Multi-label feature ranking with ensemble methods

Matej Petković^{1,2} · Sašo Džeroski^{1,2} · Dragi Kocev^{1,2}

Received: 13 July 2019 / Revised: 10 June 2020 / Accepted: 24 August 2020 / Published online: 13 October 2020
© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2020

Abstract

In this paper, we propose three ensemble-based feature ranking scores for multi-label classification (MLC), which is a generalisation of multi-class classification where the classes are not mutually exclusive. Each of the scores (*Symbolic*, *Genie3* and *Random forest*) can be computed from three different ensembles of predictive clustering trees: *Bagging*, *Random forest* and *Extra trees*. We extensively evaluate the proposed scores on 24 benchmark MLC problems, using 15 standard MLC evaluation measures. We determine the ranking quality saturation points in terms of the ensemble sizes, for each ranking-ensemble pair, and show that quality rankings can be computed really efficiently (typically 10 or 50 trees suffice). We also show that the proposed feature rankings are relevant and determine the most appropriate ensemble method for every feature ranking score. We empirically prove that the proposed feature ranking scores outperform current state-of-the-art methods in the quality of the rankings (for the majority of the evaluation measures), and in time efficiency. Finally, we determine the best performing feature ranking scores. Taking into account the quality of the rankings first and—in the case of ties—time efficiency, we identify the *Genie3* feature ranking score as the optimal one.

Keywords Feature ranking · Multi-label classification · Ensemble-based methods · Predictive clustering trees

Editor: Larisa Soldatova, Joaquin Vanschoren.

✉ Matej Petković
matej.petkovic@ijs.si

✉ Dragi Kocev
dragi.kocev@ijs.si
Sašo Džeroski
saso.dzeroski@ijs.si

¹ Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

² Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia

1 Introduction

As opposed to the standard classification problems, where the goal is to learn a model that predicts one of the two or more mutually exclusive predefined class values, e.g., predict whether a given board position leads to a win, draw or loss if both players are playing optimally, multi-label classification (MLC) is a predictive modeling task where the examples can be labeled with more than one (or even zero) of the labels from a predefined set of labels $\mathcal{L}$. In this case, we denote examples as $(\mathbf{x}, \mathbf{y})$, where (1) $\mathbf{x}$ is a vector of values of features x_i that are either numeric (the domain of x_i is a subset of $\mathbb{R}$) or nominal (the domain of x_i is a finite set of values), and (2) $\mathbf{y}$ is a subset of the label set $\mathcal{L}$. The elements of $\mathbf{y}$ are the labels that are *relevant* for a given example.

MLC problems are receiving more and more attention from the research community. For example, one of the use cases of MLC is labeling pictures with objects that appear on them. Due to the abundance of data more and more efficient methods are needed, and this is where feature ranking can play a significant role.

One can always approach a MLC problem by transforming the data and then use standard classification method to learn predictive models. The most typical approach is binary relevance where the MLC problem is divided into a set of $L = |\mathcal{L}|$ standard classification problems, and a separate model for each of the labels is built separately, i.e., a model that predicts whether a given label ℓ is relevant for a given example. In the end, these predictions are joined into a final one (Tsoumakas and Vlahavas 2007). However, building a separate model for each label might be too costly as well as it prevents the predictive model to take into account dependencies among the labels in the learning phase. A similar, yet even more time consuming approach is to build $L(L-1)/2$ predictive models, one for each pair of labels $\ell_1 \neq \ell_2$ (Elkafrawy et al. 2015). In the end, the votes for every label are summed up and the ones with the most votes are predicted as relevant.

Another in the group of problem transformation approaches is the power set approach (Tsoumakas and Vlahavas 2007) where subsets $\mathbf{y}$ are looked upon as the elements of the power set $\mathcal{P}(\mathcal{L})$. It transforms the data into a standard classification problem. A drawback of this approach is that the maximal number of classes 2^L may be really high and the data becomes very sparse.

Another group of methods are the so called method adaptation approaches where an existing method is adapted to directly address MLC problems. This has been done, for example, in the case of predictive clustering trees (Madjarov et al. 2012) and support vector machines (Elisseff and Weston 2001).

The main topic of this paper is the task of *feature ranking*. It is a machine learning task and it is very closely related to predictive modelling. Input to a feature ranking algorithm is a dataset $\mathcal{D}$ and its output are feature relevance scores $relevance(x_i)$, for all features x_i . The relevance score of a feature x_i assesses how dependent are the target values $\mathbf{y}$ on the feature x_i .

By performing a feature ranking, we can explain the predictions of complex black-box models, such as deep neural networks or ensembles. Another motivation for feature ranking is dimensionality reduction, i.e., performing feature selection (Lee and Kim 2017; Sechidis et al. 2014). The task of feature selection can be viewed as a subtask of feature ranking, where we select the most relevant features, and ignore the others when building a predictive model. In that way, the obtained models are easier to understand, faster to build and less prone to overfitting. The selection of the most relevant features is

done by first computing the feature relevances, and then retaining the features with the top k highest scores, for a $k \in \mathbb{N}$.

Approaches to the MLC feature ranking tasks are analogous to those of MLC predictive modelling. We can convert a dataset into L classification datasets (as in binary relevance), compute ranking for each label separately and aggregate feature relevances across the labels to obtain the final ranking. Other data transformation approaches are possible in combination with the many feature ranking methods for classification are applicable (Guyon and Elisseeff 2003): convert the problem into L binary classification tasks, or convert the problem into a multi-class classification task with 2^L classes by using the label power set approach. These two approaches inherit the drawbacks from the MLC task: too high computational cost due to the many binary classification tasks, or sparsity due to the large number of classes in the multi-class classification task. These drawbacks might lead to potentially irrelevant feature rankings.

Method adaptation approaches for MLC feature ranking are divided into three major groups—a heritage from the standard classification scenario (Guyon and Elisseeff 2003). The first group are *filters* and they do not need any predictive model. They are typically computationally very efficient, but can be myopic, i.e., do not make use of the feature dependencies. The second group are *wrappers* and they heavily use predictive models but are very appropriate more or less only for feature selection, since they typically return a feature subset without relevances. Usually, the procedure goes as follows: a candidate feature set is chosen at the beginning. Then, a predictive model that uses only features from this feature set is learned, the feature set is updated according to the predictive performance of the model, and the cycle repeats. The third group of feature ranking algorithms are *embedded* methods where a single predictive model is built and feature relevances are computed directly from it.

The rest of the paper is organized as follows. In Sect. 2 we describe some related work. In Sect. 3, the necessary background for the proposed scores is presented, and in Sect. 4 description of the scores is given. In Sect. 5, the experimental setup is presented (with the listing of experimental questions, evaluation procedure etc). Section 6 we present and discuss the results of the experiments. In Sect. 7, we conclude and give directions for further work.

2 Related work

The datasets in our experiments have hundreds of labels (see Table 2), hence, the data transformation approaches are inappropriate or even infeasible. These approaches include methods addressing the feature ranking through binary relevance, label power set (Spolaôr et al. 2013) and the 1-vs-1 version of the ReliefF algorithm (Kong et al. 2012). Consequently, we will only focus on the method adaptation approaches. Furthermore, as explained above, the methods from the wrapper subgroup of method adaptation methods are also not really appropriate for all feature ranking tasks, so we will review the filter and the embedded approaches and start with the former.

There exist adaptations of information theory based feature ranking scores, such as information gain (Pereira et al. 2015), or feature ranking scores that base on statistical tests, such as χ^2 feature ranking (Spolaôr and Tsoumakas 2013). However, none of them can exploit the feature dependencies, and they all need additional preprocessing of the data, since the numeric features have to be discretized prior to computing the feature ranking.

These drawbacks are addressed by the RReliefF-ML adaptation (Reyes et al. 2015) of the RReliefF algorithm (Kononenko and Robnik-Šikonja 2003) to MLC setting. This is a generalization of the regression version of the Relief algorithm, where only one set of neighbours needs to be computed in every iteration, as opposed to the other versions that base on the Relief for classification and need L (binary relevance), $\mathcal{O}(L^2)$ (the version from Kong et al. 2012) or 2^L groups (power set approach) in the worst case.

The RReliefF-ML algorithm was empirically shown to yield relevant feature rankings (Reyes et al. 2015) by using it on 34 benchmark problems and applying Friedman statistical test. However, the very basic assumption of test (independence of datasets) was not met (Petković et al. 2018). Moreover, it has been shown (Dembczyński et al. 2012) that different MLC evaluation measures result in different optimal classifiers when a predictive modelling problem is being solved, which should also reflect in the field of feature ranking. This was the motivation for parameterizing the distance in the RReliefF-ML algorithm and for proposing three different distance definitions that mimic standard MLC evaluation measures: the original hamming loss distance was compared to the distances that base on the multi-label accuracy, multi-label F_1 measure and subset accuracy. In general, the resulting MLC-Relief algorithm outperformed the original.

The only embedded feature ranking method that we could find is computed from a *Random forest* of predictive clustering trees (PCTs) (Kocev et al. 2013a). There, a *Random forest* of 500 PCTs is built and the ranking is computed out of it as proposed in Breiman (2001). The preliminary study of the method was carried out on the limited number of four benchmark datasets (all four of them are included in our study). In this work, we extend that work in the following ways:

1. In addition to *Random forest*, we propose to use two new ensemble methods: *Bagging* and *Extra trees*
2. In addition to the feature ranking score from Kocev et al. (2013a), we propose two new: *Symbolic* and *Genie3*
3. We extensively evaluate the methods on 24 benchmark datasets
4. We analyze the computational complexity of the proposed feature ranking algorithms, and show that they can be computed much more efficiently than those in Kocev et al. (2013a) (up to 50-times faster).
5. We compare the proposed methods to the non-informed features rankings and state-of-the-art feature ranking methods, and show that we outperform the baselines.

3 Background

3.1 Multi-label classification PCTs

The proposed feature ranking method is based on ensembles of predictive clustering trees (PCTs)—these are an adaptation of the standard decision trees (Breiman et al. 1984) to different types of structured output prediction tasks (Blockeel 1998; Kocev et al. 2013b), including MLC, and also clustering. The induction procedure is given in Algorithm 1 (from Table 1). The input to the method is a set E of training examples, i.e., a training set. The procedure then selects the split test from all candidate splits based on the evaluation of the heuristic score h . The best split is thus chosen to partition the subset E into subsets E_i , an internal node is created (with the selected split), and the method is called recursively on

Table 1 The algorithm for learning predictive clustering trees

Algorithm 1 PCT(E)	Algorithm 2 BestTest(E)
1: $(t^*, h^*, \mathcal{P}^*) = \text{BestTest}(E)$	1: $(t^*, h^*, \mathcal{P}^*) = (\text{none}, 0, \emptyset)$
2: if $t^* \neq \text{none}$ then	2: for each test t do
3: for each $E_i \in \mathcal{P}^*$ do	3: $\mathcal{P} =$ partition induced by t on E
4: $\text{tree}_i = \text{PCT}(E_i)$	4: $h = E \text{impu}(E) - \sum_{E_i \in \mathcal{P}} E_i \text{impu}(E_i)$
5: return $\text{Internal}(t^*, \bigcup_i \{\text{tree}_i\})$	5: if $h > h^*$ then
6: else	6: $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
7: return $\text{Leaf}(\text{Prediction}(E))$	7: return $(t^*, h^*, \mathcal{P}^*)$

Algorithm 1 gives the top-down induction procedure, while Algorithm 2 gives the best test selection procedure

each of the subsets. If no good split exists, a leaf node is created and a prediction for that node is computed.

The adaptation to a given task is completely defined by the impurity measure *impu* used when evaluating the splits (line 5 in Algorithm 2, from Table 1), and prediction function Prediction. The heuristic greedily guides the tree induction since finding the tree with the smallest depth is a NP-hard problem (Hancock et al. 1996).

In the implementation of PCTs (available at <http://source.ijs.si/ktclus/clus-public>), a label set $\mathbf{y}$ is internally represented as its incidence vector $\mathbf{y}$ whose j -th component y_j takes the value 1 when $\ell_j \in \mathbf{y}$ and takes the value 0 otherwise. The impurity of the label ℓ_j for the subset E is then defined as the variance $\text{Var}(E)_j$ of y_j . These values are aggregated in order to obtain the impurity of $\mathbf{y}$ for the subset E as

$$\text{impu}(E) = \sum_{j=1}^L \frac{\text{Var}(E)_j}{\text{Var}(\mathcal{D}_{\text{TRAIN}})_j},$$

i.e., as the average of the normalized variances of the labels.

The prediction $\hat{\mathbf{y}}$ for a subset E that belongs to a leaf is obtained by first computing the average incidence vector

$$\bar{\mathbf{y}} = \frac{1}{|E|} \sum_{(\mathbf{x}, \mathbf{y}) \in E} \mathbf{y},$$

which gives us the estimates of probabilities $\bar{y}_j = P(\ell_j \in \mathbf{y} \mid \mathbf{x})$, and then predicting $\ell_j \in \mathbf{y}$ if and only if $\bar{y}_j \geq 1/2$.

3.2 Ensembles of PCTs

An ensemble is a set of base predictive models. Its prediction for an example $\mathbf{x}$ is made by combining the predictions of the ensemble base models. In the case of MLC and ensembles of PCTs, this is typically done by first averaging the probabilities that single trees give, and then applying the threshold of 1/2 as in the single tree case.

The motivation for introducing an ensemble is that it can be viewed as more stable versions of its base models, since—if the members are diverse models (Hansen and Salamon 1990)—the ensemble predictions have lower variance and are therefore more accurate. Even though we do not use the ensembles as predictive models and rather just compute feature ranking scores out of them, the same motivation still holds. Since every tree is built

independently of the others we define the ensemble feature ranking scores as the averages of the scores over the trees in the ensemble.

To introduce some diversity into the tree induction, one has to modify the completely deterministic tree induction algorithm (Algorithm 1). There are several ways to do so and we use three of them.

Bagging and random forest Instead of being learned from the whole training set $\mathcal{D}_{\text{TRAIN}}$, each tree in the *Bagging*/*Random forest* ensemble is built from a different bootstrap replicate $\mathcal{B}$ of $\mathcal{D}_{\text{TRAIN}}$, called bag. The examples $\mathcal{D}_{\text{TRAIN}} \setminus \mathcal{B}$ are called out-of-bag examples (OOB). Additionally, instead of evaluating all possible tests in the *BestTest* procedure (Algorithm 2), only the tests that a random subset of features yield, are evaluated. The number of the retained features f is a parameter to the algorithm. Its typical values are $\lfloor \log_2 F \rfloor$ or $\lceil \sqrt{F} \rceil$, etc. If $f = F$, i.e. we keep all the features, we obtain the *Bagging* procedure, and *Random forest* procedure otherwise.

Extra trees ensembles As in *Random forest*, we again consider f features in each node, but we do not evaluate all the corresponding tests. Rather, we choose randomly only one test per feature and choose the best one among these f tests. From the bias-variance point of view, the rationale behind the Extra-Trees method is that the explicit randomization of the cut-point and feature combined with ensemble averaging should be able to reduce variance more strongly than the weaker randomization schemes used by other methods (Geurts et al. 2006). Note that, however, Extra-Tree ensembles do not use bootstrapping.

4 Feature ranking scores

We first propose and describe the *Symbolic* score. Then, we proceed explaining the *Genie3* (Huynh-Thu et al. 2010) and the *Random forest score* (Breiman 2001). In the following, a tree is denoted by $\mathcal{T}$, whereas $\mathcal{N} \in \mathcal{T}$ denotes a node. Trees form an ensemble $\mathcal{E}$ of the size $|\mathcal{E}|$. The set of all internal nodes of a tree $\mathcal{T}$ in which the feature x_i appears as part of a test is denoted as $\mathcal{T}(x_i)$.

Symbolic score The main motivation for this score is that the *relevance*(x_i) should be proportional to the number of occurrence of the feature x_i in the splits in the trees. However, tests closer to the root influence more examples, hence the depth of the nodes should be also taken into account. The relevance of feature x_i as defined by the *Symbolic* score is thus given as

$$\text{relevance}_{\text{SYMB}}(x_i) = \frac{1}{|\mathcal{E}|} \sum_{\mathcal{T} \in \mathcal{E}} \sum_{\mathcal{N} \in \mathcal{T}(x_i)} \alpha^{\text{depth}(\mathcal{N})}, \quad (1)$$

where the value of the parameter $\alpha \in (0, 1]$ controls how quickly the influence of a node decreases with its depth. Setting it to $\alpha = 1$ results in simple counting of the appearances of the features in the tree nodes.

Genie3 score The main motivation for this score is that a relevant feature x_i highly reduces the impurity of the target variable if it appears in the split in the tree. The *Genie3* relevance of the feature x_i is thus defined as

$$\text{relevance}_{\text{GENIE3}}(x_i) = \frac{1}{|\mathcal{E}|} \sum_{\mathcal{T} \in \mathcal{E}} \sum_{\mathcal{N} \in \mathcal{T}(x_i)} |E(\mathcal{N})| h^*(\mathcal{N}), \quad (2)$$

where $E(\mathcal{N})$ is the set of examples that come to the node $\mathcal{N}$, and $h^*(\mathcal{N})$ is the heuristic value of the split, i.e., the variance reduction, as described in the Algorithm 2. Additionally, greater emphasis is again put on the nodes closer to the root that are reached by higher number of examples $E(\mathcal{N})$.

Random forest (RF) score This score tests how much noising a feature decreases the predictive performance of the trees in the ensemble. The greater the performance degradation, the more important the feature is.

Once a tree $\mathcal{T}$ in an ensemble is grown, it typically overfits to the dataset it was built for, since no pruning is used in the considered ensemble methods. Thus, the performance of the tree has to be assessed by using out-of-bag examples $\text{OOB}_{\mathcal{T}}$ that were not part of the tree induction. This results in the error $\text{Err}(\text{OOB}_{\mathcal{T}})$ as measured with Hamming loss. Afterward, the values of feature x_i in the set $\text{OOB}_{\mathcal{T}}$ are randomly permuted. We denote the perturbed OOB set by $\text{OOB}_{\mathcal{T}}^i$, and the corresponding predictive error by $\text{Err}(\text{OOB}_{\mathcal{T}}^i)$. The relevance of the feature x_i as the relative increase of the error after noising, averaged over the trees in the ensemble, namely

$$\text{relevance}_{\text{RF}}(x_i) = \frac{1}{|\mathcal{E}|} \sum_{\mathcal{T} \in \mathcal{E}} \frac{\text{Err}(\text{OOB}_{\mathcal{T}}^i) - \text{Err}(\text{OOB}_{\mathcal{T}})}{\text{Err}(\text{OOB}_{\mathcal{T}})}. \quad (3)$$

It is evident that this feature ranking score cannot be computed from an ensemble of *Extra trees*, since they do not use bootstrapping. Note also that due to the possibly numerous permutations of the OOB datasets (whose size is on average more than one third of $\mathcal{D}_{\text{TRAIN}}$) and sending the examples through the trees twice, the time complexity of computing the ranking is not negligible compared to that of building the tree, which is the case for the other two feature ranking scores. However, the algorithm can be made more efficient if we note that $\text{Err}(\text{OOB}_{\mathcal{T}}^i) = \text{Err}(\text{OOB}_{\mathcal{T}})$ for the features x_i that do not appear in $\mathcal{T}$,

Finally, while the first two scores are inherently related to trees, the permutation mechanism of the *Random forest* score can also be used with the other learners. However, building an ensemble of decision trees using *Random forest* or *Bagging* is very appropriate since (1) these two ensembles use bootstrapping, and (2) the time complexity of the trees for making predictions is low.

4.1 Theoretical aspects

Here, we first give the time complexity of the proposed methods, and then the convergence properties of the scores with increasing number of trees.

Under the relaxed balancedness assumption of the trees that a tree has depth $\mathcal{O}(\log n)$, we can quickly derive (by counting the number of operations in the nodes) the following time complexities. The time complexity for inducing *Bagging* is $\mathcal{O}(|\mathcal{E}|Fln \log^2 n)$, where $|\mathcal{E}|$ is the size of the ensemble, F is the number of features, l is the number of labels, n is the number of examples.

Next, we give the following observation for computing the value of $\text{relevance}(x_i)$ (using any ensemble method).

Proposition 1 *Variance of the random variable $R_i = \text{relevance}(x_i)$ decreases linearly with the number of trees in an ensemble.*

Proof Let $R_{i,t}$ be the relevance of some fixed variable x_i as computed from the tree t . Note that the Eqs. (1)–(3) say that $R_i = \frac{1}{|\mathcal{E}|} \sum_{t=1}^{|\mathcal{E}|} R_{i,t}$. Moreover, $R_{i,t}$ are independent and identically distributed variables, thus $\text{Var}(R_i) = \text{Var}R_{i,1}/|\mathcal{E}|$. $\square$

Corollary 1 *With probability one, the order of features with respect to their ranking converges to some final ranking, when $\mathcal{E} \nearrow \infty$.*

Proof The law of large numbers assures that the distribution of R_i variables converges to the Dirac delta function, positioned at $\mathbb{E}[R_i]$. The fact that the expected values are all different equal with probability one, concludes the proof. $\square$

5 Experimental setup

5.1 Experimental questions

We perform experiments to answer the following questions:

1. For a given ranking-ensemble pair, how many trees are needed to reach the saturation point in the quality of the ranking?
2. Are the obtained feature rankings relevant, i.e., are they better than a non-informed ranking that considers all features equally important?
3. What is the most suitable ensemble method for a given feature ranking score?
4. Do the proposed feature ranking scores outperform state-of-the-art baseline?
5. Which feature ranking score is the best?

As state-of-the-art baseline, we take the MLC-Relief method.

5.2 Datasets

We use the same 24 MLC benchmark problems that were used in Petković et al. (2018). Table 2 presents the basic statistics of the datasets. The number of features ranges from 72 to 52,350. The features are numeric and nominal. The label set size L ranges from 6 to 983, while the number of training examples ranges from 322 up to 70,000. The average number of labels per example (in $\mathcal{D}_{\text{TRAIN}} \cup \mathcal{D}_{\text{TEST}}$), i.e., *label cardinality* is also given. With the exception of *Delicious* dataset, it ranges between 1.0 and 4.38.

The datasets come from different domains. *Arts*, *Business*, *Computers*, *Education*, *Entertainment*, *Health*, *Recreation*, *Reference*, *Science*, *Social* and *Society* describe the problems of finding relevant subtopics of the given main topic of a web page. *Bibtex* and *Bookmarks* are automatic tag suggestion problems, *Birds* deals with predictions of multiple bird species in a noisy environment. *Corel5k* contains Corel images. *Delicious* contains contextual data about web pages along with their tags. *Emotions* deals with emotions in music. *Enron* contains data about emails. *Genbase* and *Yeast* come from biological domain. *Mediamill* was introduced in a video annotation challenge. *Medical* comes from Medical Natural Language Processing Challenge. *Scene* deals with labelling of natural scenes. *TMC2007-500* is about discovering anomalies in text reports.

Table 2 Data characteristics: sizes of the train and test parts of the dataset, number of features F , labelset size L and label cardinality ℓ_c

dataset	$ \mathcal{D}_{\text{TRAIN}} $	$ \mathcal{D}_{\text{TEST}} $	F	L	ℓ_c
Arts (Ueda and Saito 2003)	3712	3772	23,146	26	1.65
Bibtex (Katakis et al. 2008)	4880	2515	1836	159	2.40
Birds (Briggs et al. 2013)	322	323	260	19	1.01
Bookmarks (Katakis et al. 2008)	70,000	17,856	2150	208	2.04
Business (Ueda and Saito 2003)	5710	5504	21,924	30	1.60
Computers (Ueda and Saito 2003)	6270	6174	34,096	33	1.51
Corel5k (Duygulu et al. 2002)	4500	500	499	374	3.52
Delicious (Tsoumakas et al. 2008)	12,920	3185	500	983	19.02
Education (Ueda and Saito 2003)	6030	6000	27,534	33	1.46
Emotions (Trochidis et al. 2008)	391	202	72	6	1.87
Enron (UC Berkeley 2018)	1123	579	1001	53	3.38
Entertainment (Ueda and Saito 2003)	6356	6374	32,001	21	1.41
Genbase (Diplaris et al. 2005)	463	199	1185	27	1.25
Health (Ueda and Saito 2003)	4557	4648	30,605	32	1.64
Mediamill (Snoek et al. 2006)	30,993	12,914	120	101	4.38
Medical (Pestian et al. 2007)	645	333	1449	45	1.25
Recreation (Ueda and Saito 2003)	6471	6357	30,324	22	1.43
Reference (Ueda and Saito 2003)	4027	4000	39,679	33	1.17
Scene (Boutell et al. 2004)	1211	1196	294	6	1.07
Science (Ueda and Saito 2003)	3214	3214	37,187	40	1.45
Social (Ueda and Saito 2003)	6037	6074	52,350	39	1.28
Society (Ueda and Saito 2003)	7273	7239	31,802	27	1.67
TMC2007-500 (Srivastava and Zane-Ulman 2005)	21,519	7077	500	22	2.22
Yeast (Elisseff and Weston 2001)	1500	917	103	14	4.24

5.3 Evaluation methodology

We adopted the evaluation methodology that has been previously used in MLC context (Reyes et al. 2015) and in the other types of structured output prediction (Petković et al. 2019).

We use the same train-test split of the datasets as in the MULAN repository <http://mulan.sourceforge.net/datasets-mlc.html>. A ranking is computed from the training part $\mathcal{D}_{\text{TRAIN}}$ only, and evaluated on the testing part $\mathcal{D}_{\text{TEST}}$.

The quality of the ranking is assessed by using the k NN algorithm in which as a distance measure the weighted Euclidean distance was used. For two input vectors $\mathbf{x}^1$ and $\mathbf{x}^2$, the distance between them is defined as

$$d(\mathbf{x}^1, \mathbf{x}^2) = \sqrt{\sum_{i=1}^F w_i d_i^2(\mathbf{x}_i^1, \mathbf{x}_i^2)}, \quad (4)$$

where d_i is defined as the absolute difference of the feature values scaled to the $[0, 1]$ -interval, if x_i is numeric, and as $\mathbf{1}[x_i^1 \neq x_i^2]$ ($\mathbf{1}$ is the indicator function), if x_i is nominal. The weights are set to $w_i = \max\{\text{relevance}(x_i), 0\}$, since they need to be made non-negative to

ensure that d is well defined, and also to ignore the attributes that have smaller values for importance than a randomly generated attribute would have.

The evaluation through a k NN predictive model was chosen because of three main reasons. First, this is a distance based model, hence, it can easily make use of the information contained in the feature importances in the learning phase. The second reason is k NN's simplicity: its only parameter is the number of neighbors and we set its value to 15 (based on initial experiments in Petković et al. 2018 and the experiments presented in Spyromitros et al. 2008). In the prediction stage, the neighbors' contributions to the predicted value are equally weighted, so we do not introduce additional parameters that would influence the performance. The third reason for using k NN as an evaluation model is as follows. If a feature ranking is meaningful, then when the feature importances are used as weights in the calculation of the distances k NN should produce better predictions as compared to k NN without using these weights (Wettschereck 1994).

5.4 Evaluation measures

We evaluate our feature ranking scores using 15 standard MLC evaluation measures (Madjarov et al. 2012; Vens et al. 2008). These are

- average area under recision-recall curve ($\overline{\text{AUPRC}}$), area under the average precision-recall curve (AU PRC), average area under the ROC curve ($\overline{\text{AUROC}}$),
- micro precision, micro recall, micro F_1 measure,
- multi-label precision, multi-label recall, multi-label F_1 measure, multi-label accuracy,
- Hamming loss, subset accuracy,
- one error, coverage, ranking loss.

We avoided the use of macro-averaged measures and average precision since they are ill-defined in some cases that we also encountered in our evaluation of the results. For example, these cases concern datasets containing examples $(\mathbf{x}, \mathbf{y})$ whose label set $\mathbf{y}$ is empty, thus calculating average precision leads to 0/0 expressions. Additionally, the problems might occur when a given label is never predicted as relevant. All in all, the datasets where such ill-defined cases occur are Business, Corel5k, Delicious, Genbase, Health, Medical, Reference (macro recall); Birds, Delicious and Mediamill (average precision); and all but Emotions and Scene (macro precision and macro F_1).

5.5 Statistical analysis of the results

For comparing the algorithms, we use the Friedman test. The null hypothesis H_0 is that all considered algorithms have the same performance. If it is rejected by the Friedman's test, we additionally apply Nemenyi or Bonferroni-Dunn post-hoc test. The first is used when we investigate where the statistically significant differences between *any* two algorithms occur, while the second is used when we are interested in the differences between one particular algorithm and the others. A detailed description of all tests is available in Demšar (2006).

The results of the Nemenyi and Bonferroni-Dunn tests are presented on critical distance diagrams. Each diagram shows the average rank of the algorithm over the considered datasets. Additionally, the groups of algorithms among which no statistically significant differences occur are connected with a line.

Before proceeding with the statistical analysis, we round the performances to three decimal points. In the analysis, the significance level was set to $\alpha = 0.05$.

Sometimes, it is necessary to control the false discovery rate. We do this with the Benjamini-Hochberg procedure (Benjamini and Hochberg 1995).

5.6 Parameter instantiation

We set the number of features that are considered at each node to $f = \lceil \sqrt{F} \rceil$ (Kocev et al. 2013b) when building a *Random forest*, and to $f = F$ when building an ensemble of *Extra trees* (Kocev and Ceci 2015). The remaining parameter that completely defines tree induction is the minimal number of instances in a leaf which was set to 2. We build ensembles of sizes $|\mathcal{E}| \in \{10, 25, 50, 75, 100, 150, 200, 250\}$. The proposed feature ranking scores themselves are parameter-less, except for the *Symbolic* ranking where the value of the parameter α is set to 0.5 as recommended in Petković et al. (2019).

We set the parameters of the MLC-Relief algorithm to their recommended values (Petković et al. 2018): the number of iterations m is set to $|\mathcal{D}_{\text{TRAIN}}|/4$, the number of the Relief neighbours K is set to $K = 15$, and the distance between the label sets $\mathbf{y}^1$ and $\mathbf{y}^2$ in the target space is set to the F_1 distance

$$d_{F_1}(\mathbf{y}^1, \mathbf{y}^2) = 1 - 2|\mathbf{y}^1 \cap \mathbf{y}^2| / (|\mathbf{y}^1| + |\mathbf{y}^2|). \quad (5)$$

6 Results and discussion

Due to the large number of datasets and evaluation measures, we will use the *Arts* dataset and the evaluation measures $\overline{\text{AU PRC}}$ and *Ranking Loss* as a running example, for which we show the detailed graphs. The results for the other datasets and measures will be shown in an aggregated form. Their extended version is available at <https://github.com/Petkomat/mlc-ranking>.

6.1 How many trees?

For each dataset, ranking-ensemble pair and evaluation measure M , we can define the curves that consist of the points $(e, M(e))$ where e ranges over the possible values of ensemble sizes given in Sect. 5.6. The curves for the *Arts* dataset that belong to feature rankings computed from the *Bagging* ensemble and evaluated in terms of $\overline{\text{AU PRC}}$ and *Ranking Loss* are given in Fig. 1. Since $\overline{\text{AU PRC}}$ should be maximized and *Ranking Loss* minimized, *Symbolic* score performs best in both cases (except for the 10-trees rankings, evaluated with *Ranking Loss*), and *Random forest* score performs worst. Furthermore, it is expected that more trees contribute to a more stable and better feature relevance estimates (the variances of the *relevance* estimates (x_i) is inversely proportional to the number of trees in an ensemble), and the curves are not monotonic and not very steep (except for the *Symbolic* ranking). Consequently, the graphs hint that a small number of trees suffices to achieve the saturation point of the ranking quality. Essentially, we are looking to find the minimal number of trees that need to be included into the ensemble so that the performance of the learned feature rankings is not statistically significantly different compared to the best feature ranking.

Fig. 1 Quality of the feature rankings computed from the three scores and *Bagging* ensembles, on the *Arts* dataset when the ensemble sizes varies. The quality is measured in terms of AU PRC (a) and *Ranking Loss* (b)

To further investigate that, we fix the ranking-ensemble pair and evaluation measure while the ensemble size varies. We compare the differences among the corresponding rankings over all the datasets by applying Friedman statistical test. Since there are 8 ranking-ensemble pairs and 15 evaluation measures, this results in 120 tests. However, these tests are not independent, since some of them base on the same rankings. Thus, we control the false discovery rate by the Benjamini-Hochberg procedure (Benjamini and Hochberg 1995). The null hypothesis that all feature rankings have the same quality is rejected only in 21 cases. This essentially implies that having ensembles with as little as 10 base predictive models already yield good feature ranking. The results from the statistical tests are available at the repository <https://github.com/Petkomat/mlc-ranking>.

Difference in the feature rankings was discovered (i.e., the null hypothesis is rejected) for (1) 14 out of 15 evaluation measures for the *Symbolic-Random forest* pair (the exception is one error), and (2) 7 out of 14 evaluation measures for the *Symbolic-Bagging* pair. This means that the other feature two feature ranking scores (Genie3 and Random forest) achieve their saturation points much earlier, i.e., 10 trees in an ensemble suffice. The reason that *Symbolic* score needs more trees may be its simplicity, since the actual quality of the nodes that contain a given feature in their tests is ignored when computing this score.

We next use Nemenyi post-hoc test to discover the optimal ensemble size for the remaining two combinations of the *Symbolic* ranking where 10 trees in the ensemble do not suffice: the first with *Random forest* and the second with *Bagging* ensemble. The post-hoc test reveals the groups of the algorithms with statistically significantly different performance. The results of the tests are given in Fig. 2a (*Random forest*) and b (*Bagging*): These are obtained by evaluating the feature ranking with the multi-label F_1 measure, however, the same conclusions can be drawn by inspecting the other measures (available at the repository <https://github.com/Petkomat/mlc-ranking>). The best performing group of the algorithms (denoted by the left-most red line) is similar in both cases: It contains the feature rankings that were computed from at least 50 trees which means that 50 trees will suffice to achieve the optimal performance.

In sum, we select ensembles with 10 trees as the optimal setting for the majority of ranking-ensemble pairs, except for the *Symbolic-Random forest* and *Symbolic-Bagging* – for these two we learn ensembles with 50 trees. In the analysis of the results presented in the next sections, we will use these values. Note that the selected values are conservative

Fig. 2 Average rank diagrams for the Symbolic score computed from *Random forest* (a) and *Bagging* ensembles (b) and evaluated through the Nemenyi statistical test using multi-label F_1 as the evaluation measure

for the performance of the proposed methods in terms of predictive performance, however, it is optimal in a sense of predictive performance—efficiency trade-off.

6.2 Are the feature rankings relevant?

Now that we have fix ensemble size for every ranking-ensemble pair, we proceed to the experiments that will reveal whether the proposed feature rankings outperform the non-informed baseline ranking which awards every feature relevance 1. The detailed results are again given for the *Arts* dataset on which the rankings are evaluated with AU PRC and *Ranking Loss* (Fig. 3a, b respectively). In addition to the points that present ranking-ensemble pairs, the non-informed baseline ranking is presented with black circles. The column that belongs to the *Extra trees* have one point less since *Random forest* score cannot be computed from those ensembles. One can see that feature rankings always outperform the baseline for this dataset, however, the difference between the baseline and the *Random forest-Random forest* ranking as measured in terms of *Ranking Loss* is not practically significant.

To check whether the baseline is consistently outperformed, we fix an evaluation measure and perform 8 pairwise comparisons of rankings to the baseline. At the end, we apply Benjamini-Hochberg correction of the p-values and obtain the following results.

Fig. 3 Comparison of the proposed scores to the non-informed baseline and the state-of-the-art baseline (MLC-Relief), in terms of AU PRC (a, higher is better) and *Ranking Loss* (b, lower is better) evaluation measures. The legend applies to both graphs

For the measures Hamming Loss, One Error and Micro Precision, no statistically significant differences were found. As for the remaining 12 evaluation measures, 7 out of 8 feature rankings outperform the baseline—the exception is the *Symbolic* feature score, computed from *Extra trees* ensembles. This means that every score results in relevant rankings, provided it is computed from an appropriate ensemble of trees.

6.3 Most appropriate ensemble methods

To find the most appropriate ensemble method for a given feature ranking score, we fix the feature ranking score and vary the ensemble method from which this score is computed. We use the Friedman test in the case of the *Symbolic* and *Genie3* scores since they can be computed from three different ensembles, and Wilcoxon test for *Random forest* score. This results in 45 comparisons, one for each score and measure.

The differences are typically not statistically significant (in 30 cases). The *Symbolic* score is the only one for which the differences are mostly significant. However, for the *Symbolic* score, the post-hoc Nemenyi tests do not select a single ensemble method as the best performing. More specifically, *Extra trees* have the worst average ranks according to two evaluation measures: *multi-label precision* (Fig. 4a) and *Ranking Loss* (Fig. 4b). The other evaluation measures did not yield statistically significant differences across the different ensemble types.

The differences among the different ensemble methods in the case of the *Genie3* ranking are statistically significant for micro-averaged F_1 and AU PRC. For both evaluation measures, the Nemenyi post-hoc test reveals that the *Random forest* ensembles have the worst average rank, hence, they should be avoided in combination with *Genie3*.

In the case of the *Random forest* score, the differences are statistically significant for the three area under curves measures and *Ranking Loss*: Wilcoxon test reveals that *Bagging* should be preferred over the *Random forest* ensemble.

We next investigate the results where the differences are not statistically significant. We summarize these results (in Table 3) by providing the ensemble method with the best average rank for each feature ranking score and for each evaluation measure. The results reveal that the ensemble method that should be used as a default choice are *Extra trees* for the *Genie3* score, *Bagging* for the *Random forest* score, and *Random forest* for the *Symbolic* score.

Fig. 4 The average ranks of *Symbolic* feature rankings computed from different ensemble methods. The statistical significance of differences is assessed by the Nemenyi test. The quality of the ranking is measured in terms of multi-label precision (a) and *Ranking Loss* (b)

Table 3 The type of ensemble for which the produced ranking has the best average rank, for each feature ranking score, and each evaluation measure

	Genie3	Random forest	Symbolic
Hamming loss	<i>Extra trees</i>	<i>Bagging</i>	<i>Random forest</i>
ranking loss	<i>Extra trees</i>	<i>Bagging</i>	<i>Random forest</i>
one error	<i>Extra trees</i>	<i>Bagging</i>	<i>Random forest</i>
coverage	<i>Extra trees</i>	<i>Bagging</i>	<i>Random forest</i>
ML accuracy	<i>Extra trees</i>	<i>Bagging</i>	<i>Bagging</i>
ML precision	<i>Extra trees</i>	<i>Bagging</i>	<i>Bagging</i>
ML recall	<i>Extra trees</i>	<i>Bagging</i>	<i>Bagging</i>
ML F_1 Measure	<i>Extra trees</i>	<i>Bagging</i>	<i>Bagging</i>
subset accuracy	<i>Bagging</i>	<i>Bagging</i>	<i>Bagging</i>
micro precision	<i>Random forest</i>	<i>Bagging</i>	<i>Random forest</i>
micro recall	<i>Extra trees</i>	<i>Bagging</i>	<i>Bagging</i>
micro F_1	<i>Extra trees</i>	<i>Bagging</i>	<i>Bagging</i>
AUROC	<i>Extra trees</i>	<i>Bagging</i>	<i>Random forest</i>
AUPRC	<i>Bagging</i>	<i>Bagging</i>	<i>Random forest</i>
AU PRC	<i>Extra trees</i>	<i>Bagging</i>	<i>Random forest</i>

The abbreviation ML in the evaluation measure names stand for multi-label

6.4 Are feature rankings state-of-the-art?

To answer this question, we compare the proposed feature ranking scores (computed with the most appropriate ensemble method) to the MLC-Relief baseline. The results for the *Arts* dataset are shown in Fig. 3 where in addition to the colored circles that belong to the proposed scores, black crosses that belong to the MLC-Relief score are shown. We can see that the proposed scores outperform the MLC-Relief score on this dataset.

To check whether this is valid in general, we use the Friedman statistical test. Results show that the null hypothesis is rejected for all 15 evaluation measures, so we proceed to the post-hoc test, but now, instead of Nemenyi one, Bonferroni-Dunn test is applied since we are only interested in the comparisons to MLC-Relief.

We show the detailed results for the measures AU PRC (Fig. 5a) and *Ranking Loss* (Fig. 5b). These results show that Random forest ranking is not statistically significantly different from the MLC-Relief (according to AU PRC) or MLC-Relief is statistically significantly worse than all proposed algorithms (according to *Ranking Loss*).

Fig. 5 Average rank diagrams of the proposed ensemble-based scores, compared to the MLC-Relief baseline. Statistically significant differences are assessed by the Bonferroni-Dunn test. The quality of the rankings is measured in terms of AU PRC (a) and *Ranking Loss* (b)

For the other measures, we show only the average ranks diagrams in the form of radar plots where more than one measure can be shown in one graph, see Fig. 6a–d. The closer the algorithm is to the center of the graph, i.e., rank 1, the better. We can see that MLC-Relief is mostly ranked last, except for the multi-label measures in Fig. 6d, micro recall and F_1 measure and subset accuracy when it is ranked second but last.

Thus, we conclude that our methods are state of the art. Not only that ensemble-based methods have always the best two average ranks. Their computational complexity is also better than the computational complexity of MLC-Relief (recall the time complexities of the proposed methods from Sect. 4.1). The time complexity of MLC-Relief is $\mathcal{O}(Fn^2 + nKL)$ where, additionally, K is the number of MLC-Relief neighbors. This means that MLC-Relief is quadratic in the number of examples, which might be prohibitively large, for bigger data sets.

6.5 Which feature ranking score is the best?

To see which of the proposed ranking scores is the best, we compare the Symbolic, Genie3 and Random forest scores, for each of the evaluation measures, using the Friedman test. In 9 cases, the differences among them are statistically significant, therefore

Fig. 6 The average ranks of the ensemble-based scores, compared to MLC-Relief, for all 15 evaluation measures. The legend in (d) applies to all subfigures

we can proceed to the Nemenyi post-hoc test. The average diagrams for two of them, are presented in Fig. 7a (AU PRC) and b (*multi-label precision*). In both figures, the order of the scores is the same: Genie3, Symbolic, Random forest. Based on the results from AU PRC, the difference between Genie3 and Random forest is statistically significantly different, while based on *multi-label precision*, the difference between Random forest and the other two ranking scores is statistically significantly different.

We can conclude that Random forest does not give the best rankings. Since Genie3 and Symbolic score typically result in the rankings of (approximately) the same quality, the final decision is made based on the time complexity: since we need induce 50 trees to obtain the best performance with Symbolic score, and only 10 to achieve that with Genie3 score, we identify Genie3 feature ranking score as the one that should be preferred.

7 Conclusions

In this paper, we propose three ensemble-based feature ranking scores for multi-label classification (MLC): Symbolic, Genie3 and Random forest. We compute each of them from three different ensembles of predictive clustering trees: *Bagging*, *Random forest* and *Extra trees* (with the exception that the Random forest score cannot be computed from *Extra trees* ensemble).

We extensively evaluate the proposed scores on 24 benchmark MLC problems, using 15 standard MLC evaluation measures. For each score and ensemble method, we first determine the saturation point in terms of the ensemble size, after which the quality of the ranking does not improve any further. We find out that 10 trees usually suffice, whereas the Symbolic score needs *Bagging* and *Random forest* ensembles of size 50.

Next, we show that the proposed feature ranking scores yield relevant feature rankings, by comparing them to the non-informed ranking which values all the features the same. After that, we first determine the most appropriate ensemble method for every feature ranking score and empirically prove that the proposed feature ranking scores outperform current state-of-the-art methods in the quality of the rankings (for the majority of the evaluation measures), and in time efficiency. Finally, we determine which of the proposed feature ranking scores is the best. Taking into account the quality of the rankings first and—in case of the ties—time efficiency, we find that the Genie3 feature ranking score is the optimal one.

Future work can follow two major directions. First, since the Random forest score typically yields the worst feature rankings, it could be additionally analyzed and

Fig. 7 The average rank diagrams of the proposed scores. Statistically significant differences are assessed by the Nemenyi post-hoc test. The quality of the rankings is measured in terms of AU PRC (a) and multi-label precision (b)

parameterized with MLC error measures in Eq. (3) to obtain better performance. Second, other ensemble techniques may be evaluated, such as gradient boosting, which could be even more efficient.

Acknowledgements We acknowledge the financial support of the Slovenian Research Agency via the Grants P2-0103, J2-9230, J7-1815, J7-9400 and N2-0128, and a young researcher grant to MP. The computational experiments presented here were executed on a computing infrastructure from the Slovenian Grid (SLING) initiative.

References

- Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1), 289–300.
- Blockeel, H. (1998). *Top-down induction of first order logical decision trees*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium.
- Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9), 1757–1771.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Breiman, L., Friedman, J., Olshen, R., & Stone, C. J. (1984). *Classification and regression trees*. Boca Raton: CRC.
- Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., et al. (2013). The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In *IEEE international workshop on machine learning for signal processing, MLSP* (Vol. 2013, pp. 1–8).
- Dembczyński, K., Waegeman, W., Cheng, W., & Hüllermeier, E. (2012). On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1), 5–45.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
- Diplaris, S., Tsoumakas, G., Mitkas, P., & Vlahavas, I. (2005). Protein classification with multiple algorithms. In *10th panhellenic conference on informatics (PCI 2005)*, pp. 448–456.
- Duygulu, P., Barnard, K., de Freitas, J. F. G., & Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In A. Heyden, G. Sparr, M. Nielsen, & P. Johansen (Eds.), *Computer vision—ECCV 2002* (pp. 97–112). Berlin: Springer.
- Elisseeff, A., & Weston, J. (2001). A kernel method for multi-labelled classification. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems* (Vol. 14). Berlin: Springer.
- Elkafrawy, P., Mousad, A., & Esmail, H. (2015). Experimental comparison of methods for multi-label classification in different application domains. *International Journal of Computer Applications*, 114, 1–9.
- Geurts, P., Erns, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 36(1), 3–42.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157–1182.
- Hancock, T., Jiang, T., Li, M., & Tromp, J. (1996). Lower bounds on learning decision lists and trees. *Information and Computation*, 126(2), 114–122.
- Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 993–1001.
- Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., & Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9), 1–10.
- Katakis, I., Tsoumakas, G., & Vlahavas, I. (2008). Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD 2008 discovery challenge*.
- Kocev, D., & Ceci, M. (2015). Ensembles of extremely randomized trees for multi-target regression. In *Discovery science: 18th international conference (DS 2015)*, volume 9356 of LNCS, pp. 86–100.
- Kocev, D., Slavkov, I., & Džeroski, S. (2013a). Feature ranking for multi-label classification using predictive clustering trees. In I. Katakis, D. Hernandez-Lobato, G. Martinez-Munoz, & I. Partalas (Eds.), *Solving complex machine learning problems with ensemble methods*. Prague, Czech Republic: COPEM.
- Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2013b). Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3), 817–833.

- Kong, D., Ding, C., Huang, H., & Zhao, H. (2012). Multi-label ReliefF and F-statistic feature selections for image annotation. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2352–2359.
- Kononenko, I., & Robnik-Šikonja, M. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning Journal*, 55, 23–69.
- Lee, J., & Kim, D.-W. (2017). Scls: Multi-label feature selection based on scalable criterion for large label set. *Pattern Recognition*, 66, 342–352.
- Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45, 3084–3104.
- Pereira, R. B., Plastino, A., Zadrozny, B., & Merschmann, L. H. C. (2015). Information gain feature selection for multi-label classification. *Journal of Information and Data Management*, 6(1), 48–58.
- Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D. J., Johnson, N., Bretonnel Cohen, K., et al. (2007). A shared task involving multi-label classification of clinical free text. In *Proceedings of the workshop on BioNLP 2007: Biological, translational, and clinical language processing (BioNLP '07)*, pp. 97–104.
- Petković, M., Kocev, D., & Džeroski, S. (2018). Feature ranking with relief for multi-label classification: Does distance matter? In L. Soldatova, J. Vanschoren, G. Papadopoulos, & M. Ceci (Eds.), *Discovery science* (pp. 51–65). Berlin: Springer.
- Petković, M., Kocev, D., & Džeroski, S. (2019). Feature ranking for multi-target regression. *Machine Learning Journal*, <https://doi.org/10.1007/s10994-019-05829-8>.
- Reyes, O., Morell, C., & Ventura, S. (2015). Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context. *Neurocomputing*, 161, 168–182.
- Sechidis, K., Nikolaou, N., & Brown, G. (2014). Information theoretic feature selection in multi-label data through composite likelihood. In P. Fränti, G. Brown, M. Loog, F. Escolano, & M. Pelillo (Eds.), *Structural, syntactic, and statistical pattern recognition* (pp. 143–152). Berlin: Springer.
- Snoek, C. G. M., Worring, M., van Gemert, J. C., Geusebroek, J.-M., & Smeulders, A. W. M. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th ACM international conference on multimedia* (pp. 421–430). New York, NY: ACM.
- Spolaôr, N., Cherman, E. A., Monard, M. C., & Lee, H. D. (2013). A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic Notes in Theoretical Computer Science*, 292, 135–151.
- Spolaôr, N., & Tsoumakas, G. (2013). Evaluating feature selection methods for multi-label text classification. In *BioASQ Workshop*, pp. 1–12.
- Spyromitros, E., Tsoumakas, G., & Vlahavas, I. (2008). An empirical study of lazy multilabel classification algorithms. *Artificial intelligence: Theories, models and applications, LNAI 5138* (pp. 401–406). Berlin: Springer.
- Srivastava, A. N., & Zane-Ulman, B. (2005). Discovering recurring anomalies in text reports regarding complex space systems. In *2005 IEEE Aerospace Conference*.
- Trochidis, K., Tsoumakas, G., Kalliris, G., & Vlahavas, I. (2008). Multilabel classification of music into emotions. In *2008 International conference on music information retrieval (ISMIR 2008)*, pp. 325–330.
- Tsoumakas, G., Katakis, I., & Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. In *ECML/PKDD 2008 workshop on mining multidimensional data (MMD'08)*.
- Tsoumakas, G., & Vlahavas, I. (2007). Random k-labelsets: An ensemble method for multilabel classification. In J. N. Kok, J. Koronacki, R. L. D. Mantaras, S. Matwin, D. Mladenič, & A. Skowron (Eds.), *Machine learning: ECML 2007* (pp. 406–417). Berlin: Springer.
- UC Berkeley. (2018). *Enron Email analysis project*. http://bailando.sims.berkeley.edu/enron_email.html. Accessed 2018-06-28.
- Ueda, N., & Saito, K. (2003). Parametric mixture models for multi-labeled text. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems* (Vol. 15, pp. 721–728). Cambridge: MIT Press.
- Vens, C., Struyf, J., Schietgat, L., Džeroski, S., & Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2), 185–214.
- Wettschereck, D. (1994). *A study of distance based algorithms*. PhD thesis, Oregon State University, USA.