1}^Q \frac{2 \times p_j \times r_j}{p_j + r_j} \tag{15}$$

**Micro precision** (precision averaged over all the example/label pairs) is defined as:

$$micro\_precision = \frac{\sum_{j=1}^Q tp_j}{\sum_{j=1}^Q tp_j + \sum_{j=1}^Q fp_j} \tag{16}$$

where  $tp_j, fp_j$  are defined as for macro precision.

**Micro recall** (recall averaged over all the example/label pairs) is defined as:

$$micro\_recall = \frac{\sum_{j=1}^Q tp_j}{\sum_{j=1}^Q tp_j + \sum_{j=1}^Q fn_j} \tag{17}$$

where  $tp_j$  and  $fn_j$  are defined as for macro recall.

**Micro  $F_1$**  is the harmonic mean between micro precision and micro recall. Micro  $F_1$  is defined as:

$$micro\_F_1 = \frac{2 \times micro\_precision \times micro\_recall}{micro\_precision + micro\_recall} \tag{18}$$

### A.3 Ranking based measures

**One error** evaluates how many times the top-ranked label is not in the set of relevant labels of the example. The metric  $one\_error(f)$  takes values between 0 and 1. The smaller the value of  $one\_error(f)$ , the better the performance. This evaluation metric is defined as:

$$one\_error(f) = \frac{1}{N} \sum_{i=1}^N \left[ \left[ \arg \max_{\lambda \in \mathcal{Y}} f(\mathbf{x}_i, \lambda) \right] \notin \mathcal{Y}_i \right] \tag{19}$$

where  $\lambda \in \mathcal{L} = \{\lambda_1, \lambda_2, \dots, \lambda_Q\}$  and  $\llbracket \pi \rrbracket$  equals 1 if  $\pi$  holds and 0 otherwise for any predicate  $\pi$ . Note that, for single-label classification problems, the One Error is identical to ordinary classification error.

**Coverage** evaluates how far, on average, we need to go down the list of ranked labels in order to cover all the relevant labels of the example. The smaller the value of  $coverage(f)$ , the better the performance.

$$coverage(f) = \frac{1}{N} \sum_{i=1}^N \max_{\lambda \in \mathcal{Y}_i} rank_f(\mathbf{x}_i, \lambda) - 1 \tag{20}$$

where  $rank_f(\mathbf{x}_i, \lambda)$  denotes the position of the label  $\lambda$  in the ranking. It maps the outputs of  $f(\mathbf{x}_i, \lambda)$  for any  $\lambda \in \mathcal{L}$  to  $\{\lambda_1, \lambda_2, \dots, \lambda_Q\}$  so that  $f(\mathbf{x}_i, \lambda_m) > f(\mathbf{x}_i, \lambda_n)$  implies  $rank_f(\mathbf{x}_i, \lambda_m) < rank_f(\mathbf{x}_i, \lambda_n)$ . The smallest possible value for  $coverage(f)$  is  $l_c$ , i.e., the label cardinality of the given dataset.

**Ranking loss** evaluates the average fraction of label pairs that are reversely ordered for the particular example given by:

$$ranking\ loss(f) = \frac{1}{N} \sum_{i=1}^N \frac{|D_i|}{|\mathcal{Y}_i| |\bar{\mathcal{Y}}_i|} \tag{21}$$

where  $D_i = \{(\lambda_m, \lambda_n) | f(\mathbf{x}_i, \lambda_m) \leq f(\mathbf{x}_i, \lambda_n), (\lambda_m, \lambda_n) \in \mathcal{Y}_i \times \bar{\mathcal{Y}}_i\}$ , while  $\bar{\mathcal{Y}}$  denotes the complementary set of  $\mathcal{Y}$  in  $\mathcal{L}$ . The smaller the value of  $ranking\_loss(f)$ , the better the performance, so the performance is perfect when  $ranking\_loss(f) = 0$ .

**Average Precision** is the average fraction of labels ranked above an actual label  $\lambda \in \mathcal{Y}_i$  that actually are in  $\mathcal{Y}_i$ . The performance is perfect when  $avg\_precision(f) = 1$ ; the larger the value of  $avg\_precision(f)$ , the better the performance. This metric is defined as:

$$avg\_precision(f) = \frac{1}{N} \sum_{i=1}^N \frac{1}{|\mathcal{Y}_i|} \sum_{\lambda \in \mathcal{Y}_i} \frac{|\mathcal{L}_i|}{rank_f(\mathbf{x}_i, \lambda)} \tag{22}$$

where  $\mathcal{L}_i = \{\lambda' | rank_f(\mathbf{x}_i, \lambda') \leq rank_f(\mathbf{x}_i, \lambda), \lambda' \in \mathcal{Y}_i\}$  and  $rank_f(\mathbf{x}_i, \lambda)$  is defined as in coverage above.

## Appendix B: Complete results from the experimental evaluation

In this section, we present the complete results from the experimental evaluation. We present the results based on the evaluation measures. Tables 4, 5, 6, 7 and 8 give the performance of the compared methods on each of the datasets measured in terms of the example based, label based and ranking based evaluation measures. The first column of the tables lists the dataset, while the remaining columns show the performance of each method for every dataset. The best results per dataset are shown in boldface. For the *bookmarks* dataset, HOMER did not manage to construct a predictive model within one week under the available resources. The corresponding entries in the tables with the results are marked with DNF (Did Not Finish).

To assess whether the overall differences in performance across the different approaches are statistically significant, we also employed the corrected Friedman test (Friedman 1940) and the post-hoc Nemenyi test (Nemenyi 1963) as recommended by Demšar (2006). We present the results from the Nemenyi post-hoc test with average rank diagrams (Demšar 2006). These are given in Figs. 6, 7 and 8. A critical diagram contains an enumerated axis on which the average ranks of the algorithms are drawn. The algorithms are depicted along the axis in such a manner that the best ranking ones are at the right-most side of the diagram. The lines for the average ranks of the algorithms that do not differ significantly (at the significance level of  $p=0.05$ ) are connected with a line. For the *bookmarks* dataset, we

penalize HOMER that does not finish by assigning it the lowest value (i.e., the lowest rank value) for each evaluation measure.

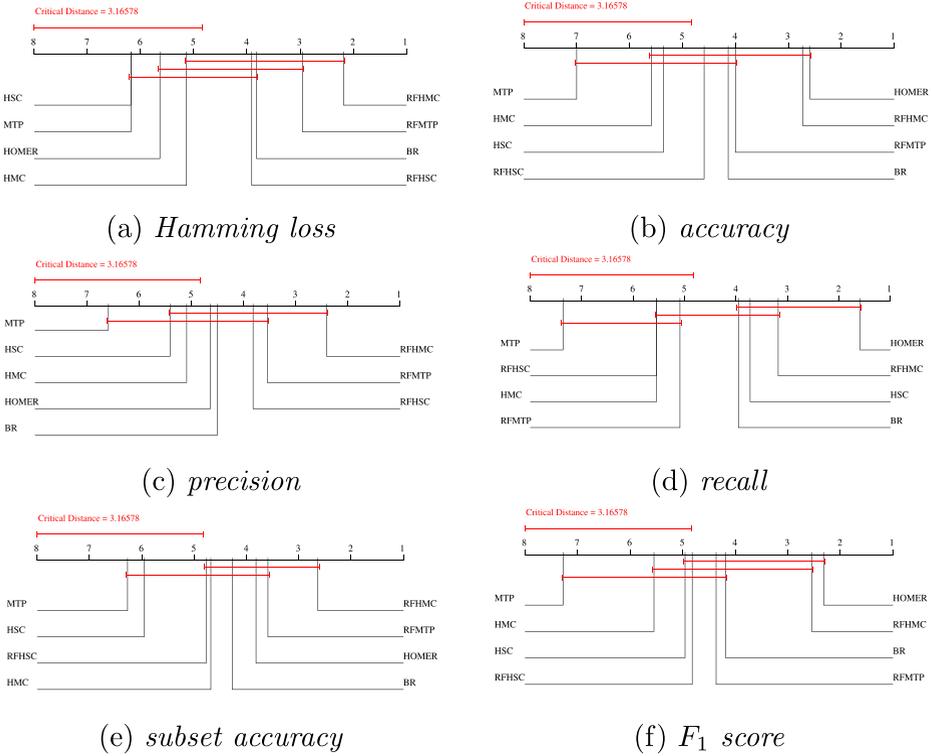
## B.1 Results on the example-based evaluation measures

**Table 4** The performance of the multi-label classification approaches in terms of the examples-based evaluation measures

	BR	HOMER	MTP	HSC	HMC	RFMTP	RFHSC	RFHMC
Hamming loss								
emotions	0.257	0.361	0.267	0.257	0.274	<b>0.189</b>	0.192	<b>0.189</b>
scene	<b>0.079</b>	0.082	0.129	0.161	0.142	0.094	0.097	0.08
yeast	<b>0.19</b>	0.207	0.219	0.276	0.216	0.197	0.198	0.195
medical	0.077	<b>0.012</b>	0.023	0.013	0.014	0.014	0.015	<b>0.012</b>
enron	<b>0.045</b>	0.051	0.058	0.055	0.052	0.046	0.046	0.046
corel5k	0.017	0.012	<b>0.009</b>	0.01	<b>0.009</b>	<b>0.009</b>	<b>0.009</b>	<b>0.009</b>
tmc2007	0.013	0.015	0.075	0.059	0.067	<b>0.011</b>	0.027	0.015
mediamill	0.032	0.038	0.034	0.043	0.033	<b>0.029</b>	0.03	<b>0.029</b>
bibtex	<b>0.012</b>	0.014	0.014	0.015	0.013	0.013	0.013	<b>0.012</b>
delicious	<b>0.018</b>	0.022	0.019	<b>0.018</b>	<b>0.018</b>	<b>0.018</b>	<b>0.018</b>	<b>0.018</b>
bookmarks	DNF	DNF	0.009	0.01	0.009	0.009	0.009	<b>0.008</b>
accuracy								
emotions	0.361	0.471	0.448	0.441	0.419	0.519	<b>0.535</b>	0.52
scene	0.689	<b>0.717</b>	0.538	0.449	0.523	0.541	0.508	0.555
yeast	0.52	<b>0.559</b>	0.44	0.401	0.469	0.478	0.491	0.475
medical	0.206	0.713	0.228	0.639	0.665	0.591	0.538	<b>0.715</b>
enron	0.446	<b>0.478</b>	0.196	0.415	0.37	0.416	0.431	0.426
corel5k	0.03	<b>0.179</b>	0	0.058	0.021	0.009	0.021	0.023
tmc2007	0.891	0.888	0.436	0.562	0.515	0.914	0.785	<b>0.916</b>
mediamill	0.403	0.413	0.354	0.349	0.386	<b>0.441</b>	0.431	0.437
bibtex	<b>0.348</b>	0.33	0.046	0.243	0.254	0.166	0.197	0.256
delicious	0.136	<b>0.207</b>	0.001	0.129	0.118	0.146	0.125	0.138
bookmarks	DNF	DNF	0.133	<b>0.223</b>	0.205	0.204	0.195	0.209
precision								
emotions	0.55	0.509	0.577	0.533	0.587	0.644	<b>0.672</b>	0.637
scene	0.718	<b>0.746</b>	0.565	0.463	0.547	0.565	0.533	0.555
yeast	0.722	0.663	0.705	0.553	0.68	0.744	0.73	<b>0.747</b>
medical	0.211	<b>0.762</b>	0.285	0.685	0.721	0.635	0.572	0.731
enron	0.703	0.616	0.415	0.616	0.61	0.709	0.71	<b>0.713</b>
corel5k	0.042	<b>0.317</b>	0	0.162	0.061	0.03	0.074	0.071
tmc2007	0.941	0.926	0.659	0.731	0.688	<b>0.977</b>	0.923	0.966
mediamill	0.731	0.597	0.694	0.524	0.716	0.772	0.767	<b>0.774</b>
bibtex	<b>0.515</b>	0.472	0.14	0.368	0.404	0.292	0.347	0.423
delicious	0.443	0.369	0.001	0.44	0.429	0.512	<b>0.528</b>	0.512
bookmarks	DNF	DNF	0.133	<b>0.261</b>	0.224	0.218	0.209	0.225

**Table 5** The performance of the multi-label classification approaches in terms of the examples-based evaluation measures

	BR	HOMER	MTP	HSC	HMC	RFMTP	RFHSC	RFHMC
recall								
emotions	0.409	<b>0.775</b>	0.534	0.594	0.501	0.582	0.618	0.588
scene	0.711	<b>0.744</b>	0.539	0.536	0.538	0.541	0.511	0.623
yeast	0.591	<b>0.714</b>	0.49	0.557	0.549	0.523	0.548	0.645
medical	0.735	<b>0.76</b>	0.228	0.671	0.692	0.599	0.549	0.69
enron	0.497	<b>0.61</b>	0.229	0.511	0.412	0.452	0.477	0.54
corel5k	0.055	<b>0.25</b>	0	0.066	0.022	0.009	0.021	0.024
tmc2007	0.928	<b>0.943</b>	0.478	0.661	0.604	0.92	0.811	0.885
mediamill	0.45	<b>0.563</b>	0.379	0.489	0.427	0.476	0.465	0.488
bibtex	0.373	<b>0.389</b>	0.046	0.29	0.271	0.167	0.199	0.26
delicious	0.155	<b>0.303</b>	0.001	0.15	0.132	0.16	0.134	0.171
bookmarks	DNF	DNF	0.137	<b>0.24</b>	0.211	0.208	0.199	0.214
F1 score								
emotions	0.469	0.614	0.554	0.562	0.54	0.611	<b>0.644</b>	0.612
scene	0.714	<b>0.745</b>	0.552	0.497	0.542	0.553	0.522	0.587
yeast	0.65	0.687	0.578	0.555	0.607	0.614	0.626	<b>0.692</b>
medical	0.328	<b>0.761</b>	0.253	0.678	0.706	0.616	0.56	0.71
enron	0.582	0.613	0.295	0.558	0.492	0.552	0.57	<b>0.615</b>
corel5k	0.047	<b>0.28</b>	0	0.093	0.032	0.014	0.032	0.036
tmc2007	0.934	0.934	0.554	0.694	0.643	<b>0.948</b>	0.863	0.924
mediamill	0.557	0.579	0.49	0.506	0.535	0.589	0.579	<b>0.599</b>
bibtex	<b>0.433</b>	0.426	0.069	0.324	0.324	0.212	0.253	0.322
delicious	0.23	<b>0.343</b>	0.001	0.224	0.201	0.244	0.214	0.256
bookmarks	DNF	DNF	0.135	<b>0.25</b>	0.217	0.213	0.204	0.219
subset accuracy								
emotions	0.129	0.163	0.223	0.173	0.144	0.307	0.287	<b>0.308</b>
scene	0.639	<b>0.661</b>	0.509	0.356	0.483	0.518	0.482	0.508
yeast	0.19	<b>0.213</b>	0.153	0.074	0.138	0.152	0.161	0.182
medical	0	0.61	0.177	0.562	0.58	0.538	0.492	<b>0.622</b>
enron	<b>0.149</b>	0.145	0.002	0.112	0.097	0.131	0.121	0.136
corel5k	0	<b>0.002</b>	0	0	<b>0.002</b>	0	0	0
tmc2007	0.772	0.765	0.215	0.28	0.253	<b>0.816</b>	0.588	0.748
mediamill	0.08	0.053	0.065	0.056	0.082	0.122	0.113	<b>0.128</b>
bibtex	<b>0.194</b>	0.165	0.004	0.113	0.144	0.098	0.115	0.165
delicious	0.004	0.001	0.001	0.003	<b>0.007</b>	<b>0.007</b>	0.005	<b>0.007</b>
bookmarks	DNF	DNF	0.129	0.187	0.188	0.189	0.181	<b>0.193</b>



**Fig. 6** The critical diagrams for the example-based evaluation measures: The results from the Nemenyi post-hoc test at 0.05 significance level on all the datasets

### B.2 Results on the label-based evaluation measures

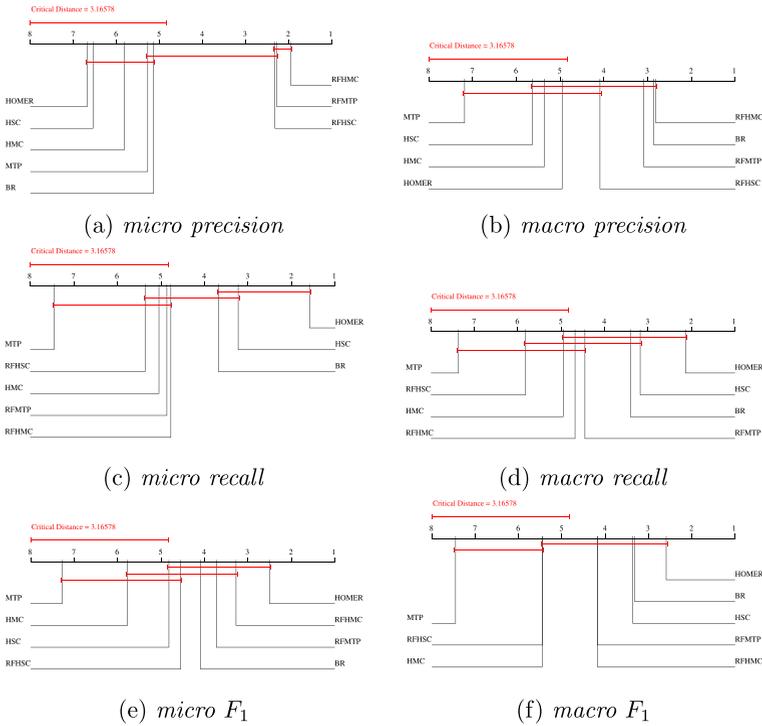
The big difference between the micro-based and macro-based evaluation measures appears due to the averaging strategy of the obtained predictions. It is more emphasized on the large datasets that have highly unbalanced number of examples per label. Namely, the averaging in the micro-based measures is made across the predictions per example for all labels, while the averaging in the macro-based measures is made across the predictions per label for all examples, which means that for macro-based measures the labels with small number of examples are equally important as the labels with large number of examples.

**Table 6** The performance of the multi-label classification approaches in terms of the label-based evaluation measures

	BR	HOMER	MTP	HSC	HMC	RFMTP	RFHSC	RFHMC
micro precision								
emotions	0.684	0.471	0.607	0.61	0.602	0.783	0.749	<b>0.785</b>
scene	0.843	0.804	0.692	0.56	0.63	0.93	<b>0.935</b>	0.932
yeast	0.733	0.647	0.699	0.546	0.68	<b>0.755</b>	0.737	0.749
medical	0.225	0.807	0.826	0.828	0.812	0.885	0.89	<b>0.895</b>
enron	0.721	0.597	0.602	0.582	0.646	0.738	0.726	<b>0.74</b>
corel5k	0.061	0.308	0	0.298	0.52	0.696	0.578	<b>0.699</b>
tmc2007	0.947	0.922	0.689	0.751	0.704	<b>0.992</b>	0.949	0.987
mediamill	0.742	0.569	0.743	0.502	0.733	0.798	<b>0.8</b>	<b>0.8</b>
bibtex	0.753	0.547	<b>1</b>	0.55	0.733	0.957	0.932	0.912
delicious	0.658	0.396	0	0.579	0.621	0.695	<b>0.714</b>	0.709
bookmarks	DNF	DNF	<b>0.947</b>	0.502	0.776	0.895	0.912	0.885
micro recall								
emotions	0.406	<b>0.782</b>	0.539	0.604	0.496	0.589	0.627	0.569
scene	0.694	<b>0.727</b>	0.521	0.523	0.527	0.523	0.497	0.514
yeast	0.587	<b>0.702</b>	0.492	0.544	0.545	0.521	0.542	0.525
medical	0.725	<b>0.742</b>	0.227	0.658	0.66	0.569	0.524	0.651
enron	0.464	<b>0.585</b>	0.247	0.473	0.386	0.422	0.445	0.431
corel5k	0.057	<b>0.248</b>	0	0.067	0.022	0.009	0.021	0.024
tmc2007	0.917	<b>0.932</b>	0.454	0.62	0.563	0.902	0.771	0.896
mediamill	0.415	<b>0.537</b>	0.351	0.459	0.393	0.435	0.422	0.427
bibtex	0.328	<b>0.353</b>	0.057	0.259	0.227	0.131	0.155	0.203
delicious	0.143	<b>0.297</b>	0	0.141	0.12	0.151	0.127	0.14
bookmarks	DNF	DNF	0.076	<b>0.171</b>	0.139	0.136	0.127	0.139
micro F1								
emotions	0.509	0.588	0.571	0.607	0.544	0.672	<b>0.682</b>	0.66
scene	0.761	<b>0.764</b>	0.594	0.541	0.574	0.669	0.649	0.663
yeast	0.652	<b>0.673</b>	0.577	0.545	0.605	0.617	0.625	0.617
medical	0.343	<b>0.773</b>	0.356	0.733	0.728	0.693	0.66	0.754
enron	0.564	<b>0.591</b>	0.35	0.522	0.483	0.537	0.552	0.545
corel5k	0.059	<b>0.275</b>	0	0.11	0.042	0.018	0.041	0.046
tmc2007	0.932	0.927	0.547	0.679	0.625	<b>0.945</b>	0.851	0.939
mediamill	0.533	0.553	0.477	0.48	0.512	<b>0.563</b>	0.552	0.557
bibtex	<b>0.457</b>	0.429	0.108	0.352	0.347	0.23	0.265	0.332
delicious	0.234	<b>0.339</b>	0	0.226	0.201	0.248	0.215	0.234
bookmarks	DNF	DNF	0.141	<b>0.256</b>	0.236	0.236	0.222	0.24

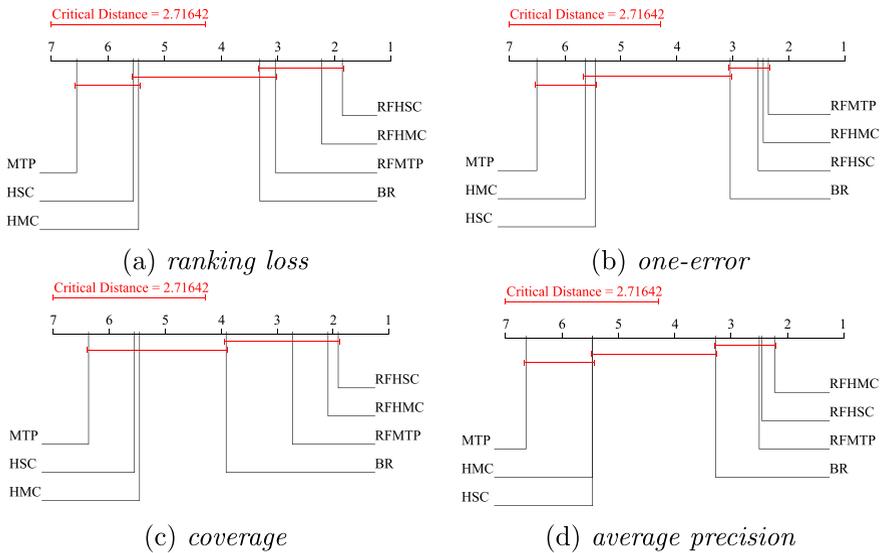
**Table 7** The performance of the multi-label classification approaches in terms of the label-based evaluation measures

	BR	HOMER	MTP	HSC	HMC	RFMTP	RFHSC	RFHMC
macro precision								
emotions	0.721	0.464	0.628	0.613	0.644	<b>0.802</b>	0.767	0.801
scene	0.844	0.807	0.682	0.569	0.629	0.919	<b>0.928</b>	0.911
yeast	0.628	0.471	0.479	0.388	0.445	<b>0.674</b>	0.587	0.58
medical	<b>0.399</b>	0.287	0.018	0.359	0.306	0.269	0.281	0.272
enron	<b>0.258</b>	0.241	0.023	0.19	0.101	0.233	0.214	0.246
corel5k	<b>0.052</b>	0.044	0	0.027	0.016	0.015	0.002	0.028
tmc2007	0.972	0.954	0.386	0.742	0.735	<b>0.997</b>	0.945	0.995
mediamill	0.112	0.107	0.04	0.161	0.217	0.441	0.306	<b>0.455</b>
bibtex	<b>0.528</b>	0.391	0.006	0.296	0.286	0.127	0.201	0.324
delicious	<b>0.299</b>	0.154	0	0.148	0.162	0.293	0.177	0.252
bookmarks	DNF	DNF	0.018	0.234	0.299	<b>0.522</b>	0.395	0.496
macro recall								
emotions	0.378	<b>0.775</b>	0.533	0.603	0.499	0.569	0.616	0.55
scene	0.703	<b>0.734</b>	0.529	0.524	0.538	0.533	0.504	0.525
yeast	0.355	<b>0.466</b>	0.269	0.389	0.308	0.286	0.301	0.285
medical	<b>0.423</b>	0.282	0.022	0.267	0.254	0.176	0.167	0.218
enron	0.12	<b>0.163</b>	0.03	0.135	0.077	0.1	0.091	0.104
corel5k	0.023	<b>0.041</b>	0	0.01	0.004	0.002	0.001	0.004
tmc2007	<b>0.915</b>	0.897	0.235	0.441	0.341	0.769	0.516	0.678
mediamill	0.049	0.074	0.029	<b>0.125</b>	0.054	0.08	0.051	0.081
bibtex	<b>0.25</b>	0.247	0.006	0.174	0.144	0.043	0.056	0.104
delicious	0.072	<b>0.103</b>	0	0.055	0.049	0.06	0.039	0.049
bookmarks	DNF	DNF	0.016	<b>0.095</b>	0.071	0.072	0.056	0.071
macro F1								
emotions	0.44	0.57	0.568	0.608	0.522	0.65	<b>0.666</b>	0.628
scene	0.765	<b>0.768</b>	0.592	0.545	0.578	0.658	0.64	0.648
yeast	0.392	<b>0.447</b>	0.293	0.388	0.327	0.322	0.326	0.321
medical	<b>0.361</b>	0.282	0.02	0.29	0.27	0.207	0.198	0.237
enron	0.143	<b>0.167</b>	0.026	0.147	0.082	0.122	0.106	0.127
corel5k	0.021	<b>0.036</b>	0	0.013	0.006	0.004	0.001	0.007
tmc2007	<b>0.942</b>	0.924	0.263	0.521	0.409	0.857	0.619	0.78
mediamill	0.056	0.073	0.031	<b>0.133</b>	0.07	0.112	0.067	0.115
bibtex	<b>0.307</b>	0.266	0.006	0.202	0.172	0.055	0.072	0.134
delicious	0.096	<b>0.103</b>	0	0.069	0.062	0.083	0.054	0.068
bookmarks	DNF	DNF	0.017	<b>0.123</b>	0.097	0.101	0.077	0.099



**Fig. 7** The critical diagrams for the label-based evaluation measures: The results from the Nemenyi post-hoc test at 0.05 significance level on all the datasets

### B.3 Results on the ranking-based evaluation measures



**Fig. 8** The critical diagrams for the ranking-based evaluation measures: The results from the Nemenyi post-hoc test at 0.05 significance level on all the datasets

**Table 8** The performance of the multi-label classification approaches in terms of the ranking-based evaluation measures

	BR	MTP	HSC	HMC	RFMTP	RFHSC	RFHMC
one error							
emotions	0.386	0.386	0.366	0.391	<b>0.262</b>	0.277	0.272
scene	<b>0.18</b>	0.389	0.386	0.413	0.21	0.211	0.225
yeast	0.236	0.264	<b>0.309</b>	0.256	0.248	0.244	0.239
medical	<b>0.135</b>	0.613	0.192	0.213	0.174	0.144	0.15
enron	0.237	0.392	0.302	0.28	0.221	0.223	<b>0.219</b>
corel5k	0.66	0.777	0.752	0.71	<b>0.608</b>	0.684	0.686
tmc2007	0.029	0.307	0.175	0.246	<b>0.006</b>	0.039	0.011
mediamill	0.188	0.22	0.233	0.197	<b>0.159</b>	0.162	0.162
bibtex	<b>0.346</b>	0.783	0.449	0.454	0.433	0.375	0.375
delicious	0.354	0.592	0.39	0.386	0.332	<b>0.317</b>	0.328
bookmarks	DNF	0.817	0.628	0.651	0.541	<b>0.526</b>	0.534
coverage							
emotions	2	2.356	2.134	2.223	<b>1.827</b>	1.851	1.866
scene	<b>0.399</b>	0.964	0.849	1.117	0.461	0.457	0.495
yeast	6.33	6.706	7.228	6.749	6.179	6.189	<b>6.177</b>
medical	1.61	5.381	2.177	3.408	1.619	1.919	<b>1.402</b>
enron	12.53	14.922	14.323	13.193	12.074	<b>11.261</b>	11.668
corel5k	104.8	115.676	142.99	114.792	107.412	<b>98.118</b>	101.93
tmc2007	1.311	4.572	2.248	3.349	<b>1.219</b>	1.397	1.235
mediamill	20.481	20.456	22.872	19.41	16.926	<b>16.265</b>	16.824
bibtex	20.926	58.6	30.359	38.224	25.854	<b>17.759</b>	20.87
delicious	530.126	691.622	577.478	548.007	504.999	<b>463.272</b>	488.441
bookmarks	DNF	73.78	47.692	50.456	34.185	<b>26.928</b>	34.122
ranking loss							
emotions	0.246	0.27	0.219	0.247	<b>0.151</b>	0.156	0.163
scene	<b>0.06</b>	0.174	0.149	0.202	0.072	0.071	0.079
yeast	<b>0.164</b>	0.2	0.228	0.196	0.167	0.166	0.165
medical	0.021	<b>0.104</b>	0.036	0.054	0.024	0.029	0.019
enron	0.084	0.114	0.099	0.094	0.079	<b>0.072</b>	0.077
corel5k	0.117	0.139	0.169	0.132	0.117	<b>0.109</b>	0.114
tmc2007	0.003	0.1	0.032	0.066	0.006	0.006	<b>0.001</b>
mediamill	0.061	0.063	0.075	0.058	0.047	<b>0.045</b>	0.047
bibtex	0.068	0.256	0.105	0.147	0.093	<b>0.058</b>	0.072
delicious	0.114	0.172	0.13	0.121	0.106	<b>0.095</b>	0.102
bookmarks	DNF	0.258	0.153	0.169	0.104	<b>0.081</b>	0.103
avg. precision							
emotions	0.721	0.713	0.753	0.731	<b>0.812</b>	0.805	0.805
scene	<b>0.893</b>	0.75	0.762	0.728	0.874	0.874	0.864
yeast	<b>0.768</b>	0.725	0.689	0.73	0.757	0.758	0.76

**Table 8** (continued)

	BR	MTP	HSC	HMC	RFMTP	RFHSC	RFHMC
medical	<b>0.896</b>	0.522	0.847	0.801	0.868	0.876	0.887
enron	0.693	0.547	0.635	0.642	0.698	<b>0.709</b>	0.7
corel5k	0.303	0.208	0.21	0.253	<b>0.334</b>	0.306	0.313
tmc2007	0.978	0.7	0.848	0.774	<b>0.996</b>	0.965	0.992
mediamill	0.686	0.654	0.632	0.684	<b>0.737</b>	0.734	0.735

## References

- Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1), 105–139.
- Blokeel, H., Raedt, L.D., & Ramon, J. (1998). Top-down induction of clustering trees. In *Proceedings of the 15th international conference on machine learning* (pp. 55–63).
- Boutell, M.R., Luo, J., Shen, X., & Brown, C.M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9), 1757–1771.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- Breiman, L., Friedman, J., Olshen, R., & Stone, C.J. (1984). Classification and regression trees. Chapman & Hall/CRC.
- Brinker, K., Fürnkranz, J., & Hüllermeier, E. (2006). A unified model for multilabel classification and ranking. In *Proceedings of the 2006 conference on ECAI 2006: 17th european conference on artificial intelligence August 29 – September 1, 2006, Riva del Garda, Italy* (pp. 489–493).
- Chang, C.C., & Lin, C.J. (2001). LIBSVM: A library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Clare, A., & King, R.D. (2001). Knowledge discovery in multi-label phenotype data. In *Proceedings of the 5th european conference on PKDD* (pp. 42–53).
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
- Duygulu, P., Barnard, K., de Freitas, J., & Forsyth, D. (2002). Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th european conference on computer vision* (pp. 349–354).
- Elisseeff, A., & Weston, J. (2005). A kernel method for Multi-Labelled classification. In *Proceedings of the annual ACM conference on research and development in information retrieval* (pp. 274–281).
- Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11, 86–92.
- Gibaja, E., & Ventura, S. (2015). A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3), 52:1–52:38.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I.H. (2009). The weka data mining software: an update. *SIGKDD Explorations*, 11, 10–18.
- Katakis, I., Tsoumakas, G., & Vlahavas, I. (2008). Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD discovery challenge* (pp. 124–135).
- Klimt, B., & Yang, Y. (2004). The enron corpus: a new dataset for email classification research. In *Proceedings of the 15th european conference on machine learning* (pp. 217–226).
- Kocev, D. (2011). Ensembles for predicting structured outputs. Ph.D. thesis, IPS Jožef Stefan, Ljubljana, Slovenia.
- Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2007). Ensembles of multi-objective decision trees. In *Proceedings of the 18th european conference on machine learning* (pp. 624–631).
- Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2013). Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3), 817–833.

- Kong, X., & Yu, P.S. (2011). An ensemble-based approach to fast classification of multilabel data streams. In *Proceedings of the 7th international conference on collaborative computing: Networking, Applications and Worksharing* (pp. 95–104).
- Levatić, J., Kocev, D., & Džeroski, S. (2014). The importance of the label hierarchy in hierarchical multi-label classification. *Journal of Intelligent Information Systems*, 45(2), 247–271.
- Li, P., Li, H., & Wu, M. (2013). Multi-label ensemble based on variable pairwise constraint projection. *Information Sciences*, 222(0), 269–281.
- Madjarov, G., Dimitrovski, I., Gjorgjevikj, D., & Deroski, S. (2015). Evaluation of different data-derived label hierarchies in multi-label classification. In *New frontiers in mining complex patterns, lecture notes in computer science*, (Vol. 8983 pp. 19–37): Springer international publishing.
- Madjarov, G., Kocev, D., Gjorgjevikj, D., & Dzeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9), 3084–3104.
- Nemenyi, P.B. (1963). Distribution-free multiple comparisons. Ph.D. thesis, Princeton University.
- Quinlan, J.R. (1993). C4.5: Programs for machine learning (Morgan Kaufmann series in machine learning) morgan kaufmann.
- Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classifier chains for multi-label classification. In *Proceedings of the 20th european conference on machine learning* (pp. 254–269).
- Silla Carlos, N.J., & Freitas, A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22, 31–72.
- Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., & Smeulders, A.W.M. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th annual ACM international conference on multimedia* (pp. 421–430).
- Srivastava, A., & Zane-Ulman, B. (2005). Discovering recurring anomalies in text reports regarding complex space systems. In *Proceedings of the IEEE aerospace conference* (pp. 55–63).
- Trohidis, K., Tsoumakas, G., Kalliris, G., & Vlahavas, I. (2008). Multilabel classification of music into emotions. In *Proceedings of the 9th international conference on music information retrieval* (pp. 320–330).
- Tsoumakas, G., & Katakis, I. (2007). Multi label classification: an overview. *International Journal of Data Warehouse and Mining*, 3(3), 1–13.
- Tsoumakas, G., Katakis, I., & Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings of the ECML/PKDD workshop on mining multidimensional data* (pp. 30–44).
- Tsoumakas, G., & Vlahavas, I. (2007). Random k-labelsets: an ensemble method for multilabel classification. In *Proceedings of the 18th european conference on machine learning* (pp. 406–417).
- Vens, C., Struyf, J., Schietgat, L., Džeroski, S., & Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2), 185–214.
- Zhang, M.L., & Zhou, Z.H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8), 1819–1837.