Decomposition of the output space in multi-label classification using feature ranking

Stevanche Nikoloski[2,3], Dragi Kocev[1,2], and Sašo Džeroski[1,2]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] Teagasc, Environment Soils and Land-use Department, Co. Wexford, Ireland
stevanche.nikoloski@ijs.si, dragi.kocev@ijs.si, saso.dzeroski@ijs.si

Abstract. Motivated by the increasing interest for the task of multi-label classification (MLC) in recent years, in this study we investigate a new approach for decomposition of the output space with the goal to improve the predictive performance. Namely, the decomposition of the output/label space is performed by constructing a label hierarchy and then approaching the MLC task as a task of hierarchical multi-label classification (HMLC). Our approach is as follows. We first perform feature ranking for each of the labels separately and then represent each of the labels with its corresponding feature ranking. The construction of the hierarchy is performed by the (hierarchical) clustering of the feature rankings. To this end, we employ three clustering methods: agglomerative clustering with single linkage, agglomerative clustering with complete linkage and predictive clustering trees. We then use predictive clustering trees to estimate the influence of the constructed hierarchies, i.e., we compare the predictive performance of models with and without exploiting the hierarchy. Moreover, we investigate the influence of the hierarchy in the context of single models and ensembles of models. We evaluate the proposed approach on 6 datasets and show that the proposed method can yield predictive performance boost across several evaluation measures.

Keywords: multi-label classification, hierarchy construction, feature ranking, decomposition of label space

1 Introduction

Considering the increasing number of new applications of multi-label learning, researchers are very motivated to develop new methods and to extensively research new ideas related to multi-label classification and decomposing of a label space. The different application problems include video and image annotations (new movie clips, genres), predicting genes and proteins (functional genomics), classification of a tweets and music into emotions, text classification (web-pages, bookmarks, e-mails,...) and others.
Multi-label classification (MLC) is a learning from a data examples where each example is associated with multiple labels. MLC deals with a label dependencies and relations which is orthogonal with existing traditional methods which take

into account label independence and learn independent functions from mapping from input space to the output (label) space.

Considering only label space and trying to construct the dependencies of a labels, can be very tedious and expensive process. Deciding also on the representation language of these dependencies can be complicated task. Typically, these dependencies are represented as hierarchies of labels. The hierarchies can then be constructed in a data-driven manner using the descriptive space and/or the label space. With this, we have an automatic and relatively cheap process to obtain the representation of the possible dependencies in the label space.

Madjarov et al. (2014) [6] present an extensive study based on evaluation of different data-derived label hierarchies in multi-label classification. The hierarchies are constructed using four clustering algorithms, agglomerative clustering with single and complete linkage, balanced k-means and predictive clustering trees. Clustering here is performed on the raw output (label) space containing labels co-occurrences (see Fig 1 - *left* table).

Next, Szymansky et al. (2016) [10] in their study address the question whether data-driven approach is significantly better than a random choice in label space division for multi-label classification as performed by RAkELd. Their results show that in almost all cases data-driven partitioning outperforms the baseline RAkELd in all evaluation measures, but Hamming loss.

In this study, we present a different idea for data-driven decomposition of label space in multi-label classification, using feature ranking. Namely, instead of using the original label space consisting of label co-occurrences (see Fig 1 - *left* table), we calculate a ranking scores of each features for each label using the Random forest algorithm for feature ranking [1] (see Fig 1 - *right* table). The obtained structure is used to evaluate whether considering the dependency in the label space can provide better predictive performance than using flat MLC problem, in other words, whether considering the MLC task as hierarchical MLC can yield better predictive performance.

	Input Space				Ouput (Label) Space					
	LowPeakAmp	LowPeakBPM	HighPeakAmp	...	λ1	λ2	λ3	λ4	λ5	λ6
#1	0.036299	-58.962537	4.698083	...	1	0	0	0	0	0
#2	0.161218	-77.425609	3.09809	...	0	0	1	0	1	1
#3	0.115987	-61.893693	4.478436	...	1	1	1	1	0	0
#4	0.086016	-83.295694	3.786274	...	1	0	1	0	1	1
...	...	...	...	...	...	...	...	...	...	...

Decomposed Output Space					
FRank λ1	FRank λ2	FRank λ3	FRank λ4	FRank λ5	FRank λ6
1.3686	12.632	22.677	14.056	5.5625	1.3281
1.5876	11.888	26.345	9.1765	5.5659	0.6738
1.4325	11.076	43.997	8.9512	19.03	1.4792
1.7408	7.8361	8.2058	10.063	8.6095	0.5607
...	...	...	...	...	...

Fig. 1: Excerpt from the original *emotions* dataset containing label co-occurrences (*left* table) and transformation of the output space using feature ranking method (*right* table)

Using the decomposed label space (see black filled cells on Fig 1 - *right* table), we use the same unsupervised clustering algorithms for hierarchy creation, as in the study by Madjarov et al (2014) [6].

The experimental evaluation is performed on 6 benchmark datasets from different domains: text, image, music and video classification and gene function prediction. The predictive performance of the methods is assessed using 9 evaluation measures used in the context of MLC.

The obtained results indicate that using the methods for creating the hierarchies using feature ranking can yield a better predictive performance as compared to the original flat MLC methods without hierarchy.

The reminder of this paper is organized as follows. Section 2 presents the background work, i.e., multi-label classification and hierarchical multi-label classification methods. Then, in Section 3 we present the decomposition of the output space using feature ranking. In Section 4, we show the experimental design. Results of our research are given in Section 5 and finally Section 6 concludes this paper.

2 Background

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

Multi-label learning is dealing with learning from examples which are associated to more than one label. There is a predefined set of labels containing all possible labels. There are two types of multi-label learning tasks: multi-label classification and multi-label ranking. The main goal of multi-label classification is to create a predictive model that will output a set of relevant labels for a given, previously unseen example. Multi-label ranking, on the other hand, can be understood as learning a model that, for each unseen examples, associates a list of rankings (preferences) on the labels from a given set of possible labels and a bipartite partition of this set into relevant and irrelevant labels. An extensive bibliography of methods for multi-label learning can be found in [11], [7] and the references therein.

The task of multi-label learning is defined as follows [5]. The input space $\mathcal{X}$ that consists of vectors of values of nominal or numeric data types i.e., $\forall x_i \in \mathcal{X}, x_i = (x_{i1}, x_{i2}, \ldots x_{iD})$, where D is a number of descriptive attributes. The output space $\mathcal{Y}$ consists of a subset of a finite set of disjoint labels $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_Q\}$ ($\mathrm{Q} > 1$ and $\mathcal{Y} \subseteq \mathcal{L}$). Given this, each example is a pair of a vector and a set from the input and output space, respectively. All of the examples then form the set of examples E. The goal is then to find a function $h : \mathcal{X} \to 2^{\mathcal{L}}$ such that from the input space assigns a set of labels to each example.

The main difference between multi-label classification and hierarchical multi-label classification (HMLC) is that in the latter the labels from the label space are organized into a hierarchy. A given example that is labeled with a given label, then it is also labeled with all its parent labels (known as a hierarchy constraint). Furthermore, an example can be labeled with multiple labels, simultaneously. That means a several paths can be followed from the root node in order to arrive at a given label.

Here, the output space $\mathcal{Y}$ is defined with a label hierarchy $(\mathcal{L}, \leq_h)$, where $\mathcal{L}$ is a set of labels and $\leq_h$ is a partial order parent-child relationship structured as a tree ($\forall \lambda_1, \lambda_2 \in \mathcal{L} : \lambda_1 \leq_h \lambda_2$ if and only if λ_1 is a parent of λ_2) [5]. Each example from the set of examples E is a pair of a vector and a set from the input and output space respectively, where the set satisfies the hierarchy constraint, i.e., $E = \{(x_i, \mathcal{Y}_i) | x_i \in \mathcal{X}, \mathcal{Y} \subseteq \mathcal{L}, \lambda \in \mathcal{Y}_i \Rightarrow \forall \lambda' \leq_h \lambda : \lambda' \in \mathcal{Y}_i, 1 \leq i \leq N\}$, where N is a number of examples in E. Same conditions as in multi-label classification should be satisfied for the quality criterion q (high predictive performance and low computational cost). In (Silla, Freitas 2011) [8], an extensive bibliography is given, where the HMLC task is presented across different application domains.

3 Decomposition of label spaces using feature ranking

In this section, we explain our idea for decomposition of label space using feature ranking and also we describe the different clustering algorithms we use.
Our proposed method about label space decomposition is outlined in procedure *DecomposeLabelSpaceFR*. First, we take the original training dataset D^{train} and using Random forest method we create feature ranks for each label separately. We then construct a dataset D^{ranks} consisting of the feature ranks. We can obtain a hierarchy using one of the clustering algorithms described bellow. The hierarchy is then used to pre-process the datasets and obtain their hierarchical variants D_H^{train} and D_H^{test}. At the end, we learn the HMC predictive models.

---

procedure DecomposeLabelSpaceFR(D^{train}, D^{test})
returns performance

1: *// create feature importance (.fimp) file with Random forest*
2: FimpPath = CreateFimp(D^{train});
3: *// Create new arff with feature ranks from fimp file*
4: D^{ranks} = CreateArffFromFimp(FimpPath);
5: **hierarchy** = Clustering(D^{ranks});
6: *//transform multi-label dataset to hierarchical multi-label one*
7: D_H^{train} = MLC2HMC(D^{train}, **hierarchy**);
8: D_H^{test} = MLC2HMC(D^{test}, **hierarchy**);
9: *//solve transformed hierarchical multi-label problem by using approach for HMC*
10: HMCModel = HMCMethod(D_H^{train});
11: *//generate HMC predictions*
12: **predictions** = HMCModel(D_H^{test});
13: *//Extract predictions only for the leaves from the HMC predictions*
14: P = ExtractLeavesPredictionsFromHMCPredictions(**predictions**);
15: **return** Evaluate(P)

Random forests as ensemble methods for predictive modeling are originally proposed by Breiman(2001) [1]. The empirical analysis of their use as feature ranking methods has been studied by Verikas et al.(2011) [13].
The usual approach for constructing random forests is to first perform bootstrap sampling on the data and then to build a decision tree for each bootstrap sample.

The decision trees are constructed by taking the best split at each level, from a randomly selected feature subset.

Huynh-Thu et al. [3] propose to use the reduction of the variance in the output space at each test node in the tree (the resulting algorithm was named GENIE3). Namely, the variables that reduce the variance of the output more are, consequently, more important than the ones that reduce the variance less. Hence, for each descriptive variable we measure the reduction of variance it produces when selected as splitting variable. If a variable is never selected as splitting variable (in any of the trees in the ensemble) then its importance will be 0.

The GENIE3 algorithm has been heavily evaluated for single-target regression tasks (e.g., for gene regulatory network reconstruction). The basic idea adopted for future ranking is the same of that proposed in GENIE3, but we use random forest of PCTs for building the ensemble. The result is a future ranking algorithm that works in the context of SOP and is independent of the SOP task considered.

In order to achieve good performance of the HMC methods it is critical to generate label hierarchies that more closely capture the relations among the labels. The only constraint when building the hierarchy is that we should take care about the leaves of the label hierarchies. They need to define the original MLC task. In particular, the labels from the original MLC problem represent the leaves of the label hierarchy, while the labels in internal nodes of the tree are so-called meta-labels. Meta-labels model the potential relations among the original labels.

For deriving the hierarchies we use three different clustering methods (two agglomerative and one divisive):

- agglomerative clustering with single linkage;
- agglomerative clustering with complete linkage and
- predictive clustering trees.

Agglomerative clustering algorithms consider each example as separate cluster at the beginning and then iteratively merge pairs of clusters based on their distance metric (linkage). If we use the maximal distance between the clusters C_1 and C_2, then this type of agglomerative clustering is using *complete* linkage, i.e., $\max\{dist(c_1, c_2) : c_1 \in C_1, c_2 \in C_2\}$. If we use the minimal distance between clusters, then the agglomerative clustering approach is with *single* linkage i.e., $\min\{dist(c_1, c_2) : c_1 \in C_1, c_2 \in C_2\}$.

We also use predictive clustering trees to construct the label hierarchies. More specifically, the setting from the predictive clustering framework we are using in this work is based on equating the target and descriptive space. Descriptive variables are used to provide descriptions for the obtained clusters. Here, the focus is using predictive clustering framework on the task of clustering instead of classification [4],[2].

An obtained hierarchies using agglomerative clustering (single and complete linkage) and using predictive clustering trees are shown in Fig 2, for *emotions* dataset.

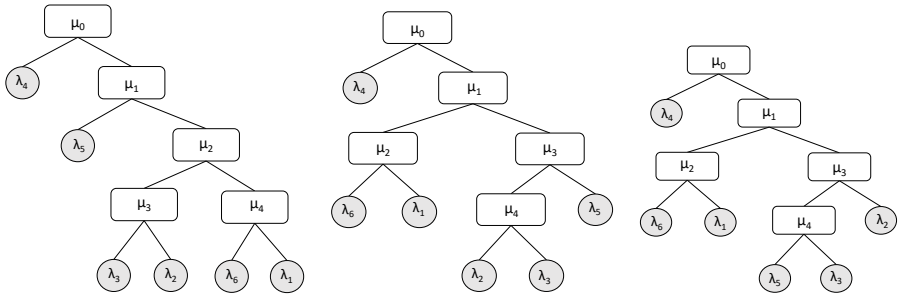

Fig. 2: Hierarchies obtained using agglomerative single (left), agglomerative complete (center) and PCTs (right) clustering methods for *emotions* dataset

PCTs are a generalization of decision trees towards the tasks of predicting structured outputs, including both MLC and HMLC. In order to apply PCTs to the task of HMLC, Vens et al. (2008) [12], define the variance and the prototype as follows. First, the set of labels for each example is represented as a vector of binary components. If the example belongs to the class c_i then the i'th component of the vector is 1 and 0, otherwise.
The variance of a set of examples E is thus defined as follows:

$$Var(E) = \frac{1}{|E|} \cdot \sum_{i=1}^{|E|} dist(\Gamma_i, \overline{\Gamma})^2 \tag{1}$$

where

$$\overline{\Gamma} = \frac{1}{|E|} \cdot \sum_{i=1}^{|E|} \Gamma_i$$

In other words, the variance $Var(E)$ in (1) represents the average squared distance between each example's class vector (Γ_i) and the mean class vector of the set ($\overline{\Gamma}$).
When we talk about HMC, then the similarity at higher levels of the hierarchy are more important than the similarity at lower levels. This is reflected with the distance term used in (1), which is weighted Euclidean distance:

$$dist(\Gamma_1, \Gamma_2) = \sqrt{\sum_{s=1}^{|\Gamma|} \theta(c_s) \cdot (\Gamma_{1,s} - \Gamma_{2,s})^2}$$

()where $\Gamma_{i,s}$ is the s'th component of the class vector Γ_i of the instance E_i, $|\Gamma|$ is the size of the class vector, and the class weights $\theta(c) = \theta_0 \cdot \text{avg}_j\{\theta(p_j(c))\}$, where $p_j(c)$ is j'th parent of the class c and $0 < \theta_0 < 1$. Latter shows that class weights $\theta(c)$ decrease with the depth of the class in the hierarchy.
Random forests of PCTs for HMC are considered in the same way as the random forest of PCTs for MLC. In the case of HMC, the ensemble is a set of PCTs for

HMC. A new example is classified by taking a majority vote from the combined predictions of the member classifiers. The prediction of the random forest ensemble of PCTs for HMC satisfy the hierarchy constraint (if the example is labeled with a given label then is automatically labeled with all its ancestor-labels).

4 Experimental design

The aim of our study is to address the following questions (i) Whether feature ranking on the label (output) space in the MLC task can be used to construct good label hierarchies? (ii) Which clustering method yields better hierarchy? (iii) How this scales from single model to ensemble of models? In order to answer the above questions, we use six multi-label classification benchmark problems from different domains. We have 2 from text classification, 3 from multimedia, includes movie clips and genres classification and 1 from a biological domain. All datasets are predefined by other researchers and divided into train and test subsets. The basic information and statistics about these datasets are given in Table 1.

In our experiments, we use 9 different evaluation measures, as suggested by [7, 11] divided into two groups: five threshold dependent measures: one example based (*hamming loss*), four ranking-based (*one-error, coverage, ranking-loss and average precision*) and four threshold independent measures(*AUROC, AUPRC, wAUPRC* and *pooledAUPRC*). The threshold independent measures are typically used in HMLC and they do not require a (pre)selection of thresholds and calculating a prediction [12]. All of the above measures offer different viewpoints on the results from the experimental evaluation.

Hamming loss is an example-based evaluation measure that evaluate how many times a pair of example and its label are misclassified. *One-error* is a ranking-based evaluation measure that evaluates how many times the top-ranked label does not exist in a set of relevant labels of the example. *Ranking loss* is a ranking-based evaluation measure that evaluates the average fraction of the label pairs that are reversely ordered for the given example. *Average precision* is ranking-based evaluation measure that evaluates the average fraction of labels which are

dataset	*domain*	$\#tr.e$	$\#t.e$	D	L	l_c
emotions	multimedia	391	202	72	6	1.87
scene	multimedia	1211	1159	294	6	1.07
yeast	biology	1500	917	103	14	4.24
medical	text	645	333	1449	45	1.25
enron	text	1123	579	1001	53	3.38
mediamill	multimedia	30993	12914	120	101	4.38

Table 1: Statistics of used benchmark tasks in terms of application domain (*domain*), number of training examples ($\#tr.e$), testing examples ($\#t.e$), number of descriptors (D), total number of labels (L) and number of labels per example.

ranked above a current label and are in the set of true labels. *Coverage*, as a ranking-based evaluation measure, evaluates how far we need to go in a list of ranked labels in order to cover all relevant labels of the given example
Precision and recall are very important measures defined for binary classification tasks with classes of positive and negative examples. Precision is a proportion of positive prediction that are correct, and recall is the proportion of positive examples that correctly predicted as positive. A precision-recall curve (PR curve) is a curve that represent the precision as a function of its recall. $AUPRC$ (area under the PR curve) is the area between the PR curve and the recall axis. $wAUPRC$ evaluates the weighted average of the areas under the individual (per class) PR-curves. If choosing some threshold, we transform the multi-label problem into a binary problems with considering binary classifier as a couple (instance, class) and predicts whether that instance belongs to that class, we can obtain a PR curves that differs depend of the varying the threshold. The area under the average PR curve (from all different threshold curves) is called *pooled*$AUPRC$. From the other side, if we consider the space of true positive rates (sensitivity) versus false positive rates (fall-out) then the curve considers the sensitivity as a function of the fall-out is called ROC-curve. The are under this ROC-curve is an evaluation measure called $AUROC$.
All our experiments are performed using the CLUS software package (`https://sourceforge.net/projects/clus/`), which implements the predictive clustering framework, including PCTs, random forests of PCTs and feature ranking [5],[9]. For the HMC approach, we use different classification thresholds t from 0 to 100 with step 2 in order to evaluate a set of trees in which the predicted label sets are constructed with different thresholds. All labels that are predicted with a probability $\geq t$, are in the predicted set. A hierarchical tree defined by our used clustering methods in HMC setting are defined as a tree shaped hierarchies. For obtaining a hierarchy using PCTs clustering we are using CLUS software and to obtain a hierarchy using agglomerative clustering method we use the R software package (function *agnes()* from the *cluster* package. For more info, see `https://stat.ethz.ch/R-manual/R-devel/library/cluster/html/agnes.html`).
Note that Madjarov et al. [6] also propose as a method for hierarchy construction – balanced k-means clustering. However, the current implementation of this algorithm works only on label co-occurrences i.e., only on binary type of labels, and as such is not directly applicable to our case, where we cluster variables of numerical type. We use Euclidean distance metric in all our algorithms that require distance (agglomerative clustering). Moreover, for random forest for feature ranking we use GENIE3 as a inference method based on variable selection with ensembles of regression trees [3].

5 Results

In this section, we present the obtained results from the experiments we performed using our novel proposed method for decomposing the output space. Table 2 and 3 show the predictive performance obtained using single trees (PCTs)

and ensemble of trees (i.e., Random Forest of PCTs), respectively. We compare the following methods for hierarchy construction:

- flat MLC problem without considering a hierarchy in the label space (*flat_MLC*);
- agglomerative clustering with single linkage (*AggSingle*);
- agglomerative clustering with complete linkage (*AggComplete*) and
- clustering using predictive clustering trees (*ClusterPCTs*).

Observing the results obtained using single PCTs (Table 2), we can see that, in general, *ClusterPCTs* and *AggComplete* methods for construction the hierarchies perform the best in most of the cases. Only in couple of cases, *AggSingle* methods

Table 2: Results of nine performance measures for *PCTs* from experiments performed on six different datasets. The best obtained results per measure are bolded, accordingly.

PCTs	Hamming loss	Average precision	Coverage	One Error	Ranking Loss	AUROC	AUPRC	wAUPRC	pooled AUPRC
ENRON									
flat_MLC	**0.0707**	0.5380	40.5130	0.4439	0.1514	0.1304	0.5846	0.3533	0.4162
AggSingle	0.0710	**0.5946**	39.6304	0.3834	**0.1036**	0.1417	0.5976	0.3665	0.4278
AggComplete	0.0721	0.5648	39.7029	**0.3817**	0.1921	**0.1479**	**0.6008**	**0.3703**	**0.4332**
ClusterPCTs	0.0721	0.5540	**39.4715**	**0.3817**	0.1939	0.1424	0.5901	0.3542	0.4181
EMOTIONS									
flat_MLC	**0.2921**	0.6692	**4.4307**	0.4505	0.3348	**0.5159**	**0.6802**	0.5086	**0.5238**
AggSingle	0.3045	0.6664	4.5693	0.4901	0.3165	0.4866	0.6614	0.4804	0.5026
AggComplete	0.2962	0.6722	4.5743	0.4703	0.3145	0.5073	0.6795	0.5081	0.5174
ClusterPCTs	**0.2921**	**0.7017**	4.5693	**0.3861**	**0.2909**	0.5050	0.6699	**0.5091**	0.5168
MEDICAL									
flat_MLC	**0.0143**	**0.7954**	**11.4474**	**0.2042**	0.1036	0.3210	0.6859	**0.6720**	**0.7016**
AggSingle	0.0151	0.7940	12.8739	0.2162	**0.0825**	0.3198	0.6848	0.6462	0.6819
AggComplete	0.0147	0.7853	12.2072	0.2222	0.1147	**0.3249**	0.6903	0.6647	0.6920
ClusterPCTs	0.0146	0.7869	11.8318	0.2312	0.0866	0.3137	**0.6964**	0.6696	0.6986
MEDIAMILL									
flat_MLC	0.0522	0.4723	77.2816	0.4445	0.2469	**0.0891**	0.5710	0.3392	0.4399
AggSingle	0.0524	0.5841	76.8684	0.3184	0.1054	0.0873	0.5704	0.3504	0.4389
AggComplete	0.0525	**0.6100**	76.7946	**0.3134**	0.0834	0.0886	0.5699	**0.3531**	0.4431
ClusterPCTs	**0.0519**	0.6036	**76.0044**	0.3508	**0.0710**	0.0877	**0.5737**	0.3523	**0.4434**
SCENE									
flat_MLC	0.2631	0.6355	4.5372	0.6860	0.1826	**0.1928**	**0.5297**	0.2551	0.9073
AggSingle	0.2507	0.4911	**4.2149**	0.6694	0.4746	0.1831	0.4791	**0.2815**	0.9033
AggComplete	**0.2466**	**0.6581**	4.5950	**0.6281**	**0.1659**	0.1912	0.4935	0.2650	**0.9073**
ClusterPCTs	**0.2466**	0.4700	4.5950	0.6612	0.5250	0.1912	0.4935	0.2650	**0.9073**
YEAST									
flat_MLC	0.2954	0.6295	11.1243	**0.4297**	0.2990	0.3541	0.5583	0.4829	0.5102
AggSingle	0.2902	0.5902	11.1221	0.5104	0.3673	0.3649	0.5744	0.5002	**0.5277**
AggComplete	**0.2893**	0.6081	**11.1091**	0.5071	0.3205	**0.3685**	**0.5778**	**0.5044**	0.5269
ClusterPCTs	0.2924	**0.6451**	11.2977	0.4547	**0.2574**	0.3580	0.5598	0.4913	0.5253

Table 3: Results of nine performance measures for *Random Forest* from experiments performed on six different datasets. The best obtained results per measure are bolded, accordingly.

Random Forest	Hamming loss	Average precision	Coverage	One Error	Ranking Loss	AUROC	AUPRC	wAUPRC	pooled AUPRC
ENRON									
flat_MLC	0.0466	**0.6976**	13.1865	0.2003	0.0782	0.2409	0.7090	0.6196	0.5767
AggSingle	0.0467	0.6955	**13.0276**	0.2055	0.0772	0.2350	0.7241	0.6153	0.5738
AggComplete	0.0468	0.6949	13.3472	0.2055	0.0781	0.2387	0.7239	0.6178	0.5751
ClusterPCTs	**0.0460**	0.6963	13.1796	**0.1986**	**0.0764**	**0.2437**	**0.7371**	**0.6205**	**0.5820**
EMOTIONS									
flat_MLC	**0.1914**	0.8134	**2.8119**	0.2673	0.1519	0.7550	0.8514	0.7543	0.7548
AggSingle	0.2013	**0.8145**	2.8366	0.2822	0.1547	0.7486	0.8521	0.7557	0.7527
AggComplete	0.1964	0.8104	2.8168	**0.2624**	**0.1506**	**0.7662**	**0.8591**	**0.7623**	**0.7687**
ClusterPCTs	0.2046	0.8141	2.8267	0.2822	0.1541	0.7541	0.8557	0.7561	0.7539
MEDICAL									
flat_MLC	0.0182	0.8582	**2.5706**	**0.3964**	0.0229	0.4316	**0.8235**	0.7873	0.8177
AggSingle	0.0195	0.8564	2.6997	0.4595	0.0242	0.4387	0.8124	0.7640	0.8035
AggComplete	**0.0180**	**0.8653**	2.5886	0.4024	**0.0219**	**0.4576**	0.8201	**0.7902**	**0.8277**
ClusterPCTs	0.0186	0.8492	2.7688	0.4114	0.0259	0.4216	0.8050	0.7773	0.8177
MEDIAMILL									
flat_MLC	**0.0303**	**0.7346**	**20.6763**	**0.1241**	0.0472	**0.2544**	0.7623	0.6708	0.6177
AggSingle	0.0304	0.7334	20.7815	0.1245	**0.0469**	0.2494	0.7647	0.6692	**0.6168**
AggComplete	0.0303	0.7334	20.7270	0.1269	0.0469	0.2541	**0.7654**	0.6683	0.6157
ClusterPCTs	0.0304	0.7337	20.8064	0.1259	0.0469	0.2481	0.7650	0.6684	0.6158
SCENE									
flat_MLC	**0.1694**	**0.6312**	2.4050	0.3388	0.2469	**0.1931**	**0.5152**	**0.4574**	**0.9058**
AggSingle	0.1736	0.6083	2.4959	0.3471	0.2723	0.1865	0.4952	0.4399	0.9038
AggComplete	0.1736	0.6237	**2.3141**	**0.3306**	**0.2339**	0.1893	0.5018	0.4340	0.9046
ClusterPCTs	0.1736	0.6237	**2.3141**	**0.3306**	**0.2339**	0.1893	0.5018	0.4340	0.9046
YEAST									
flat_MLC	**0.1973**	**0.7592**	**7.1756**	**0.2410**	**0.1657**	**0.5085**	**0.7103**	**0.7218**	**0.6751**
AggSingle	0.1992	0.7553	7.3075	**0.2410**	0.1704	0.5012	0.6988	0.7169	0.6694
AggComplete	0.2003	0.7525	7.2694	0.2465	0.1720	0.4997	0.6816	0.7131	0.6652
ClusterPCTs	0.1982	0.7550	7.2519	0.2443	0.1688	0.5038	0.6923	0.7145	0.6691

outperform the others (on enron, scene and medical datasets for the ranking-based metrics) and only in one case (in yeast dataset) for *pooledAUPRC* metric.

Comparing the best performing methods i.e, *AggComplete* and *ClusterPCTs*, we can see that in the enron and the yeast datasets, *AggComplete* give the best performing results for five out of nine performance measures including threshold-independent metrics. Furthermore, on the scene dataset, *AggComplete* method is better than *ClusterPCTs* method on the three ranking-based metrics and performs the same considering Hamming loss and pooledAUPRC. But, considering the mediamill and the emotions datasets, we can see that *ClusterPCTs* method outperform the *AggComplete* method. Latter, gives us an assumption

that maybe, *ClusterPCTs* will perform better than *AggComplete* when bigger and more complex datasets (such as mediamill) are considered.
However, when we observe Table 3, where the results from random forests are shown, we can see a different situation. Namely, on the enron dataset, *ClusterPCTs* is the best method and outperforms *AggComplete* and on the emotions dataset *AggComplete* outperforms *clusterPCTs* in six out of nine metrics cosidereing threshold-independent metrics (AUROC, AUPRC, wAUPRC and pooledAUPRC), which was not the case in the results from the PCTs. Furthermore, both methods perform similar, considering the mediamill and the scene datasets. Considering our best performing methods, *AggComplete* and *ClusterPCTs* versus the original *flat_MLC* method, for PCTs case (Table 2), we can see that the latter is not outperformed only on the emotions and the medical datasets, which are relatively small datasets. This indicates that in bigger datasets the original *flat_MLC* method will be outperformed by *AggComplete* and *ClusterPCTs* methods.
We have a different situation when we consider the random forest results (Table 3). Namely, the original MLC, *flat_MLC* method, outperforms the others on the mediamill, the yeast and the scene datasets according to most of the metrics. Finally, in our study we also considered training errors i.e., the errors made in the learning phase. There, in a large majority of the cases, the original *flat_MLC* method performed the best. This means that other methods we use for constructing the hierarchies (*AggSingle*, *AggComplete* and *ClusterPCTs*) do not overfit as the original one. This is another advantage of methods for construction the hierarchies identified from the obtained results.

6 Conclusions and further work

In this work, we have presented an approach for hierarchy construction and decomposing the output (label) space by using feature ranking. More specifically, we cluster the feature rankings to obtain a hierarchical representation of the potential relations existing among the different labels. We then address the task of MLC as a task of HMLC.
We investigated three clustering methods for hierarchy creation, agglomerative clustering with single and complete linkage, and clustering using predictive clustering trees (PCTs). The resulting problem was then approached as a HMLC problem using PCTs and random forests of PCTs for HMLC. We used six benchmark datasets (enron, emotions, medical, mediamill, scene and yeast) to evaluate the performance.
The results reveal that the best methods for hierarchy construction are agglomerative clustering with complete linkage and clustering using PCTs. Compared to the original MLC method where there is no hierarchy this improves the performance for five datasets with high number of labels. Similar conclusions, but to a lesser extent, can be made for the random forests of PCTs for HMLC - in many of the cases (datasets and evaluation measures) the predictive models exploting the hierarchy of labels yiled better predictive performance. Finally, by

considering the training error performance, we find that original MLC models overfit more than the HMLC models.
For further work, we plan to make more extensive evaluation on more datasets with diverse properties. Next, in our future study, we plan to include a comparison to co-occurrence clustering (including balanced k-means) [6] and network approaches [10]. Also, we plan to extend this approach to other tasks, such as multi-target regression.

Acknowledgment

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) and the project LANDMARK - Land management, assessment, research, knowledge base (H2020 Grant number 635201). SN holds Teagasc Walsh Fellowship.

References

1. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
2. Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S.: Fast and scalable image retrieval using predictive clustering trees. International Conference on Discovery Science pp. 33–48 (2013)
3. Huynh-Thu, V.A., Irrthum, Wehenkel, L., Geurts, P.: Inferring regulatory networks from expression data using tree-based methods. PLos One 5(9), 1–10 (2010)
4. Kocev, D.: Ensembles for predicting structured outputs. Ph.D. thesis, IPS Jožef Stefan, Ljubljana, Slovenia (2011)
5. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recognition 46(3), 817–833 (2013)
6. Madjarov, G., Dimitrovski, I., Gjorgjevikj, D., Džeroski, S.: Evaluation of different data-derived label hierarchies in multi-label classification. International Workshop on New Frontiers in Mining Complex Patterns pp. 19–37 (2014)
7. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recognition 45(9), 3084–3104 (2012)
8. Silla, C.N., Freitas, A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery 22, 31 – 72 (2011)
9. Struyf, J., Džeroski, S.: Constraint Based Induction of Multi-Objective Regression Trees. In: Proc. of the 4th International Workshop on Knowledge Discovery in Inductive Databases KDID - LNCS 3933. pp. 222–233. Springer (2006)
10. Szymanski, P., Kajdanowicz, T., Kersting, K.: How is a data-driven approach better than random choice in label space division for multi-label classification? Entropy 18, 282 (2016)
11. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer (2010)
12. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning 73(2), 185–214 (2008)
13. Verikas, A., Gelzinis, A., Bacauskiene, M.: Mining data with random forests: A survey and results of new tests. Pattern Recognition 44(2), 330–349 (2011)