The Use of the Label Hierarchy in Hierarchical Multi-label Classification Improves Performance

Jurica Levatić[1,2](✉), Dragi Kocev[1], and Sašo Džeroski[1,2]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
{Jurica.Levatic,Dragi.Kocev,Saso.Dzeroski}@ijs.si
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

Abstract. We address the task of learning models for predicting structured outputs. We consider both global and local approaches to the prediction of structured outputs, the former based on a single model that predicts the entire output structure and the latter based on a collection of models, each predicting a component of the output structure. More specifically, we compare local and global approaches in terms of predictive performance, learning time and model complexity. Moreover, we discuss the interpretability of the obtained models. We evaluate the predictive performance of the considered approaches on six case studies from three domains: ecological modelling, text classification and image classification. Finally, we identify the properties of the tasks at hand that lead to the differences in performance.

Keywords: Predictive clustering trees · Hierarchical multi-label classification · Multi-label classification · Habitat modelling · Text classification · Image classification

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. If the examples belong to two classes (e.g., the example has some property or not) the task is called binary classification. 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,

A. Appice et al. (Eds.): NFMCP 2013, LNAI 8399, pp. 162–177, 2014.
DOI: 10.1007/978-3-319-08407-7_11, © Springer International Publishing Switzerland 2014

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. [1] and Dietterich et al. [2] 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 [3]), have been proposed [4]. These methods can be categorized into two groups of methods for solving the problem of predicting structured outputs [3,4]. 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 [3] 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 that can be formulated to solving the task of HMC: 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.

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 [5–7].

We perform the evaluation of the predictive models on six practically relevant HMC datasets. The datasets come from three different domains: habitat modelling, image classification and text classification. We consider habitat models for Collembola communities in the soils of Denmark [8] and communities of organisms living in Slovenian rivers [9]. Next, we use two datasets from the 2007 CLEF cross-language image retrieval campaign [10], 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 [11] and categorization of Reuters newswire stories [12].

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

2 Predictive Modelling for HMC

In this section, we present in more detail methodology used to construct the predictive models. We first present global approaches 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).

2.1 Global Predictive Models

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 [13], which is available for download at http://clus.sourceforge.net.

PCTs are induced with a standard *top-down induction of decision trees* (TDIDT) algorithm [14]. 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 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

Table 1. The top-down induction algorithm for PCTs.

```
procedure PCT
Input: A dataset $E$
Output: A predictive clustering tree
$(t^*, h^*, \mathcal{P}^*) = \mathrm{BestTest}(E)$
if $t^* \neq none$ then
    for each $E_i \in \mathcal{P}^*$ do
        $tree_i = \mathrm{PCT}(E_i)$
    return node$(t^*, \bigcup_i \{tree_i\})$
else
    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 ($E$)
$(t^*, h^*, \mathcal{P}^*) = (none, 0, \emptyset)$
for each possible test $t$ do
    $\mathcal{P}$ = partition induced by $t$ on $E$
    $h = Var(E) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var(E_i)$
    if $(h > h^*) \wedge \mathrm{Acceptable}(t, \mathcal{P})$ then
        $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
return $(t^*, h^*, \mathcal{P}^*)$
```

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) [6], hierarchical multi-label classification [7] and prediction of time-series [15]. 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 [16]). 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 [7]. 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

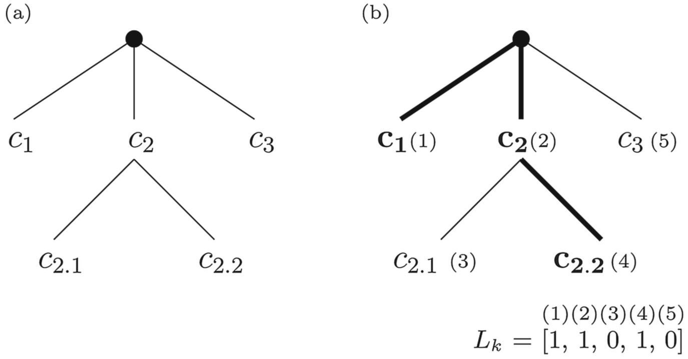


Fig. 1. Toy examples of a hierarchy structured as a tree. (a) Class label names contain information about the position in the hierarchy, e.g., $c_{2.1}$ is a subclass of c_2. (b) The set of classes $S_1 = \{c_1, c_2, c_{2.2}\}$, shown in bold, are represented as a vector (L_k).

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. 1(a,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. 1(b)) and $S_2 = \{c_2\}$, respectively. We use a vector representation with consecutive components 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.

2.2 Local Predictive Models

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 node, and local classifiers per parent node (see [3] for details).

Vens et al. [7] 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 faster induction times.

In particular, the CLUS-HSC algorithm by Vens et al. [7] 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 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.

3.1 Data Description

We use six datasets, which come from three domains: habitat modeling, image classification and text classification. The main statistics of the datasets are given in Table 2. We can observe that the datasets vary in the size, number of attributes and characteristics of the label hierarchy.

Habitat modelling [17] 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: river and soil. Namely, we construct habitat models for river water organisms living in Slovenian rivers [9] and for soil microarthropods from Danish farms [8]. 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 the image content 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 [10]: ImCLEF07A and ImCLEF07D. The goal in these datasets is to recognize which part of the human anatomy is present in the image or the orientation of the body part, respectively. Images are represented by using edge

Table 2. 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	$\mathcal{L}$	$\|\mathcal{H}\|$	$\mathcal{H}_d$	$\overline{\mathcal{L}_L}$
Slovenian rivers [9]	1060	0/16	491	724	4	25
Danish farms [8]	1944	132/5	35	72	3	7
ImCLEF07A [10]	11006	0/80	63	96	3	1
ImCLEF07D [10]	11006	0/80	26	46	3	1
Enron [11]	1648	0/1001	50	54	3	2.84
Reuters [12]	6000	0/47236	77	100	4	1.2

histograms. An edge histogram represents the frequency and the directionality of the brightness changes in the image. The structured output space consists of labels organized in hierarchy. They correspond to the anatomical (ImCLEF07A) and directional (ImCLEF07D) axis of the IRMA (Image Retrieval in Medical Applications) code [18].

Text classification is the problem of automatic annotation of textual documents to one or more categories. We used two datasets from this domain: Enron and Reuters. Enron is a labeled subset of the Enron corpus [11], 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) [12]. 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.

3.2 Experimental Design

We constructed four types of predictive models, as described in the previous section, for each of the case studies. First, we constructed single-label classification trees for each label (i.e., leaf in the label hierarchy) separately. Next, we constructed hierarchical single-label classification tree architecture. Furthermore, we constructed a multi-label classification tree for all of the leaf labels, without using the hierarchy. Finally, we constructed a hierarchical multi-label classification tree for all of the labels by using the hierarchy.

We used F-test pruning to ensure that the produced models are not overfitted and have better predictive performance [7]. The exact Fisher test is used to check whether a given split/test in an internal node of the tree results in a

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

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.

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 nodes 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.

3.3 Evaluation Measures

We evaluate the algorithms by using the Area Under the Precision-Recall Curve (AUPRC), and in particular, the Area Under the Average Precision-Recall Curve ($\mathrm{AU}\overline{\mathrm{PRC}}$) as suggested by Vens et al. [7]. The points in the PR space are obtained by varying the value for the threshold τ from 0 to 1 with step 0.02. For each value of the threshold τ, precision and recall are micro-averaged as follows:

$$\overline{Prec} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i}, \quad \text{and} \quad \overline{Rec} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i}$$

where i ranges over all classes that are leafs in the output hierarchies.

We measure the performance of the predictive models along several dimensions. First, we estimate the predictive performance of the models 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 over-fit on the training data. To this end, we use the relative decrease of the performance from the training set to the one obtained with 10-fold cross-validation. We define this as over-fit score ($OS = \frac{\mathrm{AU}\overline{\mathrm{PRC}}_{train} - \mathrm{AU}\overline{\mathrm{PRC}}_{test}}{\mathrm{AU}\overline{\mathrm{PRC}}_{train}}$). The smaller values of this score mean that the overfitting of the models is smaller. Finally, we measure the model complexity and the time efficiency of the predictive models. 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 sum 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.

We adopt the recommendations by Demšar [19] for the statistical evaluation of the results. We use the corrected non-parametric Friedman test for statistical significance on the per-fold-data for the folds of 10-fold cross validation for each dataset separately. Afterwards, to check where the statistically significant differences appear (between which methods), we use the Nemenyi post-hoc test (Nemenyi, 1963). We present the result from the Nemenyi post hoc test with an average ranks diagram as suggested by Demšar [8]. The ranks are depicted on the axis, in such a manner that the best ranking algorithms are at the rightmost side of the diagram. The algorithms that do not differ significantly (in performance) are connected with a line.

4 Results and Discussion

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

The results from the evaluation of the predictive models are given in Table 3. 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 over fit less as compared to the other models. Moreover, the results indicate that the HMC trees over-fit 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 are statistically significant for each dataset separately. The results from this analysis show that the difference in performance is statistically significant for each dataset with *p-value* smaller than $3 \cdot 10^{-5}$.

Figure 2 presents the average ranks from the Nemenyi post-hoc test for all types of models. The diagrams show that the HMC models are best performing on three domains (Slovenian rivers, Danish farms and Enron), while on the other three domains (ImCLEF07A, ImCLEF07D and Reuters) the best performing type of model is the HSC architecture. We next discuss the statistically significant differences in the datasets in more detail.

When HMC trees are the best performing method, they are statistically significantly better than the single-label trees. 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 only on the Slovenian rivers dataset, and HSC tree architecture is statistically significantly better than HMC trees on the Reuters dataset.

We further complement the information on the performance with the dataset properties from Table 2. HMC trees perform best on datasets with a large number of labels per example (25, 7 and 2.84 labels per example for the Slovenian rivers, Danish farms and Enron datasets, respectively). Conversely, HSC tree architectures perform better on datasets with a small number of labels per example (1.2, 1 and 1 for Reuters, ImCLEF07A and ImCLEF07D datasets, respectively).

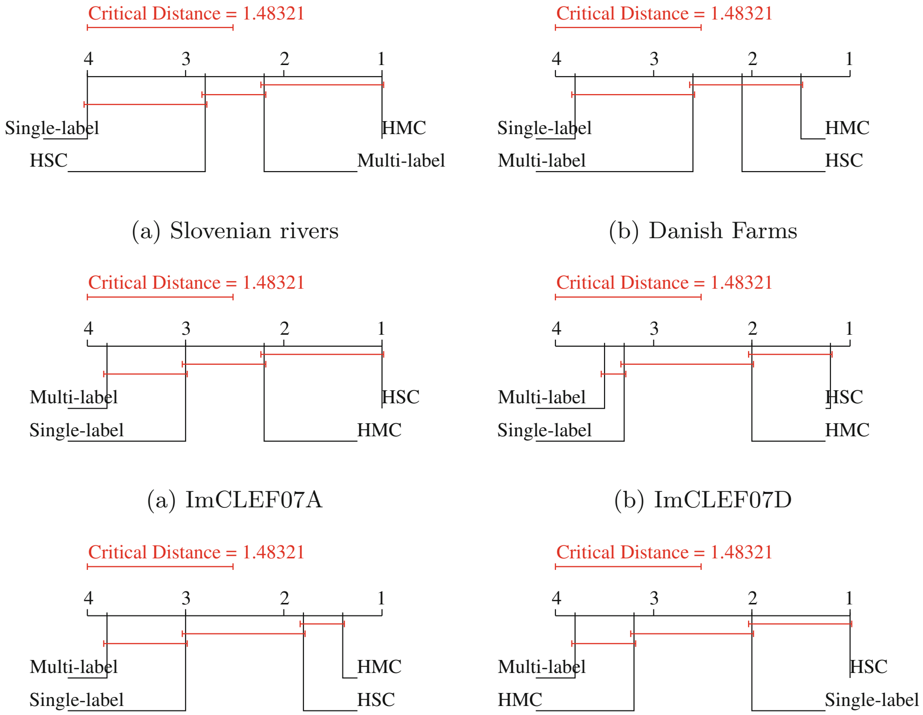
Table 3. Performance of the methods in terms of $\mathrm{AU}\overline{\mathrm{PRC}}$, decrease of training set performance relative to test set performance. (OS), Learning time (in seconds) and model complexity (the number of nodes in the decision trees). The best predictive performance for each dataset is shown in bold.

Dataset	Method	$\mathrm{AU}\overline{\mathrm{PRC}}$	OS	Learning time	Complexity
Slovenian rivers	Single-label	0.239	0.692	23.3	15336
	HSC	0.309	0.591	10.2	25035
	Multi-label	0.322	0.007	9.4	1
	HMC	**0.374**	0.132	0.6	37
Danish farms	Single-label	0.790	0.099	3.7	2605
	HSC	0.808	0.083	1.3	2873
	Multi-label	0.801	0.112	0.7	265
	HMC	**0.815**	0.065	0.4	259
ImCLEF07A	Single-label	0.571	0.375	74.4	3957
	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
ImCLEF07D	Single-label	0.515	0.483	35.4	7418
	HSC	**0.631**	0.361	20.1	9764
	Multi-label	0.511	0.484	7.78	3675
	HMC	0.615	0.198	3.0	685
Enron	Single-label	0.398	0.495	114.7	1740
	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
Reuters	Single-label	0.431	0.546	970.8	3591
	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

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 examples and (2) the dataset is extremely high-dimensional and sparse. However, this prompts for additional investigation and analysis using more benchmark datasets that exhibit similar properties.

Besides the predictive power of the models, their interpretability is often a highly desired property, especially in domains such as habitat modelling.



(a) Slovenian rivers

(b) Danish Farms

(a) ImCLEF07A

(b) ImCLEF07D

(a) Enron

(b) Reuters

Fig. 2. Average ranks diagrams for the performance of the four methods in terms of $\overline{AUPRC}$ for each of the six 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 discuss the interpretability of the models from the perspective of this domain. The predictive models that we consider here (PCTs) are readily interpretable. 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 3). In Fig. 3, 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.

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

[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.

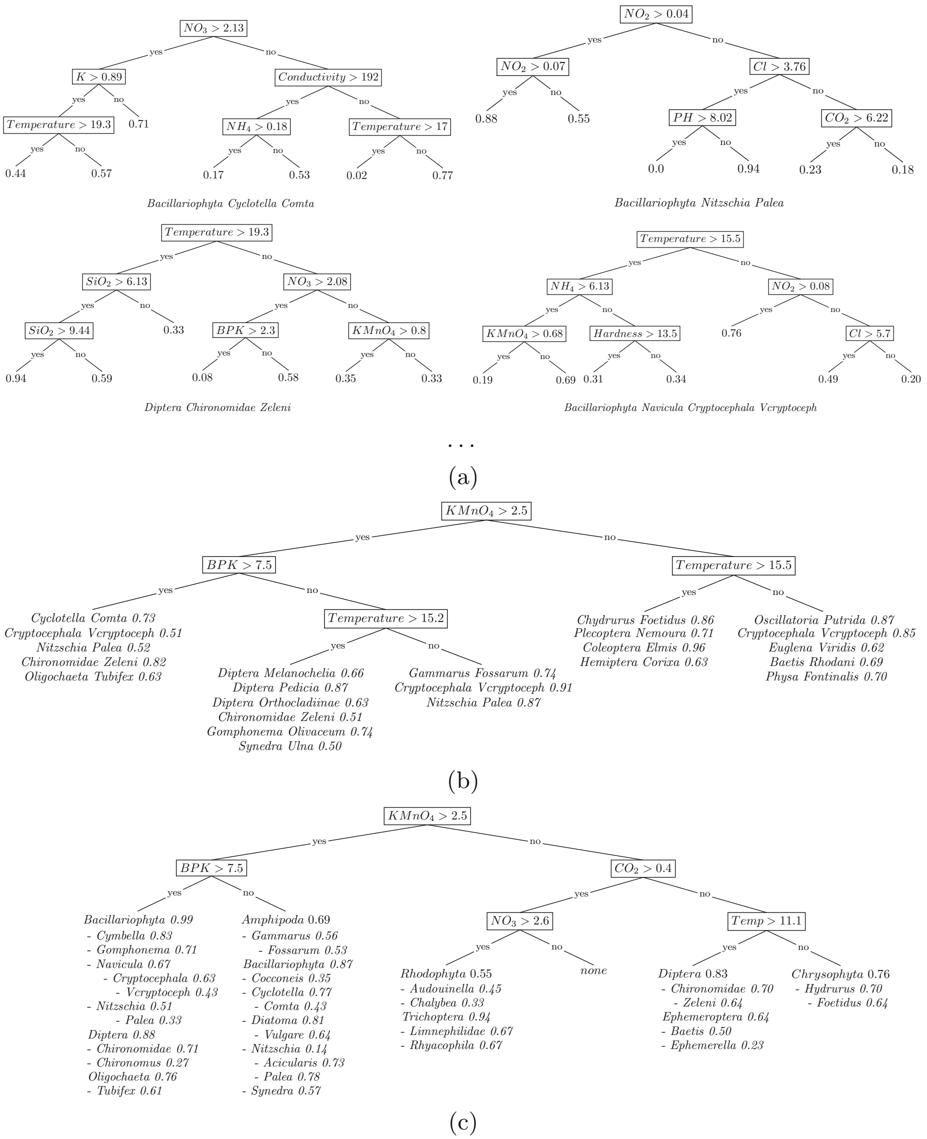


Fig. 3. Illustrative examples of decision trees (PCTs) learnt for the Slovenian rivers dataset. Single-label classification (a) produces a separate model for each of the species, whereas multi-label classification (b) and hierarchical multi-label classification (c) consider all of the species in a single tree.

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.

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 a 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 the is a high probability (0.88) that the genus *Diptera* is present (left-most leaf of the HMC tree in Fig. 3).

5 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 component 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. To this end, we consider four machine learning tasks: single-label classification, hierarchical single-label classification, multi-label classification and hierarchical multi-label classification.

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 trees: single-label trees, hierarchical single-label trees, multi-label trees and hierarchical multi-label trees.

We compare the performance of local and global predictive models on six datasets from three practically relevant tasks: habitat modelling, image classification and text classification. The results show that the inclusion of the hierarchical information in the model construction phase, i.e., for HMC trees and for HSC tree architecture, improves the predictive performance. The improvement in performance for HMC trees is more pronounced on domains that have a more populated hierarchy, i.e., on datasets with a larger number of labels per example. On the other hand, HSC tree architecture perform better in the domains where the number of labels per example is closer to one. Moreover, the models that take the hierarchy into account tend to over-fit less than the models that do not include such information (this is especially true for the HMC trees). Finally, the global methods produce less complex models and are much easier to interpret than the local models offering an overview of the complete output hierarchy.

All in all, the inclusion of hierarchy information improves the performance of the predictive models and the global models are more efficient and easier to interpret than local models.

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

1. Kriegel, H.P., Borgwardt, K., Kröger, P., Pryakhin, A., Schubert, M., Zimek, A.: Future trends in data mining. Data Min. Knowl. Disc. **15**, 87–97 (2007)
2. Dietterich, T.G., Domingos, P., Getoor, L., Muggleton, S., Tadepalli, P.: Structured machine learning: the next ten years. Mach. Learn. **73**(1), 3–23 (2008)
3. Silla, C., Freitas, A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Disc. **22**(1–2), 31–72 (2011)
4. Bakır, G.H., Hofmann, T., Schölkopf, B., Smola, A.J., Taskar, B., Vishwanathan, S.V.N.: Predicting Structured Data. The MIT Press, Cambridge (2007)
5. Blockeel, H.: Top-down induction of first order logical decision trees. Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium (1998)
6. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recogn. **46**(3), 817–833 (2013)
7. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Mach. Learn. **73**(2), 185–214 (2008)
8. Demšar, D., Džeroski, S., Larsen, T., Struyf, J., Axelsen, J., Bruns-Pedersen, M., Krogh, P.H.: Using multi-objective classification to model communities of soil. Ecol. Modell. **191**(1), 131–143 (2006)
9. Džeroski, S., Demšar, D., Grbović, J.: Predicting chemical parameters of river water quality from bioindicator data. Appl. Intell. **13**(1), 7–17 (2000)
10. Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S.: Hierchical annotation of medical images. In: Proceedings of the 11th International Multiconference - Information Society IS 2008, IJS, Ljubljana, pp. 174–181 (2008)
11. Klimt, B., Yang, Y.: The enron corpus: a new dataset for email classification research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 217–226. Springer, Heidelberg (2004)
12. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: a new benchmark collection for text categorization research. J. Mach. Learn. Res. **5**, 361–397 (2004)
13. Blockeel, H., Struyf, J.: Efficient algorithms for decision tree cross-validation. J. Mach. Learn. Res. **3**, 621–650 (2002)
14. Breiman, L., Friedman, J., Olshen, R., Stone, C.J.: Classification and Regression Trees. Chapman & Hall/CRC, New York (1984)
15. Slavkov, I., Gjorgjioski, V., Struyf, J., Džeroski, S.: Finding explained groups of time-course gene expression profiles with predictive clustering trees. Mol. BioSyst. **6**(4), 729–740 (2010)
16. Clare, A.: Machine learning and data mining for yeast functional genomics. Ph.D. thesis, University of Wales Aberystwyth, Wales, UK (2003)
17. Džeroski, S.: Machine learning applications in habitat suitability modeling. In: Haupt, S.E., Pasini, A., Marzban, C. (eds.) Artificial Intelligence Methods in the Environmental Sciences, pp. 397–412. Springer, Berlin (2009)

18. Lehmann, T., Schubert, H., Keysers, D., Kohnen, M., Wein, B.: The IRMA code for unique classification of medical images. In: Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation, pp. 440–451 (2003)
19. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)