J Intell Inf Syst
DOI 10.1007/s10844-014-0347-y

The importance of the label hierarchy in hierarchical multi-label classification

Jurica Levatić · Dragi Kocev · Sašo Džeroski

Received: 9 June 2014 / Revised: 14 November 2014 / Accepted: 17 November 2014

Abstract We address the task of hierarchical multi-label classification (HMC). HMC is a task of structured output prediction where the classes are organized into a hierarchy and an instance may belong to multiple classes. In many problems, such as gene function prediction or prediction of ecological community structure, classes inherently follow these constraints. The potential for application of HMC was recognized by many researchers and several such methods were proposed and demonstrated to achieve good predictive performances in the past. However, there is no clear understanding when is favorable to consider such relationships (hierarchical and multi-label) among classes, and when this presents unnecessary burden for classification methods. To this end, we perform a detailed comparative study over 8 datasets that have HMC properties. We investigate two important influences in HMC: the multiple labels per example and the information about the hierarchy. More specifically, we consider four machine learning tasks: multi-label classification, hierarchical multi-label classification, single-label classification and hierarchical single-label classification. To construct the predictive models, we use predictive clustering trees (a generalized form of decision trees), which are able to tackle each of the modelling tasks listed. Moreover, we investigate whether the influence of the hierarchy and the multiple labels carries over for ensemble models. For each of the tasks, we construct a single tree and two ensembles (random forest and bagging). The results reveal that the hierarchy and the multiple labels do help to obtain a better single tree model, while this is not preserved for the ensemble models.

J. Levatić (✉) · D. Kocev · S. Džeroski
Department of Knowledge Technologies, Jožef Stefan Institute, 1000 Ljubljana, Slovenia
e-mail: jurica.levatic@ijs.si

D. Kocev
e-mail: dragi.kocev@ijs.si

J. Levatić · S. Džeroski
Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
e-mail: saso.dzeroski@ijs.si

Published online: 04 December 2014

Keywords Predictive clustering trees · Ensemble methods · Hierarchical multi-label classification · Habitat modelling · Text classification · Image classification · Gene function prediction

1 Introduction

Supervised learning is one of the most widely researched and investigated areas of machine learning. The goal in supervised learning is to learn, from a set of examples with known class, a function that outputs a prediction for the class of a previously unseen example. The most widely studied machine learning task is binary classification where the goal is to classify the examples into two groups. The task where the examples can belong to a single class from a given set of m classes ($m \geq 3$) is known as multi-class classification. The case where the output is a real value is called regression.

However, in many real life problems of predictive modelling the output (i.e., the target) is structured, meaning that there can be dependencies between classes (e.g., classes are organized into a tree-shaped hierarchy or a directed acyclic graph) or some internal relations between the classes (e.g., sequences). These types of problems occur very often in various domains, such as life sciences (predicting gene function, finding the most important genes for a given disease, predicting toxicity of molecules, etc.), ecology (analysis of remotely sensed data, habitat modelling), multimedia (annotation and retrieval of images and videos) and the semantic web (categorization and analysis of text and web pages). Having in mind the needs of these application domains and the increasing quantities of structured data, Kriegel et al. (2007) and Dietterich et al. (2008) listed the task of "mining complex knowledge from complex data" as one of the most challenging problems in machine learning.

A variety of methods, specialized in predicting a given type of structured output (e.g., a hierarchy of classes (Silla and Freitas 2011)), have been proposed (Bakır et al. 2007). These methods can be categorized into two groups of methods for solving the problem of predicting structured outputs (Silla and Freitas 2011; Bakır et al. 2007). Local methods construct models for predicting component(s) of the output and then combine the individual models to get the overall model (i.e., they construct an architecture of several simple(r) models). Global methods that construct models for predicting the complete structure as a whole (also known as 'big-bang' approaches).

The global methods have several advantages over the local methods. First, they exploit and use the dependencies that may exist between the components of the structured output in the model learning phase, which can result in better predictive performance of the learned models. Next, they are typically more efficient: it can easily happen that the number of components in the output is very large (e.g., hierarchies in functional genomics can have several thousands of components), in which case learning a model for each component is not feasible. Furthermore, they produce models that are typically smaller than the sum of the sizes of the models built for each of the components.

Despite the many developed methods and their interesting applications, it is not clear when it is favorable (performance wise) to apply global and when local approaches. In this work, we focus on clarifying this important issue for the task of hierarchical multi-label classification (HMC). HMC is a variant of classification, where a single example may belong to multiple classes at the same time and the classes are organized in the form of a hierarchy. An example that belongs to some class c automatically belongs to all super-classes of c: This is called the hierarchical constraint. Problems of this kind can be found in many

domains including text classification, functional genomics, and object/scene classification. Silla and Freitas (2011) give a detailed overview of the possible application areas and the different approaches to HMC.

More specifically, we construct four types of predictive models that exploit different amounts of the information provided by the output structure, i.e., the hierarchical organization of the classes. This corresponds to four different machine learning tasks as depicted in Fig. 1: binary classification, hierarchical single-label classification, multi-label classification and hierarchical multi-label classification. The first two tasks construct (an architecture of) local predictive models, while the last two tasks construct global models.

Orthogonally to the issue of using different sources of information about the output structure, we also investigate the influence of constructing ensembles in such a setting. Ensembles are a set of (base) predictive models that can be local or global. It is widely accepted that, for basic machine learning tasks (regression and classification), ensembles improve performance of its base models (Seni and Elder 2010). Kocev et al. (2013) recently have showed that the same holds for tree ensembles for predicting structured outputs in the case of hierarchical single-label classification and hierarchical multi-label classification.

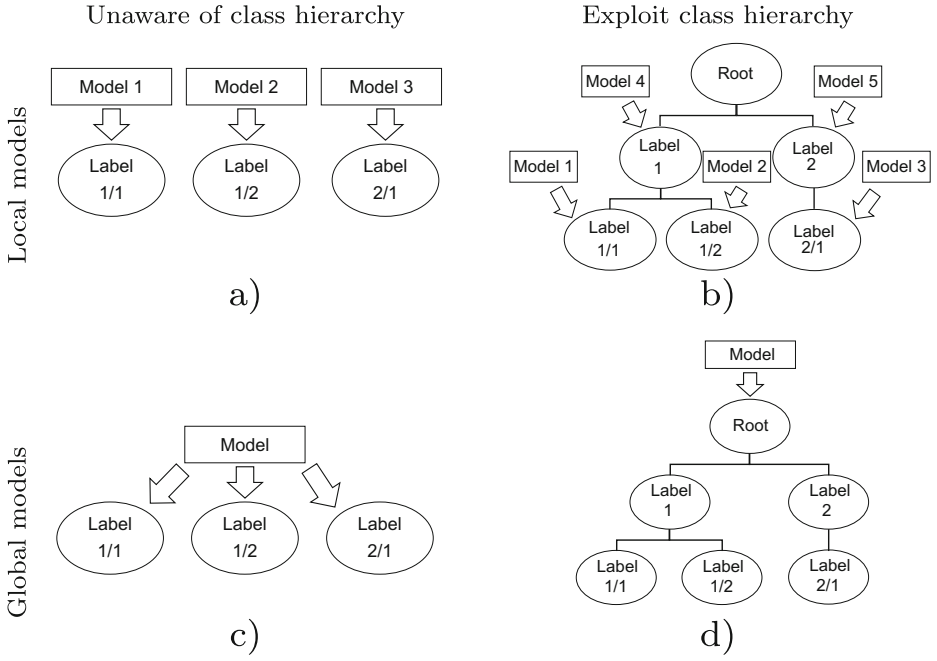

Fig. 1 Schematic representation of the four different modelling tasks we consider to investigate how exploitation of label hierarchy affects the performance. Single label classification **a**, builds a separate model for each of the leaf labels, while hierarchical single label classification **b**, builds a separate model for each edge of the label hierarchy (each model is trained by using only data that is relevant to that edge). Multi-label classification **c** and hierarchical multi-label classification **d** build one *(global)* model which considers all of the classes at once: the former approach **(c)** is unaware of the taxonomic hierarchy, while the latter approach **(d)** exploits information about the label hierarchy. For each of the four modelling task we build three types of models *(depicted as rectangles)*: single tree models, random forest ensemble and bagging ensemble. The different kinds of output of the models are given at the pointed ends of the arrows

The focus of the research in Kocev et al. (2013) was on the performance increase of the ensemble models over their respective base predictive models. The issue of the interplay between the different information about the output structure and the influence on the ensembles' predictive performance has not received much attention.

In this work, we are focused on three important questions for the task of predicting structured outputs, i.e., hierarchical multi-label classification. Firstly, we research the use of different information about the output structure in the context of single models' predictive performance. Secondly, we research the same problem in the context of ensemble models'. Finally, we investigate whether the conclusions from the investigation on single models carry over to the ensemble models. Moreover, we discuss whether it is more beneficial to use the structure of the output space or construct ensembles ignoring the output structure.

To properly evaluate the predictive performance of the different models one needs to select predictive models from the same type that can solve the four tasks enumerated above. To this end, we consider predictive clustering trees (PCTs) as predictive models. PCTs can be viewed as a generalization of standard decision trees towards predicting structured outputs. PCTs offer a unifying approach for dealing with different types of structured outputs and construct the predictive models very efficiently. They are able to make predictions for several types of structured outputs: tuples of continuous/discrete variables, hierarchies of classes, and time series (Kocev et al. 2013; Blockeel 1998; Vens et al. 2008). Furthermore, we construct and compare two types of ensembles of decision trees: random forest (Breiman 2001) and bagging (Breiman 1996).

We perform the evaluation of the predictive models on eight practically relevant HMC datasets. The datasets come from four different domains: habitat modelling, image classification, text classification and functional genomics. We consider habitat models for Collembola communities in the soils of Denmark (Demšar et al. 2006) and communities of organisms living in Slovenian rivers (Džeroski et al. 2000). Next, we use two datasets from the 2007 CLEF cross-language image retrieval campaign (Dimitrovski et al. 2008), where the goal is to annotate medical X-ray images. From the domain of text classification, we use two well known datasets: categorization of e-mails from officials of the Enron corporation (Klimt and Yang 2004) and categorization of Reuters newswire stories (Lewis et al. 2004). From the functional genomics domain, we use two datasets concerned with the task of gene function prediction for two model organisms: *Arabidopsis thaliana* and *Saccharomyces cerevisiae* (baker's or brewer's yeast) (Clare 2003).

The work presented in this paper builds upon our previous work given in Levatić et al. (2013, 2014). In Levatić et al. (2013), we investigated two datasets from the area of ecological modelling and considered only cross-validated performance estimates for comparison. In Levatić et al. (2014), we included four additional datasets from the area of image annotation and text categorization. Next, we included a comparison over the training performance (i.e., we calculated an overfit score). Furthermore, we compared the efficiency of the used methods to obtain the different models and the model sizes. We extend this work in three major directions. First, we included two other datasets from a new domain: functional genomics. Second, we included ensemble methods in the study (bagging and random forest). Finally, we have provided a more detailed analysis of the results. All in all, this study is qualitatively and quantitatively improved over the previous studies.

The remainder of this paper is organized as follows. Section 2 describes existing methods for HMC. Section 3 explains the predictive clustering trees framework and the extensions for the different tasks considered here. The experimental setup is presented in Section 4. Section 5 presents the obtained results. Finally, the conclusions are stated in Section 6.

2 Related work

In this section, we first briefly present an overview of existing methods which can deal with the task of hierarchical multi-label classification. More specifically, we discuss kernel-based, tree-based and other more application-specifically tailored methods for HMC. We then motivate the selection of predictive clustering trees as predictive models.

Several kernel based methods for the task of HMC were proposed. In these methods, SVMs are learned for each class separately and then combined so that the predictions are consistent with the hierarchical relationships (Obozinski et al. 2008; Barutcuoglu et al. 2006; Guan et al. 2008; Valentini 2011). Rousu et al. (2006) present a more direct method that does not require a second step to make sure that the hierarchy constraint is satisfied. Their method is based on a large margin method for structured output prediction which defines a joint feature map over the input and the output space. The kernel-based methods for HMC produce predictive models that are not interpretable.

Decision tree based methods take a notable place among approaches for HMC. Contrary to the previously described methods, decision trees are easily interpreted by domain experts. Clare and King (2003) adapted the well-know decision tree algorithm C4.5 (Quinlan 1993) for the task of HMC. This version of C4.5 (called C4.5H) uses the sum of entropies of the class variables to select the best split. C4.5H predicts classes on several levels of the hierarchy, assigning a larger cost to misclassification higher up in the hierarchy. Blockeel et al. (2002, 2006) proposed the idea of using predictive clustering trees (Blockeel 1998) for HMC tasks. The work of Blockeel et al. (2006) presents the first thorough empirical comparison between an HMC and HSC decision tree method in the context of tree shaped class hierarchies. Vens et al. (2008) extend the algorithm towards hierarchies structured as directed acyclic graph and show that learning one decision tree for predicting all classes simultaneously outperforms learning one tree per class (even if those trees are built by taking the hierarchy into account). Related to PCTs are distance-based decision trees (DBDT) (Estruch et al. 2006), where different distance metrics are associated to every attribute. This allows DBDT to handle structured descriptive attributes, such as sets, lists or trees (in PCTs different distance metrics are used to handle various types of outputs).

There are several other methods for HMC based on different approaches. Kiritchenko et al. (2006) performed hierarchical text categorization by expanding label sets of training examples to make them consistent with a given class hierarchy. Standard multi-class learning algorithm is then applied to modified multi-label data, followed by re-labeling of the inconsistently classified test instances. Silla and Freitas adapted the naïve Bayes approach for HMC (Silla and Freitas 2009). Otero et al. (2010) and Cerri et al. (2012) used search heuristics to discover HMC rules. In other work, Cerri et al. represented class hierarchy as a sequence of connected artificial neural networks, where the output of the one network is used as the input of the next network in the sequence (Cerri et al. 2014). Bi and Kwok (2012) proposed a hierarchically aware loss function, appropriate for both tree and DAG hierarchies and developed a Bayes-optimal classifier for HMC by using this loss function. Alaydie et al. (2012) proposed a boosting-based method for HMC, where at each iteration the label hierarchy is used to select the training set for each classifier. Recently, Barros et al. proposed a method for HMC based on the probabilistic clustering with expectation-maximization algorithm (Barros et al. 2013).

Finally, in this work, we investigate the effect of the label hierarchy and the multiple labels per example to the performance of the predictive models. We do this by the means of comparing the performance of the predictive models applied on four different modelling tasks: HMC, HSC, multi- and single-label classification. To properly evaluate the predictive

performance of the different models, one needs to select predictive models from the same type that can solve the four tasks enumerated above, i.e., to eliminate the chance of introducing bias on the results by the types of models. The selected method thus needs to be able to use label hierarchy and the multiple labels per examples for model construction and to be able to produce both local and global models. Such prerequisites considerably narrow the choice of possible methods. A framework that satisfies all of the conditions listed above is the PCT framework. Consequently, we selected to use the PCT framework throughout this study.

3 Predictive modelling for HMC

In this section, we present in more detail the methodology used to construct the predictive models. We first present the predictive clustering trees. Namely, we give the PCTs that predict the complete output (i.e., a single model for all of the possible labels in the dataset) with a single model. We then briefly describe local approaches that construct several models - each one predicting a part of the output (i.e., a model for each label separately). Finally, we describe tree ensembles for predicting structured outputs, both for global and local prediction of the structured output.

3.1 Predictive clustering trees for HMC

3.1.1 Global predictive clustering trees

The Predictive Clustering Trees (PCTs) framework views a decision tree as a hierarchy of clusters: the top-node corresponds to one cluster containing all data, which is recursively partitioned into smaller clusters while moving down the tree. The PCT framework is implemented in the CLUS system (Blockeel and Struyf 2002), which is available for download at http://clus.sourceforge.net.

PCTs are induced with a standard *top-down induction of decision trees* (TDIDT) algorithm (Breiman et al. 1984). The algorithm is presented in Table 1. It takes as input a set of examples (E) and outputs a tree. The heuristic (h) that is used for selecting the tests (t) is the reduction in variance caused by the partitioning ($\mathcal{P}$) of the instances corresponding

Table 1 The top-down induction algorithm for PCTs

procedure PCT
Input: A dataset E
Output: A predictive clustering tree
1: $(t^*, h^*, \mathcal{P}^*) = \text{BestTest}(E)$
2: **if** $t^* \neq none$ **then**
3: **for each** $E_i \in \mathcal{P}^*$ **do**
4: $tree_i = \text{PCT}(E_i)$
5: **return** node$(t^*, \bigcup_i \{tree_i\})$
6: **else**
7: **return** leaf(Prototype(E))

procedure BestTest
Input: A dataset E
Output: the best test (t^*), its heuristic score (h^*) and the partition ($\mathcal{P}^*$) it induces on the dataset (E)
1: $(t^*, h^*, \mathcal{P}^*) = (none, 0, \emptyset)$
2: **for each** possible test t **do**
3: $\mathcal{P} =$ partition induced by t on E
4: $h = Var(E) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var(E_i)$
5: **if** $(h > h^*) \wedge$ Acceptable$(t, \mathcal{P})$ **then**
6: $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
7: **return** $(t^*, h^*, \mathcal{P}^*)$

to the tests (t) (see line 4 of the BestTest procedure in Table 1). By maximizing the variance reduction, the cluster homogeneity is maximized and the predictive performance is improved.

The main difference between the algorithm for learning PCTs and a standard decision tree learner is that the former considers the variance function and the prototype function (that computes a label for each leaf) as *parameters* that can be instantiated for a given learning task. So far, PCTs have been instantiated for the following tasks: multi-target prediction (which includes multi-label classification) (Kocev et al. 2013), hierarchical multi-label classification (Vens et al. 2008) and prediction of time-series (Slavkov et al. 2010). In this article, we focus on the first two tasks.

PCTs for multi-label classification. PCTs for multi-label classification can be considered as PCTs that are able to predict multiple binary (and thus discrete) targets simultaneously. Therefore, the variance function for the PCTs for MLC is computed as the sum of the Gini indices of the target variables, i.e., $Var(E) = \sum_{i=1}^{T} Gini(E, Y_i)$. Alternatively, one can also use the sum of the entropies of class variables as a variance function, i.e., $Var(E) = \sum_{i=1}^{T} Entropy(E, Y_i)$ (this definition has also been used in the context of multi–label prediction (Clare 2003)). The CLUS system also implements other variance functions, such as reduced error, gain ratio and the m-estimate. The prototype function returns a vector of probabilities that an instance belongs to a given class for each target variable. Using these probabilities, the most probable (majority) class value for each target can be calculated.

PCTs for hierarchical multi–label classification. CLUS-HMC is the instantiation (with the distances and prototypes as defined below) of the PCT algorithm for hierarchical classification implemented in the CLUS system (Vens et al. 2008). The variance and prototype are defined as follows. First, the set of labels of each example is represented as a vector with binary components; the i^{th} component of the vector is 1 if the example belongs to class c_i and 0 otherwise. It is easily checked that the arithmetic mean of a set of such vectors contains as i^{th} component the proportion of examples of the set belonging to class c_i. The variance of a set of examples E is defined as the average squared distance between each example's class vector (L_i) and the set's mean class vector ($\overline{L}$), i.e.,

$$Var(E) = \frac{1}{|E|} \cdot \sum_{E_i \in E} d(L_i, \overline{L})^2.$$

In the HMC context, the similarity at higher levels of the hierarchy is more important than the similarity at lower levels. This is reflected in the distance measure used in the above formula, which is a weighted Euclidean distance:

$$d(L_1, L_2) = \sqrt{\sum_{l=1}^{|L|} w(c_l) \cdot (L_{1,l} - L_{2,l})^2},$$

where $L_{i,l}$ is the l^{th} component of the class vector L_i of an instance E_i, $|L|$ is the size of the class vector, and the class weights $w(c)$ decrease with the depth of the class in the hierarchy. More precisely, $w(c) = w_0 \cdot w(p(c))$, where $p(c)$ denotes the parent of class c and $0 < w_0 < 1$).

For example, consider the toy class hierarchy shown in Fig. 2a,b, and two data examples: (X_1, S_1) and (X_2, S_2) that belong to the classes $S_1 = \{c_1, c_2, c_{2.2}\}$ (boldface in Fig. 2b) and $S_2 = \{c_2\}$, respectively. We use a vector representation with consecutive components

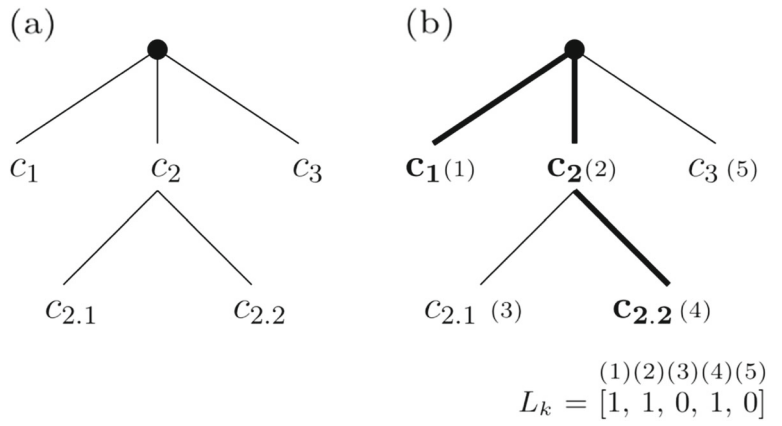

Fig. 2 Toy examples of a hierarchy structured as a tree. **a** Class label names contain information about the position in the hierarchy, e.g., $c_{2.1}$ is a subclass of c_2. **b** The set of classes $S_1 = \{c_1, c_2, c_{2.2}\}$, shown in bold, are represented as a vector (L_k)

representing membership in the classes c_1, c_2, $c_{2.1}$, $c_{2.2}$ and c_3, in that order (preorder traversal of the tree of class labels). The distance is then calculated as follows:

$$d(S_1, S_2) = d([1, 1, 0, 1, 0], [0, 1, 0, 0, 0]) = \sqrt{w_0 + w_0^2}.$$

Recall that the instantiation of PCTs for a given task requires a proper instantiation of the variance and prototype functions. The variance function for the HMC task is instantiated by using the weighted Euclidean distance measure (as given above), which is further used to select the best test for a given node by calculating the heuristic score (line 4 from the algorithm in Table 1). We now discuss the instantiation of the prototype function for the HMC task.

A classification tree stores in a leaf the majority class for that leaf, which will be the tree's prediction for all examples that will arrive in the leaf. In the case of HMC, an example may have multiple classes, thus the notion of *majority class* does not apply in a straightforward manner. Instead, the mean $\bar{L}$ of the class vectors of the examples in the leaf is stored as a prediction. Note that the value for the i^{th} component of $\bar{L}$ can be interpreted as the probability that an example arriving at the given leaf belongs to class c_i.

The prediction for an example that arrives at the leaf can be obtained by applying a user defined threshold τ to the probability; if the i^{th} component of $\bar{L}$ is above τ then the examples belong to class c_i. When a PCT is making a prediction, it preserves the hierarchy constraint (the predictions comply with the parent-child relationships from the hierarchy) if the values for the thresholds τ are chosen as follows: $\tau_i \leq \tau_j$ whenever $c_i \leq_h c_j$ (c_i is ancestor of c_j). The threshold τ is selected depending on the context. The user may set the threshold such that the resulting classifier has high precision at the cost of lower recall or vice versa, to maximize the F-score, to maximize the interpretability or plausibility of the resulting model etc. In this work, we use a threshold-independent measure (precision-recall curves) to evaluate the performance of the models.

3.1.2 Local predictive clustering trees

Local models for predicting structured outputs use a collection of predictive models, each predicting a component of the overall structure that needs to be predicted. For the task

of predicting multiple targets, local predictive models are constructed by learning a predictive model for each of the targets separately. In the task of hierarchical multi-label classification, however, there are four different approaches that can be used: flat classification, local classifiers per level, local classifiers per parent node, and local classifiers per node.

We briefly describe these approaches, for more details see (Silla and Freitas 2011; Vens et al. 2008). In flat classification, a separate local classifier is learnt for each node of the class hierarchy, where examples labeled with label corresponding to that node are considered as positive and all the others as negative. In other local approaches (local classifiers per level, per parent node, and per node), hierarchical relationships among classes are taken into account by virtue of training separate local classifiers only with the subset of examples which are labeled with a specific part of the class hierarchy. More specifically, the local classifier per level approach consists of training one multi-class classifier for each level of the class hierarchy to differentiate between nodes at each level of class hierarchy. Next, the local classifier per parent node approach builds a multi-class classifier for each parent node in the class hierarchy to distinguish between its child nodes. Finally, the local classifier per node approach consists of training one binary classifier for each node of the class hierarchy.

Vens et al. (2008) investigated the performance of the last two approaches with local classifiers over a large collection of 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 shorter induction times.

In particular, the CLUS-HSC algorithm by Vens et al. (2008) constructs a decision tree classifier for each edge (connecting a class c with a parent class $par(c)$) in the hierarchy, thus creating an architecture of classifiers. The tree that predicts membership to class c is learnt using the instances that belong to $par(c)$. The construction of this type of trees uses few instances, as only instances labeled with $par(c)$ are used for training. The instances labeled with class c are positive while the ones labeled with $par(c)$, but not with c are negative.

The resulting HSC tree architecture predicts the conditional probability $P(c|par(c))$. A new instance is predicted by recursive application of the product rule $P(c) = P(c|par(c)) \cdot P(par(c))$, starting from the tree for the top-level class. Again, the probabilities are thresholded to obtain the set of predicted classes. To satisfy the hierarchy constraint, the threshold τ should be chosen as in the case of CLUS-HMC.

In this work, we also consider the task of single-label classification. We consider this to be a special case of multi-label classification where the number of labels is 1. To this end, we use the same algorithm as for the multi-label classification trees. We call these models single-label classification trees.

3.2 Ensembles of predictive clustering trees for HMC

We consider ensembles of PCTs for structured prediction, as implemented by Kocev et al. (2013) in the CLUS system. The PCTs in the ensembles are constructed by using the bagging (Breiman 1996) and random forests (Breiman 2001) methods that are often used in the context of decision trees. The algorithms of these ensemble learning methods are presented in Table 2. Bagging (Table 2, left) is an ensemble method that constructs the different classifiers by making bootstrap replicates of the training set and using each of these replicates to construct a predictive model. Each bootstrap sample is obtained by randomly sampling training instances, with replacement, from the original training set, until an equal number

Table 2 The ensemble learning algorithms: bagging and random forests

procedure Bagging(E, k) **returns** Forest	**procedure** RForest$(E, k, f(D))$ **returns** Forest
1: $F = \emptyset$	1: $F = \emptyset$
2: **for** $i = 1$ **to** k **do**	2: **for** $i = 1$ **to** k **do**
3: $E_i = bootstrap(E)$	3: $E_i = bootstrap(E)$
4: $T_i = PCT(E_i)$	4: $T_i = PCT\_rnd(E_i, f(D))$
5: $F = F \bigcup \{T_i\}$	5: $F = F \bigcup \{T_i\}$
6: **return** F	6: **return** F

Here, E is the set of the training examples, k is the number of trees in the forest, and $f(D)$ is the size of the feature subset considered at each node during tree construction for random forests.

of instances as in the training set is obtained. Breiman (1996) showed that bagging can give substantial gains in predictive performance, when applied to an unstable learner (i.e., a learner for which small changes in the training set result in large changes in the predictions), such as classification and regression tree learners.

A random forest (Table 2, right) is an ensemble of trees, where diversity among the predictors is obtained by using bootstrap replicates as in bagging, and additionally by changing the set of descriptive attributes during learning. To learn a random forest, the PCT algorithm for tree construction (Table 1) is changed to PCT_rnd: randomized version of the selection of attributes, which replaces the standard selection of attributes. More precisely, at each node in the decision trees, a random subset of the descriptive attributes is taken, and the best attribute is selected from this subset. The number of attributes that are retained is given by a function f of the total number of descriptive attributes D (e.g., $f(D) = 1$, $f(D) = \lfloor \sqrt{D} + 1 \rfloor$, $f(D) = \lfloor log_2(D) + 1 \rfloor$ ...). By setting $f(D) = D$, we obtain the bagging procedure.

To construct global and local ensemble models, corresponding type of PCTs are used as a base model, i.e., to construct global ensemble for the HMC task, PCTs for hierarchical multi-label classification are used as a base model. Note that, for the HSC task, ensembles can be constructed in two ways: an ensemble of architectures or an architecture of ensembles. The first approach creates the ensemble by creating multiple architectures. These multiple architectures can be created on different bootstrap replicates, on different feature spaces, by different local classifiers etc. The second approach is simpler and, instead of a single local classifier, uses an ensemble as a classifier at each branch of class hierarchy. The HSC ensembles in this work are constructed by following the second approach, since it is closer to the learning of local classifiers for predicting multiple target variables (separate single-label model for each of the targets).

The prediction of an ensemble for a new instance is obtained by combining the predictions of all the base predictive models from the ensemble. For classification tasks, different aggregation schemes can be applied, such as majority of probability distribution voting. We used probability distribution voting, as suggested by Bauer and Kohavi (1999). For global models, per target probability distribution voting was used.

4 Experimental design

In this section, we present the design of the experimental evaluation of the predictive models built for the four machine learning tasks considered. We begin by describing the data used. We then outline the specific experimental setup for constructing the predictive

models. Finally, we present the evaluation measure for assessing the predictive performance of the models.

4.1 Data description

We use eight datasets, which come from four domains: habitat modeling, image classification, text classification and functional genomics. The main statistics of the datasets are given in Table 3. We can observe that the datasets vary in the size, number of attributes and characteristics of the label hierarchy.

Habitat modelling (Džeroski 2009) focuses on spatial aspects of the distribution and abundance of plants and animals. It studies the relationships between environmental variables and the presence/abundance of plants and animals. This is typically done under the implicit assumption that both are observed at a single point in time for a given spatial unit (i.e., sampling site). We investigate the effect of environmental conditions on communities of organisms in two different ecosystems, i.e., river and soil ecosystems. Namely, we construct habitat models for water organisms living in Slovenian rivers (Džeroski et al. 2000) and for soil microarthropods from Danish farms (Demšar et al. 2006). The data about the organisms that live in the water of Slovenian rivers was collected during six years (1990 to 1995) of monitoring of water quality performed by the Hydro-meteorological Institute of Slovenia (now Environmental Agency of Slovenia). The data for the soil microarthropods from Danish farms describes four experimental farming systems (observed during the period 1989-1993) and a number of organic farms (observed during the period 2002-2003). The structured output space in these case studies is the taxonomic hierarchy of the species. Since different species are considered in the two domains, their respective output spaces will be different.

In image classification, the goal is to automatically annotate images with labels. The labels typically represent visual concepts that are present in the images. In this work, we are concerned with the annotation of medical X-ray images. We use two datasets from the 2007 CLEF cross-language image retrieval campaign (Dimitrovski et al. 2008): ImCLEF07A and ImCLEF07D. The goal in these datasets is to recognize which part of the human anatomy is present in the image and the orientation of the body part, respectively. Images are represented by using edge histograms. An edge histogram represents the frequency and the directionality of the brightness changes in the image. The structured output space consists

Table 3 Characteristics of the datasets: N is the number of instances, D/C is the number of descriptive attributes (discrete/continuous), $\mathcal{L}$ is the number of labels (leafs in the hierarchy), $|\mathcal{H}|$ is the number of nodes in the hierarchy, $\mathcal{H}_d$ is the maximal depth of the hierarchy, $\overline{\mathcal{L}_L}$ is the average number of labels per example

Domain	N	D/C	$\mathcal{L}$	$\lvert\mathcal{H}\rvert$	$\mathcal{H}_d$	$\overline{\mathcal{L}_L}$
Slovenian rivers (Džeroski et al. 2000)	1060	0/16	491	724	4	25
Danish farms (Demšar et al. 2006)	1944	132/5	35	72	3	7
ImCLEF07A (Dimitrovski et al. 2008)	11006	0/80	63	96	3	1
ImCLEF07D (Dimitrovski et al. 2008)	11006	0/80	26	46	3	1
Enron (Klimt and Yang 2004)	1648	0/1001	50	54	3	2.84
Reuters (Lewis et al. 2004)	6000	0/47236	77	100	4	1.2
SeqAra-FunCat (Clare 2003)	3718	2/4448	148	196	4	0.94
ExprYeast-FunCat (Clare 2003)	3783	4/547	161	417	4	2.28

of labels organized in a hierarchy. They correspond to the anatomical (ImCLEF07A) and directional (ImCLEF07D) axis of the IRMA (Image Retrieval in Medical Applications) code (Lehmann et al. 2003).

Text classification is the problem of automatic annotation of textual documents with one or more categories. We used two datasets from this domain: Enron and Reuters. Enron is a labeled subset of the Enron corpus (Klimt and Yang 2004), prepared and annotated by the UCBerkeley Enron Email Analysis Project.[1] The e-mails are categorized into several hierarchically organized categories concerning the characteristics of the e-mail, such as genre, emotional tone or topic. Reuters is a subset of the 'Topics' category of the Reuters Corpus Volume I (RCV1) (Lewis et al. 2004). RCV1 is a collection of English language stories published by the Reuters agency between August 20, 1996, and August 19, 1997. Stories are categorized into hierarchical groups according to the major subjects of a story, such as Economics, Industrial or Government. In both domains, the text documents are described with their respective bag-of-words representation.

In functional genomics, various data (e.g., DNA microarray measurements) are used to describe gene and protein functions. Machine learning methods are valuable tools for predicting gene functions, taken from a predefined set of functions (Schietgat et al. 2010). Predicted functions with highest confidence can be used to guide lab experiments and reduce the number of needed tests. We used two datasets concerned with two important model organisms: the SeqAra-FunCat dataset (Clare 2003) is concerned with gene functions for the plant *Arabidopsis thaliana*, whereas ExprYeast-FunCat dataset (Clare 2003) is concerned with gene functions for *S. cerevisiae* or baker's yeast. Descriptive attributes for the former dataset consist of features calculated from amino acid sequences, such as amino acid ratios, molecular weight and sequence length. Descriptive attributes for the latter dataset consist of microarray gene expression levels measured under various experimental conditions, such as heat shock or nitrogen depletion. Gene functions for both datasets come from FunCat catalogue of gene functions (Ruepp et al. 2004), which has a tree-shaped hierarchy.

4.2 Experimental design

We constructed predictive models corresponding to four types of modelling tasks, as described in the previous section: single-label classification trees (separate model for each leaf in the label hierarchy), hierarchical single-label classification (architecture of models), multi-label classification (one model for all of the leaf labels, without using the hierarchy) and hierarchical multi-label classification (one model for all of the labels by using the hierarchy). For each modelling task, we constructed single tree model(s), random forest tree ensembles and ensembles of bagged classification trees. In total, twelve predictive models for each of the case studies were built.

For single predictive clustering trees, we used F-test pruning to ensure that the produced models are not overfitted and have better predictive performance (Vens et al. 2008). The exact Fisher test is used to check whether a given split/test in an internal node of the tree results in a statistically significant reduction in variance. If there is no such split/test, the node is converted to a leaf. A significance level is selected from the values 0.125, 0.1, 0.05, 0.01, 0.005 and 0.001 to optimize predictive performance by using internal 3-fold cross validation.

[1] http://bailando.sims.berkeley.edu/enron_email.html

Ensemble models (random forests and bagging) were constructed with 100 trees. Trees were not pruned and the number of random features used in random forest was set to $\lfloor \log_2(D) + 1 \rfloor$, where D is the total number of features, as recommended by Breiman (2001).

We evaluate the predictive performance of the models on the classes/labels that are leafs in the target hierarchy. We made this choice in order to ensure a fair comparison across the different tasks. Namely, if we consider all labels (the leaf labels and the inner node labels), the single-label classification task will be very close to the task of hierarchical single-label classification; similarly, the task of multi-label classification becomes very close to the task of hierarchical multi-label classification. Moreover, by evaluating only the performance on leaf labels, we are measuring more precisely the influence of the inclusion of the different kinds of information in the learning process on the predictive performance of the models. To further ensure this, we set the w_0 parameter for the weighted Euclidean distance for HMC to the value of 1: all labels in the hierarchy contribute equally. By doing this, we measure only the effect of including the multi-label information (considering the multiple labels simultaneously) and the hierarchy information.

4.3 Evaluation measures

We evaluate the algorithms by using as performance measure the area under the Precision-Recall curve (AUPRC), and in particular, the area under the average Precision-Recall curve (AU$\overline{\text{PRC}}$) as suggested by Vens et al. (2008). The points in the Precision-Recall (PR) space are obtained by changing the value of the threshold τ from 0 to 1 with step 0.02. For each value of the threshold τ, precision and recall values are micro-averaged as follows: $\overline{Prec} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i}$, and $\overline{Rec} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i}$, where i ranges over all classes that are leafs in the output hierarchies.

AUPRC is a general and a threshold independent performance measure, closely related to it is the area under the receiver operating characteristic curve (AUROC). However, AUROC rewards predictive models for correctly predicting negative examples, which can give an overly optimistic estimate of the performance when there is a large skew in the class distribution (i.e., the number of positive and negative examples is imbalanced) (Davis and Goadrich 2006). Since this is the case in the datasets considered here, we have chosen to evaluate the studied methods by using the AUPRC measure.

We measure the performance of the predictive models along several dimensions. First, we estimate the predictive performance of the models by using 10-fold cross-validation.. Second, we assess the descriptive power of the models by evaluating them on the training set. Next, we measure how much the different models tend to overfit the training data. To this end, we use the relative decrease of performance from the one on the training set to the one obtained with 10-fold cross-validation. We define the overfit score as:

$$OS = \frac{\text{AU}\overline{\text{PRC}}_{train} - \text{AU}\overline{\text{PRC}}_{test}}{\text{AU}\overline{\text{PRC}}_{train}}.$$

Smaller values of this score mean less overfitting. Finally, we measure the complexity of the predictive models and the time efficiency of learning them. The model complexity for the global models is the number of nodes in a given tree, while the model complexity for the local models is the number of all nodes from all trees. Similarly, the running time of the global models is the time needed to construct the model, while the running time for the local models is the time needed to construct all of the models.

For the statistical evaluation of the results, we employed the corrected Friedman test and the post hoc Nemenyi test as recommended by Demšar (2006). The Friedman test is a non-parametric test for multiple hypotheses testing. It ranks the algorithms according to their performance, thus the best performing algorithm gets the rank of 1, second best the rank of 2 etc., and in case of ties it assigns average ranks. Then, the Friedman test compares the average ranks of the algorithms and calculates the Friedman statistic χ_F^2, distributed according to the χ_F^2 distribution with $k-1$ degrees of freedom (k being the number of algorithms). If there is a statistically significant difference in the performance, than we can proceed with a post hoc test. The Nemenyi post-hoc test is used to compare all the classifiers to each other. In this procedure, the performance of two classifiers is significantly different if their average ranks differ more than some critical distance. The critical distance depends on the number of algorithms, number of datasets and critical value (for a given significance level) that is based on the Studentized range statistic and can be found in statistical textbooks. We present the result from the Nemenyi post hoc test with an average ranks diagram as suggested by Demšar (2006). The ranks are depicted on the axis, in such a manner that the best ranking algorithms are at the right-most side of the diagram. The algorithms that do not differ significantly (in performance) for a significance level of 0.05 are connected with a line.

We test statistical significance of the differences in performance (1) for each dataset separately and (2) across all datasets. For the per-dataset comparison (Figs. 4, 6 and 7), we perform a Friedman test for each dataset on the folds of 10-fold cross validation (i.e., the methods are ranked by their per-fold performance). For the overall comparison (Figs. 3 and 5), we perform Friedman test by considering all eight datasets at once (i.e., the methods are ranked by their cross validation performance).

5 Results and discussion

In this section, we present the results from the experimental evaluation. We discuss the obtained models first in terms of their predictive performance and efficiency, and then in terms of their interpretability.

5.1 Performance of single tree models

The results from the evaluation of the single tree predictive models are given in Table 4. A quick inspection of the performance reveals that the best results are obtained by models that exploit the information about the underlying output hierarchy. Next, the models that include the hierarchy information tend to overfit less as compared to the other models. Moreover,

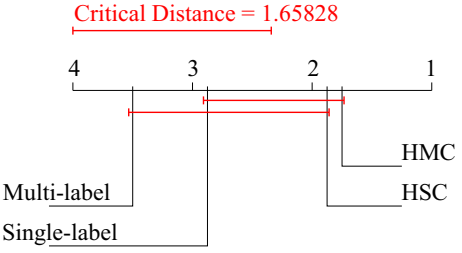

Fig. 3 Average ranks diagram for the performance of all single tree models in terms of AUPRC across all of the datasets. Better algorithms are positioned on the right-hand side, the ones that differ by less than the critical distance for a *p-value* = 0.05 are connected with a line

Table 4 Performance of single tree models in terms of AU$\overline{\text{PRC}}$, relative decrease between training set performance and test set performance (overfit score - OS), Learning time (LT, in seconds) and model complexity (the number of nodes in the decision trees)

Dataset	Method	AU$\overline{\text{PRC}}$	OS	LT	Complexity
	Single-label	0.239	0.692	23.3	15336
Slovenian	HSC	0.309	0.591	10.2	25035
rivers	Multi-label	0.322	0.007	9.4	1
	HMC	**0.374**	0.132	0.6	37
	Single-label	0.790	0.099	3.7	2605
Danish	HSC	0.808	0.083	1.3	2873
farms	Multi-label	0.801	0.112	0.7	265
	HMC	**0.815**	0.065	0.4	259
	Single-label	0.571	0.375	74.4	3957
ImCLEF07A	HSC	**0.665**	0.324	27.3	10054
	Multi-label	0.530	0.462	13.5	3553
	HMC	0.592	0.182	3.4	635
	Single-label	0.515	0.483	35.4	7418
ImCLEF07D	HSC	**0.631**	0.361	20.1	9764
	Multi-label	0.511	0.484	7.8	3675
	HMC	0.615	0.198	3.0	685
	Single-label	0.398	0.495	114.7	1740
Enron	HSC	0.466	0.434	25.1	3168
	Multi-label	0.385	0.584	13.8	1259
	HMC	**0.488**	0.110	3.3	55
	Single-label	0.431	0.546	970.8	3591
Reuters	HSC	**0.481**	0.510	781.4	7004
	Multi-label	0.332	0.654	191.8	2949
	HMC	0.373	0.365	42.5	593
	Single-label	0.152	0.837	2159.9	2678
SeqAra-FunCat	HSC	**0.158**	0.836	1999.7	6558
	Multi-label	0.129	0.864	3030.0	1839
	HMC	0.143	0.511	213.7	157
	Single-label	0.154	0.589	4045.9	1727
ExprYeast-FunCat	HSC	0.120	0.868	421.4	14323
	Multi-label	0.148	0.766	2122.8	2013
	HMC	**0.167**	0.187	50.6	65

The best predictive performance for each dataset is shown in bold.

the results indicate that the HMC trees overfit the least on these datasets. Finally, the global models (especially HMC) are more efficient than their local counterparts, in terms of both running time and model complexity.

We further examine the results by performing a statistical significance test. In particular, we performed the Friedman test to check whether the observed differences in performance among the studied methods are statistically significant, when taken over all datasets. We present the result of this test in Fig. 3. It reveals that the HMC trees are overall the best performing method and their performance is statistically significantly better than the multi-label classification trees. We can also see that methods that exploit the hierarchy information perform better than the ones that do not exploit this information.

We further discuss the results in terms of statistical tests for each dataset separately. Figure 4 presents the average ranks from the Nemenyi post-hoc test for single tree models. The diagrams show that the HMC models are best performing on four domains (Slovenian

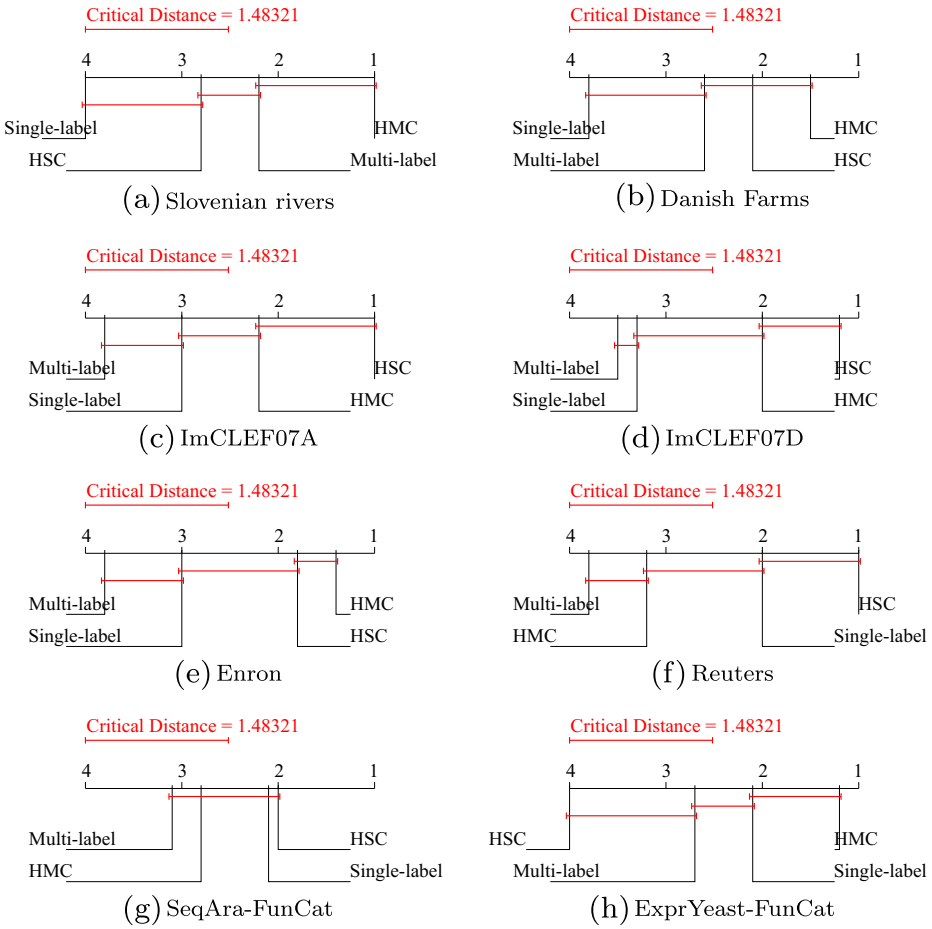

Fig. 4 Average ranks diagrams for the performance of the single tree PCTs in terms of $\overline{\text{AUPRC}}$ for each of the eight datasets. Better algorithms are positioned on the right-hand side, the ones that differ by less than the critical distance for a *p-value* $= 0.05$ are connected with a line

rivers, Danish farms, Enron and ExprYeast-FunCat), while on the other four domains (ImCLEF07A, ImCLEF07D, Reuters and SeqAra-FunCat) the best performing type of model is the HSC architecture. We next discuss the statistically significant differences in more detail.

When HMC trees are the best performing method, they are statistically significantly better than the single-label trees (with the exception of ExprYeast-FunCat dataset where there is no statistically significant difference between the two methods). In the remaining cases, the differences are not statistically significant (although HMC trees are better than single-label trees also on ImCLEF07A and ImCLEF07D). HMC trees are statistically significantly better than HSC tree architecture on the Slovenian rivers and ExprYeast-FunCat datasets, and HSC tree architecture is statistically significantly better than HMC trees on the Reuters dataset.

We further relate the performance of the methods with the dataset properties from Table 3. HMC trees perform best on datasets with a large number of labels per example (25, 7, 2.84 and 2.28 labels per example for the Slovenian rivers, Danish farms, Enron and ExprYeast-FunCat datasets, respectively). Conversely, HSC tree architectures perform better on datasets with a small number of labels per example (1.2, 1, 1 and 0.94 for Reuters, ImCLEF07A and ImCLEF07D and SeqAra-FunCat datasets, respectively). The output hierarchy is much more populated in the former case, thus, allowing the learning of HMC trees to fully exploit the dependencies between the labels. This in turn provides predictive models with better predictive power. Similar behavior can be observed for the models that do not exploit the output hierarchy: the multi-label trees are better on datasets with more labels per example, while the single-label tree are better on datasets with fewer labels per example.

We next discuss the poor performance of the global models on the Reuters dataset. This is the only dataset where HMC trees have worse predictive performance than single-label trees. The poor predictive performance is mainly due to two reasons: (1) the dataset has a small number of labels per example and (2) the dataset is extremely high-dimensional and sparse. However, this prompts for further investigation and analysis using additional benchmark datasets that exhibit similar properties.

5.2 Performance of ensemble models

The results from the evaluation of the ensemble models are given in Table 5. There are two major findings that are made apparent by the results: the ensembles clearly outperform their single model counterparts and the performance of the ensembles with different information about the output structure is approximately the same. The first finding is somewhat expected and similar results were previously obtained (Kocev et al. 2013; Schietgat et al. 2010). The second finding, however, prompts further examination.

Kocev et al. (2013) showed that tree ensembles for both HMC and HSC perform equally well. Here, we can add that also tree ensembles for single-label classification and multi-label classification perform equally well as HMC and HSC ensembles. Moreover, the single-label tree ensembles often perform better than the competition. This could be due to the fact that ensembles are very powerful predictive models, whose performance is limited by the data quality rather than their inability to discover regularities in the data. In such cases, the hierarchical relationships among classes are less helpful (or even not helpful at all) for improving predictive performance than in the case of single tree models. However, we would like to emphasize that there is a notable difference between global and local ensemble models in terms of learning time: global ensembles are learnt much faster than the local ensembles.

Table 5 Performance of random forests and bagging in terms of AU$\overline{\text{PRC}}$, relative decrease between training set performance and test set performance (overfit score - OS) and Learning time (LT, in seconds)

		Random forest			Bagging		
Dataset	Method	AU$\overline{\text{PRC}}$	OS	LT	AU$\overline{\text{PRC}}$	OS	LT
	Single-label	0.442	0.558	703.9	0.432	0.568	1201.1
Slovenian	HSC	0.444	0.542	1115.9	0.434	0.555	2770.0
rivers	Multi-label	**0.446**	0.551	253.8	**0.439**	0.558	732.3
	HMC	**0.446**	0.552	56.6	**0.439**	0.559	132.4
	Single-label	0.806	0.055	67.9	0.801	0.087	193.6
Danish	HSC	0.810	0.061	65.3	**0.816**	0.091	176.0
farms	Multi-label	0.798	0.052	5.3	0.801	0.086	56.2
	HMC	**0.815**	0.058	6.1	0.812	0.098	20.2
	Single-label	**0.884**	0.116	513.1	**0.871**	0.129	4160.1
ImCLEF07A	HSC	0.851	0.145	190.5	0.846	0.151	1499.6
	Multi-label	0.860	0.140	137.8	0.849	0.151	808.5
	HMC	0.858	0.142	35.4	0.850	0.150	297.9
	Single-label	**0.870**	0.130	221.9	**0.861**	0.139	1839.0
ImCLEF07D	HSC	0.831	0.164	324.2	0.836	0.143	984.0
	Multi-label	0.853	0.147	61.4	0.849	0.145	507.5
	HMC	0.852	0.148	102.3	0.843	0.154	208.9
	Single-label	**0.565**	0.371	149.5	**0.568**	0.425	4191.4
Enron	HSC	0.561	0.331	173.1	0.564	0.408	1507.7
	Multi-label	0.544	0.356	14.8	0.562	0.423	638.9
	HMC	0.555	0.369	16.9	0.563	0.426	263.2
	Single-label	**0.485**	0.448	1189.8	**0.718**	0.282	50517.5
Reuters	HSC	0.176	0.313	570.9	0.675	0.320	32480.4
	Multi-label	0.439	0.264	54.9	0.623	0.376	12600.2
	HMC	0.330	0.309	185.9	0.642	0.358	3345.2
	Single-label	**0.341**	0.658	727.6	**0.396**	0.604	132347.8
SeqAra-	HSC	0.262	0.731	918.9	0.314	0.680	82598.1
FunCat	Multi-label	0.287	0.712	472.2	0.375	0.625	160563.2
	HMC	0.269	0.731	57.6	0.341	0.661	20389.0
	Single-label	**0.248**	0.744	2215.2	0.240	0.733	178879.9
ExprYeast-	HSC	0.227	0.754	618.4	0.232	0.757	26255.1
FunCat	Multi-label	0.230	0.731	1797.9	**0.250**	0.684	132862.1
	HMC	0.239	0.755	265.3	**0.250**	0.741	13408.4

The best predictive performance for each dataset is shown in bold.

We further examine the results by performing a statistical significance test. In particular, we performed the Friedman test to check whether the observed differences in performance among the ensemble methods are statistically significant when taken over all datasets. We present the result of this test in Fig. 5. The test did not find a statistically significant difference in the performance of the different ensemble models. However, the single-label classification ensembles are on average the best performing.

Next, we discuss the overfit score and of the models. The results show that all of the ensemble models overfit approximately the same on the training data as measured by the overfit score. Moreover, the overfit score is typically smaller for the ensemble models than the single-tree models. Since the ensemble models are not interpretable, we do not look at the model size (although global ensemble models have typically smaller size than local ensemble models).

We then compare the performance of the two ensemble construction methods: random forests and bagging. The performance of random forest and bagging is very close to each other, with the exception of the sparse Reuters datasets. The random forest method is not suitable for sparse datasets due to the relatively small number of randomly selected features in each node of the tree. These features could have zeros for each example value in a given leaf (due to the sparsity) which will lead to performance deterioration.

A closer inspection of the results reveals that the small differences in performance of the different types of ensemble models are connected to the number of labels per example, similarly as for the single tree models. On datasets with a less populated class hierarchy (ImCLEF07A, ImCLEF07D, Reuters and SeqAra-FunCat) the single-label models achieve better predictive performance than other models. On datasets with a more populated class hierarchy (Danish farms, Slovenian rivers, Enron and ExprYeast-FunCat) the difference between local and global models are less pronounced. On these datasets, there is no statistically significant difference (Figs. 6 and 7) between single-label model and methods which use the class hierarchy (HSC and HMC), with exception of random forest on ExprYeast-FunCat dataset, where single-label is significantly better than HSC but not than HMC. On datasets with a less populated hierarchy, single-label models are significantly better than HMC and HSC.

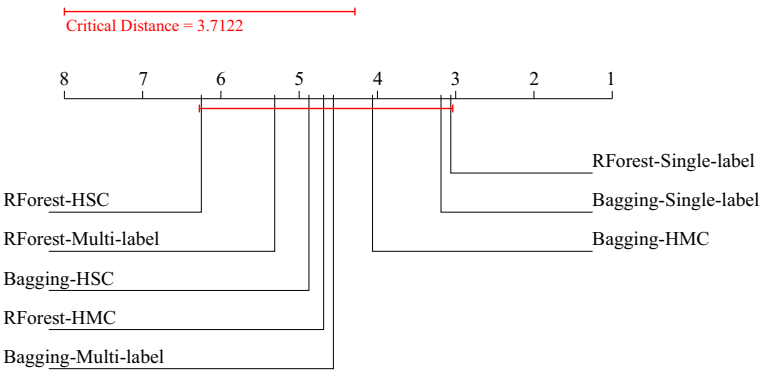

Fig. 5 Average ranks diagrams for the performance of all ensemble models in terms of AU$\overline{\text{PRC}}$ for all of the datasets. Better algorithms are positioned on the right-hand side, the ones that differ by less than the critical distance for a *p-value* = 0.05 are connected with a line

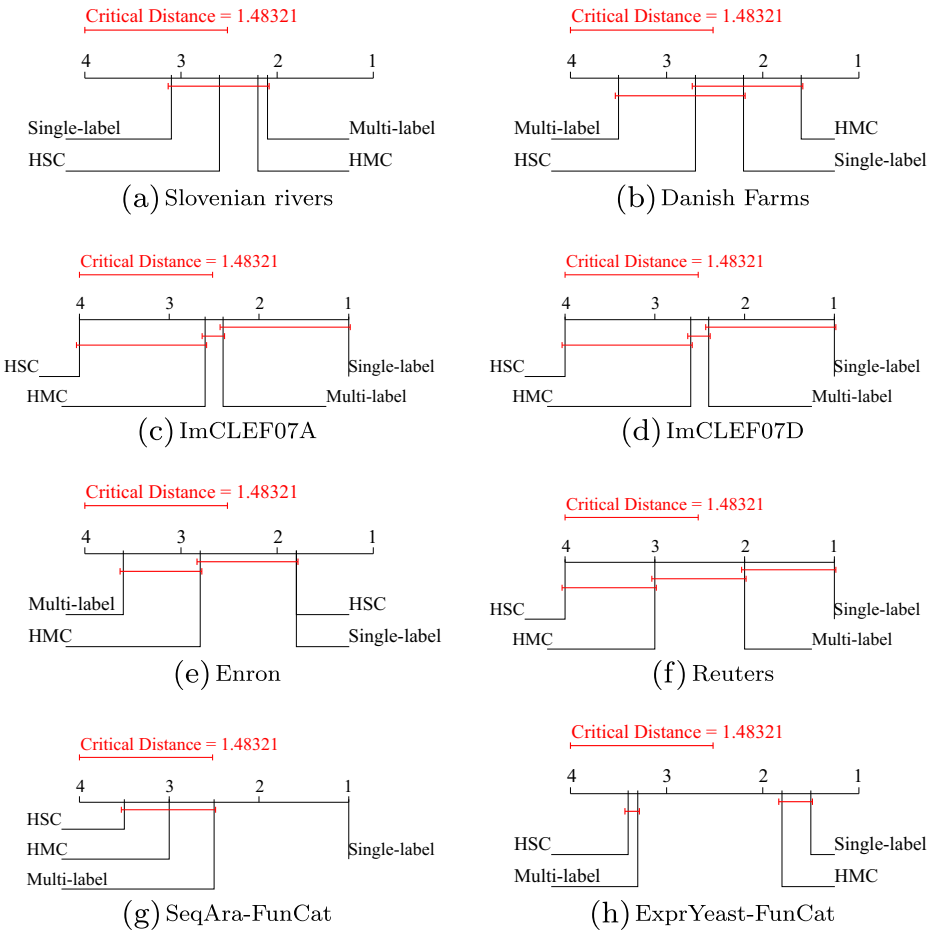

Fig. 6 Average ranks diagrams for the performance of random forests in terms of AU$\overline{\text{PRC}}$ for each of the eight datasets. Better algorithms are positioned on the right-hand side, the ones that differ by less than the critical distance for a *p-value* = 0.05 are connected with a line

5.3 Interpretability of predictive models

Besides the predictive power of the models, their interpretability is often a highly desired property, especially in domains such as habitat modelling. We discuss the interpretability of the models from the perspective of this domain. The single tree predictive models that we consider here (PCTs) are readily interpretable (ensemble models are not interpretable, and are not considered in this section). However, the difference in the interpretability of the local and global models is easy to notice. Firstly, global models, especially HMC trees, have considerably smaller complexity than the (collections of) local models (Table 4).

In Fig. 8, we present illustrative examples of the predictive models for the Slovenian rivers dataset. We show several PCTs for single-label classification, a tree for multi-label classification and a tree for hierarchical multi-label classification.

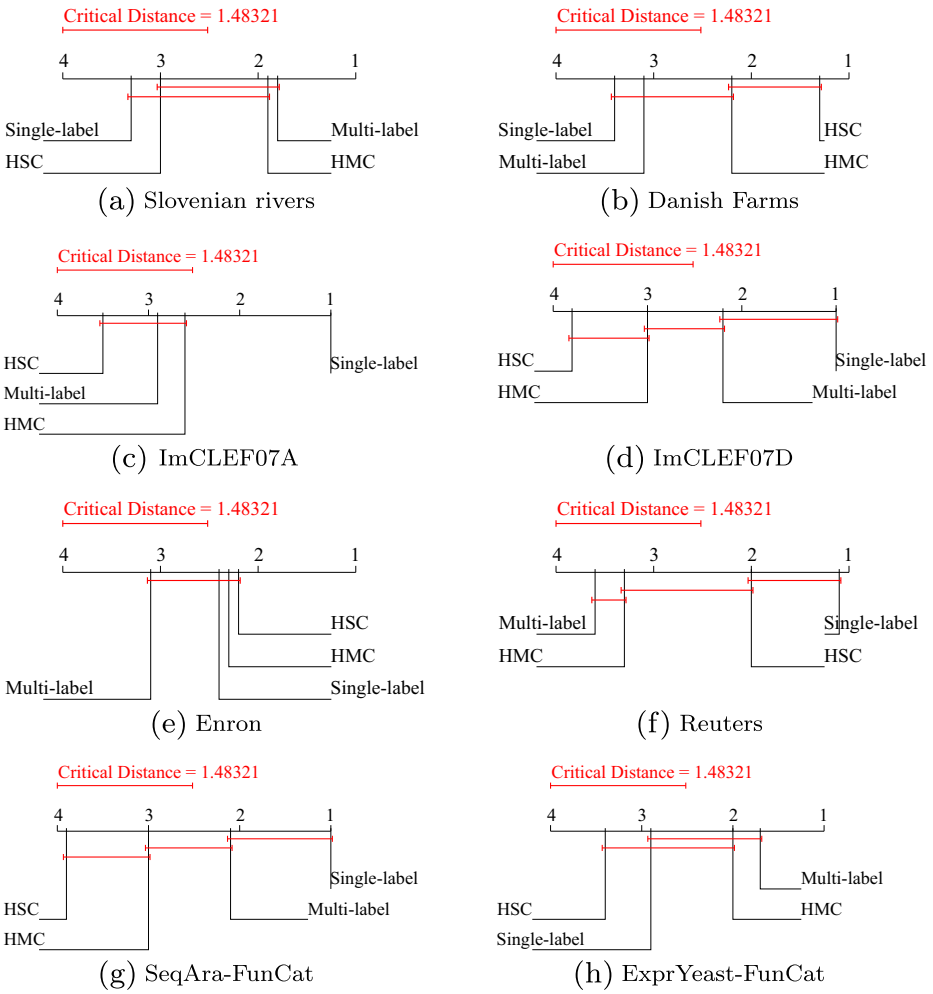

Fig. 7 Average ranks diagrams for the performance of bagging in terms of AU$\overline{\text{PRC}}$ for each of the eight datasets. Better algorithms are positioned on the right-hand side, the ones that differ by less than the critical distance for a *p-value* = 0.05 are connected with a line

We can immediately notice the differences between the local and global predictive models. The local models[2] offer information only for a part for the output space, i.e., they are valid just for a single species. In order to reconstruct the complete community model, one needs to look at the separate models and then try to make some overall conclusions. However, this could be very tedious or even impossible in domains with high biodiversity where there are hundreds of species present, such as the domain we consider here - Slovenian rivers.

[2]Note that the hierarchical single-label classification models will be similar to the single-label classification models, with the difference that the predictive models are organized into a hierarchical architecture. This makes the interpretation of the HSC models an even more difficult task.

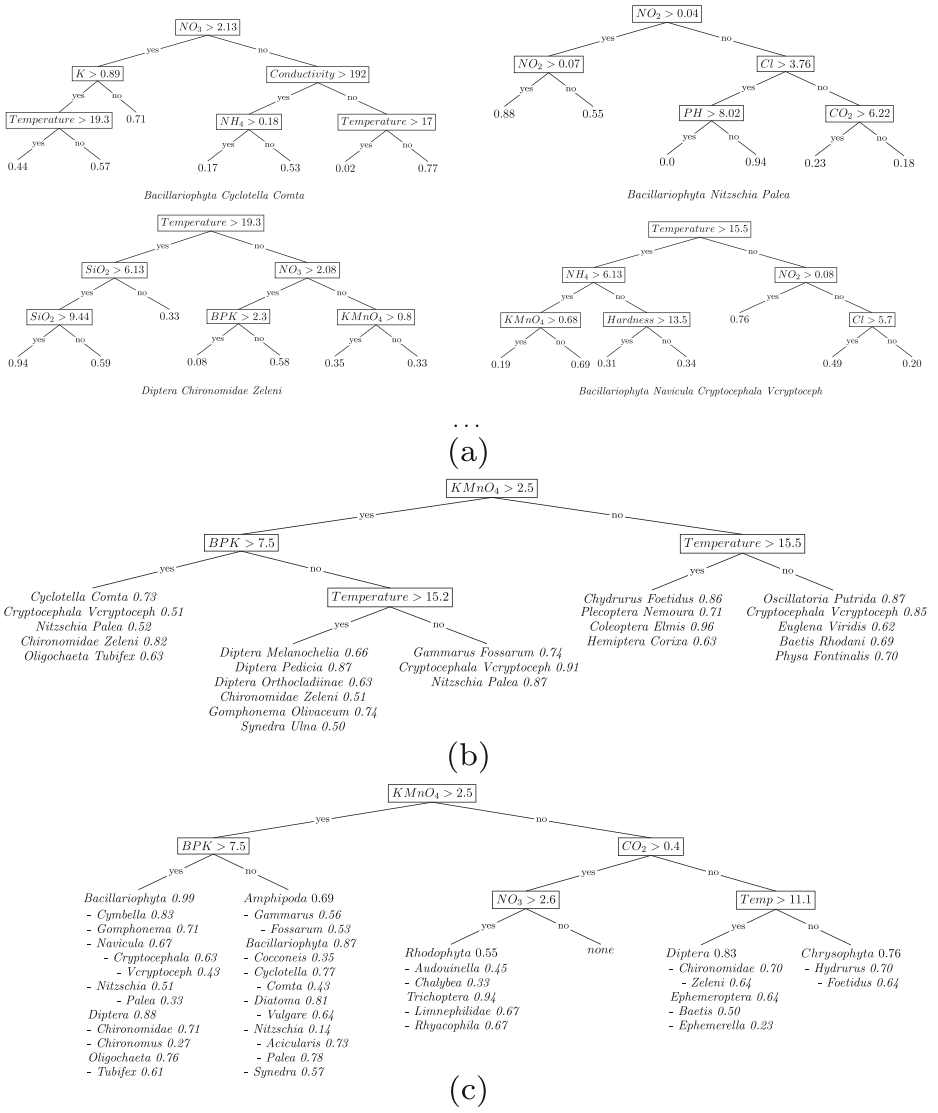

Fig. 8 Illustrative examples of decision trees (PCTs) learnt for the Slovenian rivers dataset. Single-label classification **(a)** produces a separate model for each of the species, whereas multi-label classification **(b)** and hierarchical multi-label classification **(c)** consider all of the species in a single tree

On the other hand, the global models are much easier to interpret. The single global model is valid for the complete structured output, i.e., for the whole community of species present in the ecosystem. The global models are able to capture the interactions present between the species, i.e., which species can co-exist at locations with given physico-chemical properties. Moreover, the HMC models, as compared to the multi-label models, offer additional information about the higher taxonomic ranks. For example, the HMC model could state that there is a low probability (0.27) that the species *Diptera*

chironomus is present under the given environmental conditions, while there is a high probability (0.88) that the genus *Diptera* is present (left-most leaf of the HMC tree in Fig. 8).

6 Conclusions

We address the task of learning predictive models for hierarchical multi-label classification, which take as input a tuple of attribute values and predict a set of classes organized into a hierarchy. We consider both global and local approaches for prediction of structured outputs. The former are based on a single model that predicts the entire output structure, while the latter are based on a collection of models, each predicting a part of the output structure.

We investigate the differences in performance and interpretability of the local and global models. More specifically, we examine whether including information in the form of hierarchical relationships among the labels and considering the multiple labels simultaneously helps to improve the performance of the predictive models. Moreover, we investigate whether inclusion of the information on the output structure also improves the performance of ensemble models. To this end, we consider four machine learning tasks: single-label classification, hierarchical single-label classification, multi-label classification and hierarchical multi-label classification; and two types of models able to solve those tasks: single-trees and ensembles.

We use predictive clustering trees as predictive models, since they can be used for solving all of the four tasks considered here. We construct and evaluate four types of single tree models: single-label trees, hierarchical single-label trees, multi-label trees and hierarchical multi-label trees. Additionally, for each of the mentioned tree types, we construct two types of ensembles: random forests and bagging.

We compare the performance of local and global predictive models on eight datasets from four practically relevant tasks: habitat modelling, image classification, text classification and functional genomics. The results show that the inclusion of the information about the class hierarchy has different importance for single tree models and ensemble models. Ensemble models are in general more accurate than single tree models, but are uninterpretable. Therefore, if the models needs to be interpreted, single tree models should be used.

The inclusion of the hierarchical information in the model construction phase for single trees improves the predictive performance irregardless of whether we use HMC trees or HSC tree architecture. HMC trees should be used on domains with a well-populated class hierarchy ($\mathcal{L} > 2$), while the HSC tree architecture will perform better if the number of labels per example is closer to one. We would like to note that HSC architectures are complex and not easy to interpret; therefore, a HMC model is still the best choice if interpretability is of more importance than predictive performance.

Inclusion of the information on the output structure (i.e., class hierarchy) brings less (or no) advantage in terms of predictive performance to ensemble methods as compared to single tree methods. However, there are considerable differences in the learning time between global and local ensemble methods. While, the single-label ensembles achieved the best predictive performance, HMC ensembles are much more efficient in terms of learning time than the single-label ensembles and should be used if time is an issue (especially random forests, since they are faster than bagging).

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).

References

Alaydie, N., Reddy, C.K., Fotouhi, F. (2012). Exploiting label dependency for hierarchical multi-label classification. In *Proceedings of the 16th Pacific-Asia conference on advances in knowledge discovery and data mining* (pp. 294–305). Berlin Heidelberg New York: Springer.

Bakır, G.H., Hofmann, T., Schölkopf, B., Smola, A.J., Taskar, B., Vishwanathan, S.V.N. (Eds.) (2007). *Predicting structured data*. Cambridge, MA: The MIT Press.

Barros, R.C., Cerri, R., Freitas, A.A., de Carvalho, A.C.P.L.F. (2013). Probabilistic clustering for hierarchical multi-label classification of protein functions. In H. Blockeel, K. Kersting, S. Nijssen, F. Železný (Eds.), *Machine learning and knowledge discovery in databases*, Lecture Notes in Computer Science, (Vol. 8189 pp. 385–400). Berlin Heidelberg: Springer.

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

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, *36*(1), 105–139.

Bi, W., & Kwok, J.T. (2012). Hierarchical multilabel classification with minimum bayes risk. In *Proceedings of the 12th international conference on data mining* (pp. 101–110).

Blockeel, H. (1998). Top-down induction of first order logical decision trees. Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium.

Blockeel, H., Bruynooghe, M., Džeroski, S., Ramon, J., Struyf, J. (2002). Hierarchical multi-classification. In *Proceedings of the ACM SIGKDD workshop on multi-relational data mining* (pp. 21–35).

Blockeel, H., Schietgat, L., Struyf, J., Džeroski, S., Clare, A. (2006). Decision trees for hierarchical multilabel classification: A case study in functional genomics. In *Knowledge discovery in databases: PKDD*, Lecture Notes in Computer Science (Vol. 4213 pp. 18–29). Berlin Heidelberg: Springer.

Blockeel, H., & Struyf, J. (2002). Efficient algorithms for decision tree cross-validation. *Journal of Machine Learning Research*, *3*, 621–650.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J. (1984). *Classification and regression trees*. London, UK: Chapman & Hall/CRC.

Cerri, R., Barros, R.C., de Carvalho, A.C.P.L.F. (2012). A genetic algorithm for hierarchical multi-label classification. In *Proceedings of the 27th annual ACM symposium on applied computing* (pp. 250–255).

Cerri, R., Barros, R.C., de Carvalho, A.C.P.L.F. (2014). Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, *80*(1), 39–56.

Clare, A. (2003). Machine learning and data mining for yeast functional genomics. Ph.D. thesis, University of Wales Aberystwyth, Aberystwyth, UK.

Clare, A., & King, R.D. (2003). Predicting gene function in *Saccharomyces cerevisiae*. *Bioinformatics*, *19*(S2), ii42–49.

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on machine learning* (pp. 233–240).

Demšar, D., Džeroski, S., Larsen, T., Struyf, J., Axelsen, J., Bruns-Pedersen, M., Krogh, P.H. (2006). Using multi-objective classification to model communities of soil. *Ecological Modelling*, *191*(1), 131–143.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Dietterich, T.G., Domingos, P., Getoor, L., Muggleton, S., Tadepalli, P. (2008). Structured machine learning: The next ten years. *Machine Learning*, *73*(1), 3–23.

Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S. (2008). Hierchical annotation of medical images. In *Proceedings of the 11th international multiconference - information society* (pp. 174–181). Ljubljana:JSI.

Džeroski, S. (2009). Machine learning applications in habitat suitability modeling. In S.E. Haupt, A. Pasini, C. Marzban (Eds.), *Artificial intelligence methods in the environmental sciences* (pp. 397–412): Springer Netherlands.

Džeroski, S., Demšar, D., Grbović, J. (2000). Predicting chemical parameters of river water quality from bioindicator data. *Applied Intelligence*, *13*(1), 7–17.

Estruch, V., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J. (2006). Web categorisation using distance-based decision trees. *Electronic Notes in Theoretical Computer Science*, *157*(2), 35–40.

Guan, Y., Myers, C.L., Hess, D.C., Barutcuoglu, Z., Caudy, A., Troyanskaya, O. (2008). Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, *9*(S1), S3+.

Kiritchenko, S., Famili, F., Matwin, S., Nock, R. (2006). Learning and evaluation in the presence of class hierarchies: Application to text categorization. In L. Lamontagne, M. Marchand (Eds.), *Advances in artificial intelligence*, Lecture Notes in Computer Science, (Vol. 4013 pp. 395–406). Berlin Heidelberg: Springer.

Klimt, B., & Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In J.F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), *Machine learning: ECML*, Lecture Notes in Computer Science, (Vol. 3201 pp. 217–226). Berlin Heidelberg: Springer.

Kocev, D., Vens, C., Struyf, J., Džeroski, S. (2013). Tree ensembles for predicting structured outputs. *Pattern Recognition*, *46*(3), 817–833.

Kriegel, H.P., Borgwardt, K., Kröger, P., Pryakhin, A., Schubert, M., Zimek, A. (2007). Future trends in data mining. *Data Mining and Knowledge Discovery*, *15*, 87–97.

Lehmann, T., Schubert, H., Keysers, D., Kohnen, M., Wein, B. (2003). The IRMA code for unique classification of medical images. In *Medical imaging: PACS and integrated medical information systems: Design and evaluation* (pp. 440–451).

Levatić, J., Kocev, D., Džeroski, S. (2013). The use of the label hierarchy in hmc improves performance: A case study in predicting community structure in ecology. In *Proceedings of the workshop on new frontiers in mining complex patterns held in conjunction with ECML/PKDD2013* (pp. 189–201).

Levatić, J., Kocev, D., Džeroski, S. (2014). The use of the label hierarchy in hierarchical multi-label classification improves performance. In A. Appice, et al. (Eds.), *New frontiers in mining complex patterns*, Lecture Notes in Computer Science, (Vol. 8399 pp. 1–16): Springer International Publishing.

Lewis, D.D., Yang, Y., Rose, T.G., Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, *5*, 361–397.

Obozinski, G., Lanckriet, G., Grant, C., Jordan, M.I., Noble, W.S. (2008). Consistent probabilistic outputs for protein function prediction. *Genome Biology*, *9*(S1), S6+.

Otero, F.E., Freitas, A.A., Johnson, C.G. (2010). A hierarchical multi-label classification ant colony algorithm for protein function prediction. *Memetic Computing*, *2*(3), 165–181.

Quinlan, J.R. (1993). *C4. 5: Programs for machine learning* Vol. 1. San Francisco, CA: Morgan Kaufmann.

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

Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M., et al. (2004). The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, *32*(18), 5539–5545.

Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., Džeroski, S. (2010). Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, *11*(2), 1–14.

Seni, G., & Elder, J.F. (2010). *Ensemble methods in data mining: Improving accuracy through combining predictions*: Morgan & Claypool Publishers.

Silla, C., & Freitas, A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, *22*(1-2), 31–72.

Silla, C.N., & Freitas, A.A. (2009). A global-model naive bayes approach to the hierarchical prediction of protein functions. In *Proceeding of the 9th IEEE international conference on data mining* (pp. 992–997).

Slavkov, I., Gjorgjioski, V., Struyf, J., Džeroski, S. (2010). Finding explained groups of time-course gene expression profiles with predictive clustering trees. *Molecular BioSystems*, *6*(4), 729–740.

Valentini, G. (2011). True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *8*(3), 832–847.

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