Multi-label Classification Using Random Label Subset Selections

Martin Breskvar[1,2(✉)], 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
Martin.Breskvar@ijs.si

Abstract. In this work, we address the task of multi-label classification (MLC). There are two main groups of methods addressing the task of MLC: problem transformation and algorithm adaptation. Methods from the former group transform the dataset to simpler local problems and then use off-the-shelf methods to solve them. Methods from the latter group change and adapt existing methods to directly address this task and provide a global solution. There is no consensus on when to apply a given method (local or global) to a given dataset. In this work, we design a method that builds on the strengths of both groups of methods. We propose an ensemble method that constructs global predictive models on randomly selected subsets of labels. More specifically, we extend the random forests of predictive clustering trees (PCTs) to consider random output subspaces. We evaluate the proposed ensemble extension on 13 benchmark datasets. The results give parameter recommendations for the proposed method and show that the method yields models with competitive performance as compared to three competing methods.

Keywords: Multi-label classification · Structured outputs · Output space decomposition · Predictive clustering trees · Ensemble methods

1 Introduction

Supervised learning is a very actively researched area of machine learning. Its goal is to learn models able to provide predictions for previously unseen examples of data. Single-target prediction scenarios are very common and applicable in many domains. However, not all solutions to problems can be *fitted* into one predicted variable. It is very possible that a more complex representation of the data is needed. This is a challenge because it requires methods to predict more than one variable of interest. In that sense, we move towards structured output prediction (SOP) tasks. Examples of SOP tasks are MT regression (MTR), multi-label classification (MLC), time series prediction etc.

This work focuses on solving the MLC task where a given example can be annotated with one or more labels. For instance, a gene could have more than one function, an image can contain different objects, a document can belong to several categories, a disease can manifest with multiple symptoms, etc. This

A. Yamamoto et al. (Eds.): DS 2017, LNAI 10558, pp. 108–115, 2017.
DOI: 10.1007/978-3-319-67786-6_8

particular area of research attracts the attention of the community due to the increasing number of possible applications in various domains (multimedia, biology, medicine, semantic web, legislation,...). Traditional MLC approaches consider individual labels separately, i.e., they are local and transform the dataset into multiple single-label datasets (a dataset for each label) and then solve the multiple single-label tasks with off-the-shelf methods. The key observation here is that such approaches assume that labels are not related: If label relations exist, these approaches are not able to take advantage of their knowledge. Therefore, MLC approaches should be global and exploit potential relations between labels to produce more accurate models.

Notwithstanding, given a dataset, it is not clear which type of method one should use: a local or a global. There is no consensus on this issue [6]. On some datasets, it is preferable to use local, while on other global methods. Having this in mind, we believe that the best method should combine the advantages of both groups. We hence propose a method for MLC that randomly samples the output/label space and learns global models for the sampled label space. Furthermore, we combine the multiple models into an ensemble.

Output space selection and transformation methods already exist in the scope of MLC. One of the most well-known methods is Random k-Labelsets (RAkEL) [8]. It is a problem transformation method as it constructs an ensemble of ST classification models to solve the task of MLC. It does so by selecting random subset of labels (size is determined by the k parameter) for each base model. RAkEL then builds a powerset of the selected subset of labels and trains a ST classification model on it. This approach has been extended towards data-driven partitioning of the label space, which is achieved by using community detection algorithms from social networks [7]: These find better label subspaces as opposed to randomly selecting them. Another data-driven approach uses label hierarchies obtained by hierarchical clustering of flat label sets by using annotations that appear in the training data [5]. Finally, a dimensionality reduction method that uses random forests with Gaussian subspaces has been proposed [3]. This method also belongs to the algorithm adaptation group. It reduces the output space by making random projections of the output space into a new space which represents a highly compressed version of the original label space.

2 MLC Using Random Label Subset Selections

The proposed method is based on the predictive clustering (PC) framework. More specifically, we use predictive clustering trees (PCTs) that can be seen as a generalization of decision trees for the task of structured output prediction. The standard top-down induction of decision tree (TDIDT) algorithm is used to generate PCTs. The pseudo code for the randomized PCT induction algorithm (RPCT) is shown on the left side of Table 1 and it takes the following inputs: (i) a dataset S, (ii) a function $\delta_c(X)$ that randomly samples c descriptive variables from dataset X without replacements and (iii) a set of attributes R_t, that the learning process should use for supervision.

The RPCT algorithm first randomly samples from the pool of all available descriptive attributes for the current dataset. The sampled descriptive attributes, along with the target attributes R_t provided as input, are used to calculate the best possible split point (i.e., the best test) to use for partitioning the data instances. After the best test is found the data are split according to it. This process continues recursively until a stopping criterion is met and the prototype function is invoked. We use a prototype function that returns a vector of probabilities that an example belongs to the positive class for each target variable.

The test selection is handled by the *BestTest* function: It begins by removing the target attributes which should not be considered (Table 1, right, line 2). $\Pi(S, R_d, R_t)$ is a projection function that reduces the original dataset S to S_R by only considering descriptive and target attributes from sets R_d and R_t respectively. All possible tests on S_R are evaluated and the one that reduces the variance the most (w.r.t. S_R) is selected (Table 1, right, lines 3–9). The variance calculation function is also a parameter and can be instantiated based on the type of machine learning task we want to solve. In this paper, we focus on MLC so we calculate the variance as the sum of Gini indices over the individual target variables from the set $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ as $Var(S) = \sum_{i=1}^{q} Gini(S, \lambda_i)$.

Ensembles combine the predictions of multiple predictive models to achieve better predictive performance. Predictions for new examples are made by querying base models and combining their predictions. In this section, we describe the process of generating ensembles, where the base models are not all learned from all available target attributes, but rather each model is learned from a (different) subset of them. For this, we will need the parameter R_t defined above. We named this ensemble method Random Output Selections (ROS).

Regular PCTs use the whole target space to calculate the heuristic score. The proposed ensemble approach introduces random selections in the output

Table 1. The top-down induction of randomized predictive clustering trees

Function $RPCT(S, \delta_c, R_t)$
Out: A predictive clustering tree
1: $R_d \leftarrow \delta_c(S)$
2: $(t^*, h^*, \mathcal{P}^*) \leftarrow$ BestTest(S, R_d, R_t)
3: **if** $t^* \neq none$ **then**
4: **for each** $S_i \in \mathcal{P}^*$ **do**
5: $tree_i \leftarrow$ RPCT(S_i, δ_c, R_t)
6: **end for**
7: **return** node$(t^*, \bigcup_i \{tree_i\})$
8: **else**
9: **return** leaf(Prototype(S))
10: **end if**

Function $BestTest(S, R_d, R_t)$
Out: Selected test t^*
Out: Heuristic score h^* of test t^*
Out: Partitioning $\mathcal{P}^*$ induced by t^* on S
1: $(t^*, h^*, \mathcal{P}^*) \leftarrow (none, 0, \emptyset)$
2: $S_R \leftarrow \Pi(S, R_d, R_t)$
3: **for each** possible test t in S_R **do**
4: $\mathcal{P} \leftarrow$ partitioning induced by t on S_R
5: $h \leftarrow Var(R_t, S_R) - \sum_{S_i \in \mathcal{P}} \frac{|S_i|}{|S_R|} Var(R_t, S_i)$
6: **if** $(h > h^*)$ **then**
7: $(t^*, h^*, \mathcal{P}^*) \leftarrow (t, h, \mathcal{P})$
8: **end if**
9: **end for**
10: **return** $(t^*, h^*, \mathcal{P}^*)$

space, i.e., individual PCTs do not consider the whole target space anymore. Each base model (PCT) is consequently learned from only those targets that were included in the randomly generated partition R_t provided to it by the function Π. The output space partitions are generated before the induction of base models and are independent of the base model learning algorithm. The algorithm for construction of subspaces has the following parameters: (i) the number of base models b, (ii) a function $\theta_v(X)$ that samples uniformly at random without replacement v items from the set X and (iii) a set of target attributes (labels) T. ROS first creates a subspace which considers all target attributes, to make sure that every target attribute is considered by at least one base model. We generate the remaining $b-1$ subspaces with the θ_v function. We build ROS ensembles of PCTs by using the randomized PCT algorithm (RPCT). Each base model is learnt from different bootstrap replicate. Such perturbations of the learning set have been proven useful in cases, where unstable base models, such as decision trees, are used. RPCT introduces additional randomization while learning its individual base models by considering only a subset of descriptive attributes at each step, i.e., when selecting the best test at a given node by calling the function $\delta_c(X)$ just before. In addition, ROS randomly selects a subset of targets for each PCT in the ensemble (we refer to the method as RF-ROS).

Ensembles combine predictions of their base models. In this study, we use two different prediction-combining techniques, i.e., aggregation functions: (i) total averaging (i.e., the most commonly used voting technique) and (ii) subspace averaging. Total averaging combines votes of the individual base models using probability per-target distribution voting for all targets [1]. Subspace averaging does the same, but only the labels considered during learning of the respective base model participate in the voting.

3 Experimental Design

This section presents the experimental questions posed, benchmark datasets, the experimental setup and the evaluation measures used. We designed the experimental evaluation having the following research questions in mind:

1. What is the recommended label subspace size for RF-ROS ensembles?
2. Does it make sense to change the aggregation function, i.e., can subspace averaging improve the predictive performance of RF-ROS models?
3. Considering predictive performance, how do RF-ROS ensembles compare to other competing methods?

We use 13 publicly available benchmark datasets: Emotions, Scene, Yeast, Birds, TMC 2007, Genbase, Medical, Enron, Mediamill, Bibtex, Bookmarks, Corel 5k, and Delicious. The datasets vary in terms of number instances, descriptive and target attributes. More details about the datasets are available at the MULAN repository (http://mulan.sourceforge.net/datasets.html).

To evaluate the performance of the RF-ROS, we generated ensembles with different output space sizes: $v \in (\frac{q}{4}, \frac{q}{2}, \frac{3q}{4}, \sqrt{q}, \log q)$ with q the number of labels.

We also experimented with two aggregation functions: total and subspace averaging. We then compare the performance of RF-ROS with the performance of: (i) Random forests of standard PCTs (RF-PCT) [4], (ii) Random k-Labelsets (RAkEL) models [8] and (iii) Random forests with Gaussian subspaces (RF-Gauss) [3].

RF-PCT and RF-ROS ensembles used 100 PCTs (ensembles are typically saturated at that point) and descriptive space size $v = \lfloor 0.1 \cdot q \rfloor + 1$ [4]. The trees in the ensembles were not pruned [1]. For RAkEL models, the k parameter (size of labelset) was set to $q/2$ and the number of models to $min(2q, 100)$. A support vector machine (SVM) classifier was selected as a learning algorithm within RAkEL, with a linear kernel and a complexity constant $C = 1$. In RF-Gauss, the number of Gaussian subspace components was set to $\log q$. The other RF-Gauss parameters were set to $n_{min} = 1$ and $k = \sqrt{q}$ [3]. The statistical evaluation of the results was performed according to the guidelines of Demšar [2]. All statistical tests on the predictive performance values were conducted at the significance level $\alpha = 0.05$ (using three decimal places).

In order to determine the predictive performance of the induced models, we empirically evaluate them according to 12 different measures that belong to two groups: example and label based measures. The example based measures considered are: hamming loss, accuracy, precision, recall, F1, subset accuracy. The label based measures considered are: micro/macro precision, micro/macro recall, micro/macro F1 [6]. Results in terms of different measures lead to the same conclusions: In order to conserve space, we present only results for the example based measures F1 (more is better) and Hamming loss (less is better) in Table 2.

Table 2. The performance of the considered methods in terms of the example based measures F1 and Hamming loss. DNF (did not finish) denotes algorithms that did not produce results. The numbers in bold denote best performance on a dataset.

	Example based F1					Hamming loss				
				RF-ROS					RF-ROS	
Name	RAkEL	RF-Gauss	RF-PCT	Tot-75	Sub-LOG	RAkEL	RF-Gauss	RF-PCT	Tot-75	Sub-LOG
Emotions	**0.637**	0.534	0.574	0.582	0.588	0.205	0.2	0.197	**0.196**	0.198
Scene	**0.681**	0.413	0.574	0.558	0.591	0.098	0.111	0.09	0.093	**0.088**
Yeast	**0.64**	0.573	0.587	0.583	0.602	0.2	0.199	**0.198**	**0.198**	0.199
Birds	**0.658**	0.51	0.566	0.556	0.579	0.05	0.048	0.044	0.044	**0.043**
TMC 2007	**0.81**	0.992	0.908	0.902	0.926	0.033	**0.001**	0.015	0.016	0.012
Genbase	**0.996**	0.991	0.981	0.981	0.986	**0.001**	**0.001**	0.002	0.002	**0.001**
Medical	**0.789**	0.515	0.673	0.669	0.683	**0.01**	0.016	0.013	0.013	0.012
Enron	**0.562**	0.508	0.527	0.518	0.559	0.049	0.047	0.046	0.046	**0.045**
Mediamill	DNF	0.545	**0.549**	0.547	0.541	DNF	**0.03**	**0.03**	**0.03**	0.032
Bibtex	DNF	0.173	0.211	0.209	**0.305**	DNF	0.014	**0.013**	**0.013**	**0.013**
Bookmarks	DNF	0.2	**0.206**	0.203	0.175	DNF	**0.009**	**0.009**	**0.009**	**0.009**
Corel	DNF	0.018	0.007	0.009	**0.089**	DNF	**0.009**	**0.009**	**0.009**	0.01
Delicious	DNF	**0.237**	0.194	0.193	0.202	DNF	**0.018**	**0.018**	**0.018**	0.021

4 Results

The proposed method has two degrees of freedom: target subspace size and aggregation function. Figure 1 shows the performance of RF-ROS on four datasets with various label and example counts. The plots for each dataset also show the point (total averaging, 100% target space, always the rightmost data point) which represents the performance of the RF-PCT model on that dataset. The results suggest that subspace averaging outperforms total averaging (especially for subset sizes below 50%). Moreover, the two aggregation functions exhibit inverse behavior w.r.t. the target subspace size. Total averaging performs better with larger target subspaces while subset averaging is better for smaller ones. When the target subspace size increases, both variants converge to a performance similar to that of the original RF-PCT method. This behavior is expected because larger subset size leads to larger overlap between the set of all target variables and its subsets.

We also observe that the performance of models with different aggregation functions converges at different rates. Although we observe convergence towards RF-PCT on all datasets, we speculate that the convergence rate is dataset dependent. For instance, on the Delicious dataset, both variants already converge with a target subspace size of 25%. On the Bibtex dataset, this number is a bit higher (50%) and on the Yeast and Scene datasets even higher (75%).

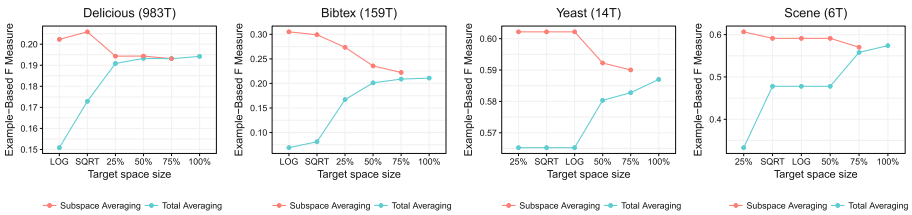
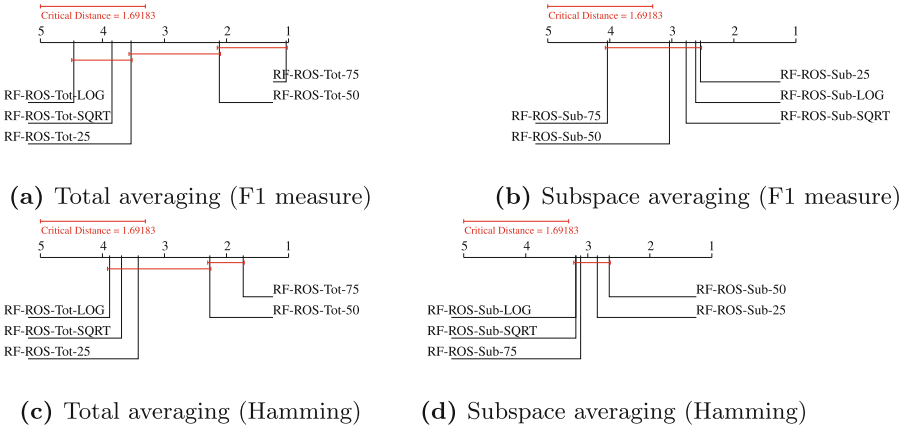


Fig. 1. Example based F1 results for Delicious, Bibtex, Yeast and Scene datasets.

Figure 2 shows average rank diagrams that confirm our speculations. Figures 2a and c show some statistically significant differences, so we recommend a larger subspace size ($v = \frac{3q}{4}$) with total averaging. Figures 2b and d do not show any statistically significant differences between the considered RF-ROS variants. Nevertheless, we recommend using the smallest evaluated subspace size ($v = \log q$) to be used with subspace averaging, as this is most efficient.

We compared the model performances of RF-ROS variants using these recommended parameters to the performance of RF-PCT, RAkEL and RF-Gauss (Fig. 3). The diagrams do not show any statistical significance in terms of F1. It is immediately visible that RAkEL performs very well. Although it did not finish on five datasets, it can still be considered a serious competitor on datasets with smaller label spaces. However, its predictive performance comes at a high



(a) Total averaging (F1 measure) **(b)** Subspace averaging (F1 measure)

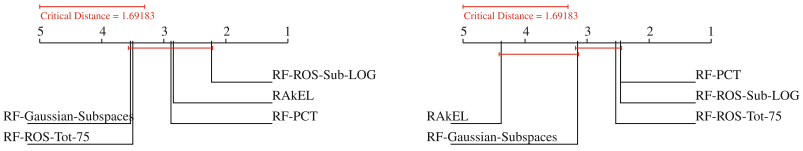
(c) Total averaging (Hamming) **(d)** Subspace averaging (Hamming)

Fig. 2. Average rank diagrams of the RF-ROS variants (F1 and Hamming loss).

computational cost. This method is hindered by the fact that it uses label powersets and SVMs to generate models which makes the running times of RAkEL substantially longer. RAkEL is not a clear winner w.r.t. the average rank diagrams because the method was penalized for not finishing. If we take RAkEL out of consideration, the average rank diagrams in Fig. 3 suggest that the proposed method performs at least as well as the competition.

RF-ROS-Sub-LOG is ranked better than RF-PCT in terms of F1 and equally ranked in terms of Hamming loss. RF-ROS-Tot-75 also performs well in terms of Hamming loss measure but is ranked last w.r.t F1. Moreover, we observe that RF-ROS-Sub-LOG is ranked better than RF-Gauss and RAkEL.

Here, we summarize the answers to our experimental questions. Regarding the recommended label subspace size, RF-ROS should be instantiated with $v = \log q$. It could be beneficial to use a slightly larger subspace size on datasets with larger label spaces (i.e., $v \in (\sqrt{q}, \frac{q}{2})$). Next, subspace averaging should be preferred, because total averaging seems to degrade the predictive performance of the models and (with larger label subspace sizes) converges to the performance of the original method (RF-PCT). Note that even if we do not use the optimal value for the subspace size, the performance of RF-ROS is lower-bounded by



(a) Example based F1 **(b)** Hamming loss

Fig. 3. Average rank diagrams for RF-ROS and its competitors.

RF-PCT. Finally, RF-ROS ensembles perform well compared to the competition, which especially holds for the RF-ROS-Sub-LOG variant.

5 Conclusions and Future Work

We have proposed and evaluated a novel ensemble method for MLC, namedRF-ROS, that uses subsets of the label space to induce base models. We have experimented with different subspace sizes and two voting mechanisms, and found that the proposed method improves random forest models with PCTs as base learners. We have also shown that the proposed method generates models that performs equally well or better than the competition.

Future work is planned that will include evaluation against models generated by additional MLC methods. We will also add experiments on additional datasets. Next, we would like to try a new aggregation function where we would include predictions of the default model (i.e., predictions on the whole training set). We would also like to include out-of-bag errors to estimate the quality of individual base models and use this in conjunction with the mentioned aggregation function. Finally, a possible direction for future work is the extension of label subspace generation process that would work for hierarchies.

Acknowledgements. We acknowledge the financial support of the Slovenian Research Agency via the grants P2-0103,L2-7509, and a young researcher grant to MB, as well as the European Commission, through grants ICT-2013-612944 MAESTRA and ICT-2013-604102 HBP SGA1.

References

1. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Mach. Learn. **36**(1), 105–139 (1999)
2. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
3. Joly, A., Geurts, P., Wehenkel, L.: Random forests with random projections of the output space for high dimensional multi-label classification. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS, vol. 8724, pp. 607–622. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44848-9_39
4. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recogn. **46**(3), 817–833 (2013)
5. Madjarov, G., Gjorgjevikj, D., Dimitrovski, I., Džeroski, S.: The use of data-derived label hierarchies in multi-label classification. J. Intel. Inf. Syst. **47**(1), 57–90 (2016)
6. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recogn. **45**(9), 3084–3104 (2012)
7. Szymański, 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**(8), 282 (2016)
8. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: an ensemble method for multilabel classification. In: Kok, J.N., Koronacki, J., Mantaras, R.L., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS, vol. 4701, pp. 406–417. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74958-5_38