Machine Learning (2020) 109:1509–1563
https://doi.org/10.1007/s10994-020-05879-3

An empirical analysis of binary transformation strategies and base algorithms for multi-label learning

Adriano Rivolli[1] · **Jesse Read**[2] · **Carlos Soares**[3] · **Bernhard Pfahringer**[4] · **André C. P. L. F. de Carvalho**[5]

Received: 24 April 2018 / Revised: 9 January 2020 / Accepted: 7 April 2020 / Published online: 10 June 2020

Abstract
Investigating strategies that are able to efficiently deal with multi-label classification tasks is a current research topic in machine learning. Many methods have been proposed, making the selection of the most suitable strategy a challenging issue. From this premise, this paper presents an extensive empirical analysis of the binary transformation strategies and base algorithms for multi-label learning. This subset of strategies uses the one-versus-all approach to transform the original data, generating one binary data set per label, upon which any binary base algorithm can be applied. Considering that the influence of the base algorithm on the predictive performance obtained by the strategies has not been considered in depth by many empirical studies, we investigated the influence of distinct base algorithms on the performance of several strategies. Thus, this study covers a family of multi-label strategies using a diversified range of base algorithms, exploring their relationship over different perspectives. This finding has significant implications concerning the methodology of evaluation adopted in multi-label experiments containing binary transformation strategies, given that multiple base algorithms should be considered. Despite these improvements in strategy and base algorithms, for many data sets, a large number of labels, mainly those less frequent, were either never predicted, or always misclassified. We conclude the experimental analysis by recommending strategies and base algorithms in accordance with different performance criteria.

Keywords Multi-label learning · Binary transformation · Comparison of strategies · Base algorithms · Empirical analysis

Editor: Eyke Hüllermeier.

✉ Adriano Rivolli
rivolli@utfpr.edu.br

Extended author information available on the last page of the article

1 Introduction

Multi-label learning has been investigated widely by the machine learning community in recent years (de Carvalho and Freitas 2009; Tsoumakas et al. 2010; Gibaja and Ventura 2014). It deals with classification tasks where an instance can be simultaneously classified into more than one class. Each class is represented by one label. Several domains, such as text (Klimt and Yang 2004; Pestian et al. 2007), multimedia (Duygulu et al. 2002; Zhou and Zhang 2006; Briggs et al. 2013) and biology (Elisseeff and Weston 2001), are intrinsically multi-label.

A common approach to dealing with multi-label classification tasks is to transform the original data set into one or more single-label data sets. A conventional binary classification algorithm, called *base algorithm* here, is used to induce predictive models for each one of them. As such, a transformation strategy defines how to decompose the original task into a set of single-label tasks and to combine the results obtained from these tasks to solve the original task (Tsoumakas et al. 2010). Many strategies have been proposed to address the multi-label tasks and transform the data, exploring different aspects, such as label correlation (Read et al. 2011; Cherman et al. 2012; Montañes et al. 2014), dimensionality reduction (Tsoumakas et al. 2008; Zhang and Wu 2015) and class imbalance (Zhang and Wu 2015; Tsoumakas et al. 2011b).

Although the base algorithm can be seen as a hyperparameter for transformation strategies, it is generally fixed for all strategies, so that only a single base algorithm is considered in the whole experiment (Read et al. 2011; Montañes et al. 2014; Madjarov et al. 2012). Given that a comprehensive comparison of the binary transformation strategies, using different base algorithms, has not yet been performed, this study assesses the hypothesis that the base algorithms can have a stronger influence than the binary transformation strategies on the predictive performance of multi-label models. At a glance, it may seem trivial to be investigated, however, if the choice of a base algorithm is more important regarding the quality of the results than the specific strategy, then several of them should be considered in empirical studies evaluating these strategies.

In the multi-label literature, the most similar comparative study was performed by Madjarov et al. (2012), where 12 strategies (including 3 binary transformation strategies) were evaluated under several measures, using the original train and test partition of 11 benchmark data sets. Even though a variety of different algorithms were considered, the transformation strategies were evaluated with a single base algorithm, Support Vector Machine (SVM). Another large empirical study covering multiple ensemble strategies (Moyano et al. 2018) used only the C4.5 decision tree as the base algorithm. Nevertheless, a few studies have considered using more than one base algorithm. These studies include Tsoumakas and Katakis (2007) and Cherman et al. (2012), who did not compare strategies using different base algorithms; and Zufferey et al. (2015), who compared strategies with distinct base algorithms, but just in a single data set.

Methods using Automatic Machine Learning (Auto-ML) to address multi-label classification tasks also consider multiple base algorithms (de Sá et al. 2017, 2018; Wever et al. 2018, 2019). During the search for a solution, the Auto-ML method may find a suitable combination between strategies and base algorithms that optimizes a fitness function. In these cases, choosing the base algorithm is seen as part of the solution and the comparison of the strategies does not fix a base algorithm, as observed in other studies.

Since the most common strategies are based on binary transformations, this paper will focus on these strategies. Hence, 10 binary transformation strategies and 5 different base

algorithms (plus one with its hyperparameters tuned) were evaluated using 5×2-fold cross-validation for 20 benchmark data sets. In contrast to previous studies, which used null hypothesis significance testing, we ran Bayesian statistic tests (Benavoli et al. 2017) to assess the statistical significance of the differences in the predictive performance of the assessed strategies over different evaluation measures. To the best of our knowledge, this is the most extensive multi-label empirical study carried out so far.

The results reported reinforce the claim that the predictive performance obtained by transformation strategies is affected by the base algorithm used. Thus, experimental studies in multi-label learning must take into account experiments with several different base algorithms. In particular, many of the binary transformation strategies obtained very similar results, with differences mainly being due to the choice of the base algorithm used. Therefore, previous comparative studies (Madjarov et al. 2012; Moyano et al. 2018) might have reached different conclusions if other base algorithms had been employed. Additionally, for many data sets, the investigated strategies consistently predicted only a subset of the existing labels, never assigning the remaining labels to any instance. This problem was previously observed in the food truck data set (Rivolli et al. 2018), however, as far as we know, it has never been widely investigated.

The rest of the paper is organized as follows: Sect. 2 formally defines the main concepts relevant for multi-label learning. Section 3 details the investigated strategies. Section 4 describes the experimental design, including data sets, evaluation procedures, base classifiers, tools, and hyperparameter values adopted. Section 5 presents, analyzes and discusses the empirical results. In the last section, conclusions are drawn concerning relevant findings from the experimental study and future work directions.

2 Multi-label learning

In multi-label learning, an instance can be simultaneously associated with more than one label. The main tasks in this field are *Multi-Label Classification* and *Label Ranking*.

Multi-Label Classification (MLC), the most common task (Tsoumakas et al. 2010), induces a predictive model $h : \mathcal{X} \rightarrow \mathcal{Y}$ from a set of training data $\mathcal{D}$, which later assigns labels to new examples. This task can be formally defined as follows. Let $\mathcal{D}$ be a set of labeled instances, such that $\mathcal{D} = \left\{(x_1, Y_1), ..., (x_n, Y_n)\right\}$. Every labeled instance is composed of $x_i = (x_{i1}, x_{i2}, ..., x_{id}) \in \mathbb{R}^d$, and $Y_i \subseteq \mathcal{L}$, such that $\mathcal{L} = \left\{\lambda_1, \lambda_2, ..., \lambda_q\right\}$ is the set of all q labels λ_i. For the sake of convenience, the labels associated with the i^{th} instance, also called label set, can be seen as a binary vector $y_i = (y_{i1}, y_{i2}, \ldots, y_{iq}) \in \{0, 1\}^q$, where $y_{ij} = 1$ *iff* $\lambda_j \in Y_i$ and $y_{ij} = 0$ *iff* $\lambda_j \notin Y_i$. Finally, model h is used to predict, for a test instance $(x_i, ?)$, the set of relevant labels $\hat{Y}_i$ (or $\hat{y}_i$ as a binarized prediction).

In the *Label Ranking* (LRK) task, a model outputs the ranked labels for each test instance. This ranking can easily be computed using any model that provides a score value indicating its probability of being relevant to a given instance. Thus, the higher the score value, the better its ranking position. In turn, MLC can be derived from the LRK formulation (Gibaja and Ventura 2015).

A multi-label model can be obtained by using two approaches (Tsoumakas and Katakis 2007), *problem transformation* and *algorithm adaptation*. The former converts the original multi-labeled data into a set of binary or multi-class data sets, whereas for the latter, the multi-label support is embedded into the algorithm's structure. Thus, the transformation

approach fits the data to the algorithms, and the adaptation approach fits the algorithms to the data (Zhang and Zhou 2014).

A straightforward transformation is to build a binary classifier for each label individually. This is known as the *Binary* approach. On the other hand, a multi-class transformation can be considered, in which each label set (combination of labels) is mapped to one class. Both approaches are *algorithm independent* (de Carvalho and Freitas 2009), in the sense that any traditional classification algorithm that is capable of handling such problems can be used as the base algorithm.

We want to emphasize that the binary transformation approach implies that algorithms are trained separately, but not necessarily independently; this will become apparent in the following section. In addition, many hybrid approaches exist, such as *Pairwise*, which models pairwise combinations (a one-vs-one approach), and subset approaches, which includes the well-known RAkEL strategy (Tsoumakas et al. 2011a).

Binary transformation generates at least one data set per label. Each binary data set $\mathcal{D}'_j$ is related to the label λ_j. The instances associated with λ_j are labeled with a class value of "1", all others are labelled with a class value of "0".The number of binary data sets generated is defined by $|\mathcal{D}'| = mq$, where m is the number of data sets per label. Therefore, the complexity of this family of strategies is linear in the number of labels q. Negative aspects of this approach include the tendency to generate rather imbalanced data sets and the fact that some of these strategies ignore the relationships between labels (Zhou et al. 2012).

The binary transformation strategies are organized into three groups, *one-round*, *stacking*, and *ensemble*, according to the value of m. *One-round* strategies are the simplest strategies, with $m = 1$. A special case of one-round is *chaining*, which increases the input space by adding already predicted labels as features to predict the others, in a chain. In *stacking* strategies, two rounds of training and prediction steps are performed, thus $m = 2$. They augment the input space in the second round by using the values of the labels predicted in the first round as features. When all the labels are used, they are called *full-stacking*. When only a subset of the labels is used, they are called *pruned-stacking*. Finally, in the *ensemble* strategies more than two models for each label ($m > 2$) are used and usually, the value of m is a hyperparameter defined by the user. When the same instances and attributes are shared by all internal models, the ensemble is *homogeneous*. However, when each member and label use distinct data sets as training data, the ensemble is *heterogeneous*. The former can be seen as an ensemble of multi-labeled data, whereas the latter as multiple ensembles of single-label data (Gibaja and Ventura 2015). These groups and their strategies are detailed in Sect. 3.

A base classification algorithm must always be chosen to induce predictive models for each transformed data set $\mathcal{D}'$. Later, these models are used to predict the relevance of each label for new instances. If the models predict a score instead of a class, the strategies support both tasks, MLC and LRK (Gibaja and Ventura 2015). Logically, if the base algorithms are responsible for predicting a score and the binary transformation strategies are independent from them, any transformation strategy can be used to solve them. Distinctions among them will not be considered in the rest of this paper.

As previously mentioned, this study is restricted to analyzing strategies based on binary transformation, which are relevant for a broad group of researchers and practitioners. Besides, for most of them, their individual models can be trained separately (thus, allowing for parallelism), they are simple to interpret, they have been successfully used in many state-of-the-art comparisons in the literature, and they usually exhibit acceptable time complexity, almost linear with the number of labels. Using separate classifiers, each focused on only one label, allows for higher flexibility, choosing potentially different approaches

Table 1 Binary transformation strategies organized into groups/subgroups according to the number of binary models per label and their main characteristic

Group	Subgroup	Strategy	References
One-round	-	BR	Boutell et al. (2004)
	Chaining	CC	Read et al. (2011)
		NS	Senge et al. (2013)
Stacking	Full	BR+	Cherman et al. (2012)
		DBR	Montañes et al. (2014)
		RDBR	Rauber et al. (2014)
	Pruned	MBR	Godbole and Sarawagi (2004)
		PruDent	Alali and Kubat (2015)
Ensemble	Homogeneous	EBR	Read et al. (2011)
		ECC	Read et al. (2011)

on a per-label basis. Furthermore, new labels can usually be added to the problem without retraining the models built for existing labels. In general, as some of the strategies are conceptually quite similar to each other, their practical differences may be highlighted by comparing their performances using different base algorithms, an approach we put forward in this paper.

3 Strategies

In this section, the 10 binary transformation strategies considered are described. Table 1 presents the strategies organized into groups, defined by the number of binary models generated per label, and the subgroups according to their main characteristic.

3.1 One-round

The *one-round* strategies are characterized by generating only a single binary data set for each label. Binary models are induced from these data sets and used for multi-label prediction. The strategies from this group differ mainly by how they transform the data sets.

Binary Relevance (BR) (Boutell et al. 2004) is the simplest and most popular multi-label strategy (Luaces et al. 2012; Montañes et al. 2014). For each label λ_j, an independent binary data set is generated according to

$$\mathcal{D}'_j = \left\{(x_i, y_{ij}) \mid 1 \leq i \leq n\right\}, \tag{1}$$

and will be used to induce a binary model θ_j. The prediction is performed using the values of all binary models as follows:

$$h_{br} = \{\lambda_j \mid \theta_j(x) = 1,\ 1 \leq j \leq q\}. \tag{2}$$

3.1.1 Chaining

The *Classifier Chains* (CC) strategy (Read et al. 2009, 2011) organizes the labels in a chain and increases the original input space of the transformed data set for a given label with the values of all previous labels in the chain. Thus, the data set is transformed as follows:

$$\mathcal{D}'_j = \left\{([x_i, y_{i1}, y_{i2}, \ldots, y_{i(j-2)}, y_{i(j-1)}], y_{ij}) \mid 1 \leq i \leq n\right\}. \tag{3}$$

The model related to the first label in the chain is obtained exclusively from the original input data, without adding any predictive attributes, as shown in Eq. 1. The other models increase their input space by adding $j-1$ new attributes, where j is the position of the respective label in the chain. During the prediction phase, as the labels are predicted, their values are used to increase the input space, as shown next

$$\begin{aligned} h_{cc} &= \{\lambda_j \mid \hat{y}_j = 1,\ 1 \leq j \leq q\}, \text{where} \\ \hat{y}_j &= \theta_j([x, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{(j-2)}, \hat{y}_{(j-1)}]). \end{aligned} \tag{4}$$

Nested Stacking (NS) (Senge et al. 2013) brings two modifications to CC. In the training phase, it uses the predicted labels instead of the real labels. Furthermore, in the prediction phase, it makes a subset correction, in order to predict only preexisting label sets.

The transformation step is similar to Eq. 3. However, the original label values y are changed by the predicted values $\hat{y}$, such that

$$\mathcal{D}'_j = \left\{([x_i, \hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{i(j-2)}, \hat{y}_{i(j-1)}], y_{ij}) \mid 1 \leq i \leq n\right\},$$

where $\hat{y}_{ij}$ is the prediction of the binary model θ_j for the instance x_i presented in the training data. The prediction is obtained similarly to Eq. 4 followed by the subset correction. The $\hat{y}$ is replaced by $y^* \in Y$, which is the vector in Y that is most similar to $\hat{y}$, such that

$$\begin{aligned} h_{ns} &= \{\lambda_j \mid y^*_j = 1,\ 1 \leq j \leq q\}, \text{where} \\ y^* &= \arg\min_{y \in Y} dist(\hat{y}, y), \end{aligned}$$

and *dist* is the hamming distance, which corresponds to the number of differences between two binary vectors. When more than one minimum is found, the label set with the higher frequency in the training data is selected.

3.2 Stacking

The stacking strategies are characterized by using the stacked generalization learning paradigm (Wolpert 1992). In the multi-label context, they use two rounds of binary transformation, where in the second round, the input space is augmented by the information from the labels obtained from the first round.[1] The main difference among the stacked strategies is how they choose the labels that would augment the input space. Some of them use all labels (full stacking), while others use only a subset of labels (pruned stacking).

3.2.1 Full stacking

BR+ (Cherman et al. 2012) and *Dependent Binary Relevance* (DBR) (Montañes et al. 2014) are very similar to each other. In the training phase, they perform exactly the same procedure. The first round is characterized by the induction of a BR model, according

[1] Although CC and NS also augment the input space, they are not considered stacking, given that only one-round is performed.

to Eqs. 1 and 2. In the second round, the transformation is performed by increasing the input space using the original labels. To illustrate how it works, let $\phi_j(y)$ be a function that removes the label λ_j from the vector y, such that

$$\begin{aligned} \mathcal{D}_j^{''} &= \left\{([x_i, \phi_j(y_i)], y_{ij}) \mid 1 \leq i \leq n\right\}, \text{ where} \\ \phi_j(y) &= (y_1, \ldots, y_{(j-1)}, y_{(j+1)}, \ldots, y_q). \end{aligned} \tag{5}$$

It should be noted though, that there is a subtle difference in the prediction phase, precisely, in the second round. DBR predicts the labels using the second round binary models that use the labels obtained from the first round binary models. Using the ϕ function presented in Eq. 5, the prediction is obtained as follows:

$$h_{dbr} = \{\lambda_j \mid \theta_j^{''}([x, \phi_j(h_{br}(x))]) = 1,\ 1 \leq j \leq q\}.$$

Differently, BR+ updates the labels from the first round binary models while the second prediction is occurring. Given a chain of labels (for example, $\lambda_1 \prec \lambda_2 \prec \cdots \prec \lambda_q$), the prediction is obtained in the following way:

$$\begin{aligned} &h_{br+} = \{\lambda_j \mid \theta_j^{''}([x, \phi_j(\hat{y})]) = 1,\ 1 \leq j \leq q\}, \\ &\text{for each } j, \quad \hat{y} = (\hat{y}_1, \ldots, \hat{y}_{(j-1)}, \theta_j^{''}([x, \hat{y}])), \hat{y}_{(j+1)}, \ldots, \hat{y}_q). \end{aligned} \tag{6}$$

Recursive Dependent Binary Relevance (RDBR) (Rauber et al. 2014) induces two models as DBR does, but it uses the second model several times in a recursive way. The labels predicted for the second model are used to update the input space and the second round is executed again until either the result converges or a fixed number of iterations is reached. In practice, it is the same process as in Eq. 6, but while BR+ does only one update, RDBR updates recursively several times until a stopping criterion is reached.

3.2.2 Pruned stacking

The *Meta-BR* (MBR) strategy[2] (Godbole and Sarawagi 2004; Read et al. 2011) augments the input space using the values of the most correlated labels (Tsoumakas et al. 2009). The Pearson product moment correlation coefficient for categorical variables ρ is computed for each pair of labels and a threshold τ is used to define which labels should augment the space of attributes. The data set in the second round is obtained in the following way:

$$\begin{aligned} \mathcal{D}_j^{''} &= \left\{([x_i, \phi_j(\hat{y}_i)], y_{ij}) \mid 1 \leq i \leq n\right\}, \text{where} \\ \phi_j(\hat{y}) &= \{\hat{y}_l \mid \rho(\lambda_j, \lambda_l) \geq \tau,\ 1 \leq l \leq q\}, \end{aligned}$$

and $\phi(\hat{y}_i)$ returns only the most related labels. Unlike the other stacked strategies, instead of using the original labels in the second transformation, it uses the predicted labels obtained in the first round.

The final prediction is the result of the binary models in the second step, such that:

[2] Also known as 2BR (Tsoumakas et al. 2009), *Meta-Stacking* (Read et al. 2009) and *Stacking* (Montañes et al. 2014).

$$h_{mbr} = \{\lambda_j \mid \theta_j''([x, \phi_j(h_{br}(x))]) = 1,\ 1 \leq j \leq q\}.$$

The *Pruned and confiDent* (PruDent) strategy (Alali and Kubat 2015) uses only the most relevant labels, as MBR does, and the original values to augment the second input space, as BR+ and DBR do. The Information Gain (IG) measure is used to prune the irrelevant labels based on a threshold τ. The PruDent transformation is the same as Eq. 5, with the exception of the ϕ function:

$$\phi_j(y) = \{y_l \mid IG(\lambda_j, \lambda_l) \geq \tau,\ 1 \leq l \leq q,\ l \neq j\}.$$

Contrary to the others, PruDent assigns a label to an example if either one of the corresponding models, first or second round, predicts it. The predictions are done in the following way:

$$h_{prud} = \{\lambda_j \mid \theta_j(x) = 1 \vee \theta_j''([x, \phi_j(h_{br}(x))]) = 1,\ 1 \leq j \leq q\}.$$

3.3 Ensemble

Ensemble of Binary Relevance (EBR) and *Ensemble of Classifier Chains* (ECC) (Read et al. 2011) are simply ensembles of models induced by the BR strategy and by the CC strategy, respectively. Both BR and CC use bagging and choose different random subsets of the attributes for each bagging iteration. To illustrate how EBR computes predictions, let m be the number of models in the ensemble and ϕ_i a function for selecting a random subset of attributes:

$$h_{ebr} = \{\lambda_j \mid \left(\frac{1}{m}\sum_{l=1}^{m} \hat{y}_{lj}\right) > \tau,\ 1 \leq j \leq q\},\ \text{where}$$
$$\hat{y}_l = h_{br}^l(\phi_l(x)),$$

$\hat{y}_{lj}$ is the predicted value of the BR model l for the label λ_j and τ is a threshold value.[3] For the ECC strategy, internal models are built using h_{cc} with different chains, avoiding the influence that choosing an inappropriate chain could have on the results.

4 Experimental design

This section presents an experimental comparison across the binary transformation strategies and base algorithms. It describes the multi-label data sets, followed by a short overview of evaluation measures and procedures. Next, it explains the methodology adopted and the environmental setup.

[3] It can either be a predefined value, such as 0.5 (Read et al. 2011) or dynamically defined using the cardinality value of the training data set (Read et al. 2009).

Table 2 Characteristics of the multi-label data sets

Data set	Domain	Inst	Attr	Lbl	lSets	PUL	lCard	lDen	Dep	IID	Corr
20ng	text	19,300	1006	20	55	0.31	1.03	0.05	0.08	0.9	0.45
birds	audio	337	260	15	115	0.53	1.84	0.12	0.08	0.75	0.39
cal500	audio	502	68	141	502	1.00	25.54	0.18	0.14	0.67	0.15
corel5k	image	4995	499	218	2940	0.76	3.37	0.02	0.16	0.97	0.12
emotions	audio	593	72	6	27	0.15	1.87	0.31	0.28	0.38	0.41
enron	text	1702	1001	42	722	0.74	3.34	0.08	0.12	0.84	0.22
fapesp	text	251	7286	18	61	0.46	1.35	0.08	0.11	0.85	0.57
flags	other	194	19	7	54	0.44	3.39	0.48	0.15	0.35	0.40
image	image	2000	294	5	20	0.10	1.24	0.25	0.15	0.51	0.33
langlog	text	1197	916	38	223	0.53	1.31	0.03	0.06	0.93	0.29
mediamill	image	42,177	120	101	6554	0.63	4.56	0.05	0.22	0.93	0.10
medical	text	949	1421	20	55	0.22	1.20	0.06	0.19	0.88	0.76
msd-195	audio	2901	180	38	267	0.09	2.47	0.07	0.24	0.87	0.13
ohsumed	text	13,929	1002	23	1147	0.50	1.66	0.07	0.04	0.86	0.32
scene	image	2407	294	6	15	0.20	1.07	0.18	0.11	0.64	0.43
slashdot	text	3776	1079	18	149	0.35	1.18	0.07	0.05	0.87	0.34
stackex-chess	text	1612	585	78	725	0.72	2.07	0.03	0.10	0.95	0.37
tmc2007-500	text	28,596	500	22	1172	0.35	2.22	0.10	0.11	0.81	0.38
yeast	biology	2417	103	14	198	0.39	4.24	0.30	0.25	0.54	0.18
yelp8	image	10,784	668	8	117	0.06	2.26	0.28	0.11	0.48	0.23

4.1 Data sets

Table 2 lists the 20 multi-label data sets used for the experiments. They are from distinct domains (column *Domain*) and have a wide diversity in their characteristics. The columns *Inst*, *Attr* and *Lbl* are respectively the number of instances, attributes and labels. Label sets (*lSets*) is the amount of distinct label combination, proportion of unique label sets (*PUL*) indicates the proportion of label sets related to a single instance, label cardinality (*lCard*) measures the average number of labels per instance, label density (*lDen*) describes the average frequency of labels, dependency (*Dep*) shows the average unconditional labels' dependency (Luaces et al. 2012), inner imbalance degree (*IID*) measures the average label imbalance in the binary data sets (Raez et al. 2004) and, finally, correlation (*Corr*) indicates the average correlation between the predictive attributes and the labels.

Letting ρ_{jk} be the Pearson correlation coefficient between the j^{th} attribute and the label λ_k, the correlation is computed as

$$Corr = \frac{1}{q}\sum_{k=1}^{q} max(|\rho_{1k}|, |\rho_{2k}|, ..., |\rho_{dk}|),$$

where d is the number of attributes. A high value for this measure means that there is at least one attribute which is strongly correlated to each label, while a low value indicates the opposite.

These data sets are frequently used as benchmarks for multi-label experiments. They come from different domains, organized here as text, image, audio, biology and other. The

text-domain data sets are related to aviation safety reports (tmc2007-500, Srivastava and Zane-Ulman 2005), medical documents (medical, Pestian et al. 2007), emails (enron, Klimt and Yang 2004), newsgroups (20ng, Lang 1995), scientific literature (fapesp, Cherman et al. 2014; ohsumed, Joachims 1998), web forums (stackex_chess, Charte et al. 2015), and web content (langlog and slashdot, Read et al. 2011). Text data sets have a higher number of attributes than most of the data sets from the other domains and also contain the largest average value of correlation between attributes and labels.

The image-domain data sets are related to food (yelp), images extracted from videos (mediamill, Snoek et al. 2006), scene classification (image, Zhou and Zhang 2006; scene, Boutell et al. 2004), and vector graphics (corel5k, Duygulu et al. 2002). They have the highest average number of labels and label sets of all domains. The data sets with the highest average dependency degree among the labels are from the audio domain. They are related to detecting emotions in songs (emotion, Trohidis et al. 2011), the identification of music styles (msd-195, Bernardini et al. 2014), music effects classification (cal500, Turnbull et al. 2008) and sounds of birds (birds, Briggs et al. 2013).

The last two data sets are yeast (Elisseeff and Weston 2001), a data set from the biology domain that associates gene expressions with biological functions, and flags (Gonçalves et al. 2013), a data set of the countries where the color of their respective flags are the labels.

The data sets come from the Cometa repository (Charte et al. 2018), an exhaustive collection of MLC data sets, integrated with the tools used in this work. The exceptions are the data sets fapesp and msd-195 obtained from their respective authors, and yelp8 from the Kaggle website.[4] The data sets were preprocessed with three operations. First, the labels with less than 10 instances were removed to ensure a minimum number of instances with each label in the training and test folds. Next, instances with no labels were also removed. Finally, predictive attributes with constant values were removed.

Concerning the characteristics shown in Table 2, the density (LDen) and the inner imbalance degree (IID) are inversely correlated. As the density increases, the imbalance degree decreases, and vice-versa. We did not find high correlation among the other characteristics.

4.2 Evaluation measures

The evaluation of the predictive performance of multi-label strategies requires using different measures to assess different dimensions (Tsoumakas et al. 2010). They are organized here in example-based, label-based and ranking measures. The example-based measures summarize the predictive performance over all instances, whereas the label-based measures summarize the performance over all labels. The ranking measures are a specialization of the former, using the prediction scores instead of the crisp values. As many evaluation measures are highly correlated with each other (Pereira et al. 2018), a subset was used.

[4] see https://www.kaggle.com/c/yelp-restaurant-photo-classification.

4.2.1 Example-based measures

Hamming-loss (HL) is an error measure that evaluates the misclassification rate for each label of every instance (Schapire and Singer 1999). This measure does not distinguish between false positive and false negative errors, giving the same weight for both, as shown next

$$HL = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{q}|h(x_i)\,\Delta\, Y_i|, \text{ where}$$
$$A\,\Delta\, B = (A - B) \cup (B - A). \tag{7}$$

While *Hamming-loss* is the most relaxed measure, *Subset-accuracy* (SA) is the strictest (Gibaja and Ventura 2015). It accounts only for correctly predicted label sets, ignoring the partial hits. A partially correct prediction is valued the same was as a completely incorrect one, such that the set of predicted or observed labels is treated as a class value in single-label classification (Zhang and Zhou 2014). It is computed as

$$SA = \frac{1}{n}\sum_{i=1}^{n} I(h(x_i) = Y_i), \text{ where}$$
$$I(\cdot) = \begin{cases} 1 & \text{if the predicate is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Let us call the labels associated with an instance of relevant labels. We can use them to define the following measures: Precision is the fraction of relevant labels among those predicted. A high precision indicates a high ability of a model to correctly predict the labels, although not necessarily all of them. Recall is the fraction of relevant labels that have been predicted out of all relevant labels. A high recall indicates that a model predicts many labels correctly, but not necessarily only the relevant labels. Thus, the F_1 *measure* (F1) computes the harmonic mean between precision and recall. A model with a high value in this measure can predict the relevant labels accurately and only them. It does not take the true negatives into account, combining just the rate of relevant labels among the predicted ones and the rate of predicted relevant labels over all relevant labels. F1 is computed as

$$F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|h(x_i) \cap Y_i|}{|h(x_i)| + |Y_i|}. \tag{9}$$

4.2.2 Label-based measures

Label-based measures usually come in two variants: micro-averaged and macro-averaged. The macro-averaged measures summarize the label distribution by giving the same weight to all labels (Yang 1999). They assess the consistency across all labels. Thus, they are too sensitive to the performance on the least common labels, which is usually low (Jackson and Moulinier 2002).

To illustrate how these measures work, let TP, FP, TN and FN be respectively the true positive, false positive, true negative and false negative values from a confusion matrix, such that

$$Precision_b = \frac{TP}{TP + FP}, \tag{10}$$

$$Recall_b = \frac{TP}{TP + FN}, \tag{11}$$

$$F1_b = \frac{2TP}{2TP + FP + FN}. \tag{12}$$

The macro label-based version computes the previous measures for each label and returns their average value, such that

$$\text{macro-}\beta = \frac{1}{q}\sum_{j=1}^{q} \beta(TP_j, FP_j, TN_j, FN_j),$$

where $\beta = \{Precision_b \mid Recall_b \mid F1_b\}$, from Eqs. 10, 11 and 12, respectively.

The label problem measures, MLP and WLP, (Rivolli et al. 2018) will be also considered. The *Missing Label Prediction* (MLP) measure indicates the proportion of labels that are never predicted by a strategy. The *Wrong Label Prediction* (WLP) measure, which can be seen as a generalization or relaxation of MLP, represents the case where a label might be predicted for some instances, but these predictions are always wrong. Eqs. 13 and 14 formalize these measures, respectively. In an ideal scenario, their expected value is zero.

$$MLP = \frac{1}{q}\sum_{j=1}^{q} I(TP_j + FP_j == 0) \tag{13}$$

$$WLP = \frac{1}{q}\sum_{j=1}^{q} I(TP_j == 0) \tag{14}$$

4.2.3 Ranking measures

Ranking measures consider the ranking of labels instead of the quality of bipartitions, which defines the labels predicted. *One-error* (OE) is an extreme measure that only assesses the error of the label predicted with most confidence. This measure is computed as follows:

$$OE = \frac{1}{n}\sum_{i=1}^{n} I(\arg\max_{\lambda_j \in \mathcal{L}} f(x_i, \lambda_j) \notin Y_i)$$

Ranking-loss (RL) computes the average rate of label pairs that are incorrectly sorted when using their predicted probabilities. It is calculated as follows:

$$RL = \frac{1}{n}\sum_{i=1}^{n} \frac{|\{(\lambda_j, \lambda_k) | f(x_i, \lambda_j) \leq f(x_i, \lambda_k), (\lambda_j, \lambda_k) \in Y_i \times \overline{Y}_i\}|}{|Y_i||\overline{Y}_i|},$$

where $\overline{Y}_i = \mathcal{L} \setminus Y_i$.

4.3 Multi-label baselines

Different baselines were adopted, optimizing different measures. With the exception of the baseline_{RL}, they were proposed by Metz et al. (2012). The baseline_{F1} literally predicts the label set that maximizes the F1 measure (Eq. 9) for the training data, such that

$$baseline_{F1} = \arg\max_{\hat{Y} \subseteq \mathcal{L}} F1(Y, \hat{Y}),$$

where $\hat{Y}$ is the label set predicted. This baseline is also used to compare the label based measures *macro-F1*, *macro-precision* and *macro-recall*.

The baseline_{HL} predicts the labels present in more than 50% of the training instances, such that

$$baseline_{HL} = \{\lambda_j \,|\, freq(\lambda_j) > 0.5, 1 \leq j \leq q\},$$

where $freq(\lambda_j)$ is the frequency of the label λ_j in the training data. In turn, baseline_{SA} predicts the most frequent label set in the training data, such that

$$baseline_{SA} = \arg\max_{\hat{Y} \subseteq \mathcal{L}} \sum_{i=1}^{n} I(Y_i = \hat{Y})$$

where I is the indicator function defined in Eq. 8.

Finally, the baseline_{RL} (Rivolli et al. 2018), an adaptation of the $General_B$ baseline (Metz et al. 2012), predicts a ranking of labels according to their frequency, such that

$$rank(\lambda_j) = |\mathcal{L}| - |\{\lambda_k \mid \lambda_k \in \mathcal{L}, freq(\lambda_j) > freq(\lambda_k)\}|,$$

and

$$baseline_{RL} = \{\lambda_j \,|\, rank(\lambda_j) \leq lcard,\ 1 \leq j \leq q\},$$

where *lcard* is the label cardinality of the training data. This baseline is used for the ranking measures: *one-error* and *ranking-loss*.

4.4 Base algorithms

The strategies described in Sect. 3 require using a base algorithm to induce binary models. Algorithms that are frequently used as the base algorithm in multi-label experiments are *Decision Tree Induction Algorithms* (Cherman et al. 2012; Alali and Kubat 2015; Tsoumakas et al. 2009), *Logistic Regression* (LR) (Montañes et al. 2014; Rauber et al. 2014; Senge et al. 2013; Tsoumakas et al. 2009) and *Support Vector Machines* (SVM) (Read et al. 2011; Cherman et al. 2012; Li and Zhang 2014; Luaces et al. 2012; Madjarov et al. 2012; Tsoumakas et al. 2009).

Two classification algorithms that have been very successful in classification tasks, but not commonly used for multi-label classification, *Random Forest* (RF) and *eXtreme Gradient Boosting* (XGB), complete the set of base algorithms used in our experiments.

The *k-Nearest Neighbors* and *Naive Bayes* algorithms were initially considered. They were discarded because they did not show competitive results when compared with the others. Although other base algorithms, such as Multilayer Perceptron, could also be

Table 3 Hyperparameters values for the strategies used in the experiments

Strategy	Parameters/Values
BR/DBR	-
CC/NS	`chain = random(`$\mathcal{L}$`)`
BR+	`strategy = "Dyn"`
EBR/ECC	`m=10`
	`subsample = 0.75`
	`attr.space = 0.5`
MBR/PruDent	`phi = 0.1`
RDBR	`max.iterations = 5`
	`batch.mode = FALSE`

Table 4 Hyperparameter values of the base algorithms used in the experiments

Base algorithm	Parameters/Values	References
C5.0	`trials = 1`	Quinlan (1993)
	`CF = 0.25`	
	`minCases = 2`	
LR	-	Gelman and Hill (2007)
RF	`ntree = 500`	Breiman (2001)
SVM	`kernel = "radial"`	Chang and Lin (2011)
	`cost = 1`	
	`gamma = 1 / d`	
SVMt	`kernel = "radial"`	Madjarov et al. (2012)
	`cost =` $[2^{-5}, 2^{-3}, \ldots, 2^{13}, 2^{15}]$	
	`gamma =` $[2^{-15}, 2^{-13}, \ldots, 2^{1}, 2^{3}]$	
XGB	`nrounds = 100`	Chen and Guestrin (2016)
	`eval_metric = "error"`	
	`early_stop_round = 2`	

investigated, they were not considered because those selected were able to support the claims addressed in this paper.

4.5 Experimental setup

The experiments were carried out using the R environment. The data sets were handled using code from the `mldr` package (Charte and Charte 2015). The strategies used R code from the `utiml` package (Rivolli and de Carvalho 2018). By default, `utiml` prevents empty predictions (Liu and Chen 2015), in which case the strategy outputs the label with the highest probability/score, preventing an example from being predicted without any labels.

Most strategies and base algorithms used in the experiments require the definition of hyperparameter values. Table 3 shows, for each strategy used, the default values recommended by the packages for the main hyperparameters.

The implementation of the base algorithms used in the experiments come from the packages `C50`, `stats`, `randomForest`, `e1071` and `xgboost` for C5.0, LR, RF, SVM

and XGB, respectively. Table 4 shows the values used for the hyperparameters of each base algorithm, which were those recommended in the corresponding package. SVMt is a tuned version of SVM for the *macro-F1* measure, where the range of values used in a Grid Search procedure is reported. To validate the hyperparameter values, holdout with 70% for training and 30% for validation is adopted for all data sets. SVM was singled out for tuning, due to the high effect of hyperparameter values on its performance (Mantovani et al. 2015).

All results were obtained using 5×2-fold cross-validation with paired folds across all combinations of strategies and base algorithms. An iterative algorithm for the stratification of multi-labeled data (Sechidis et al. 2011) was applied to ensure similar label distributions between training and test data.

Different from previous comparative studies in the multi-label domain, two Bayesian statistical tests were used (Benavoli et al. 2017). The Bayesian hierarchical correlated t-test was used to compare two strategies over multiple data sets, whereas the Bayesian correlated t-test was used for a single data set. When comparing two strategies, the Bayesian statistical test outputs the probability of three situations: strategy 1 is the best (left); strategy 2 is the best (right); and there is a draw between them (rope), which is a region of practical equivalence that indicates an insignificant difference in performance between the strategies. Benavoli et al. (2017) suggest the interval $[-0.01, 0.01]$, which represents a difference of 1% for a measure whose range is [0, 1]. This interval was used for all evaluation measures, with the exception of *hamming-loss*, where the interval was modified $[-0.001, 0.001]$ due to its finer granularity when compared to the other measures. Otherwise, no statistical differences was observed, given that, for *hamming-loss*, the number of mistakes made by a strategy is divided by the number of test instances times the number of labels. Thus, the larger the data set, the smaller the differences between the strategies.

5 Experimental results

This section presents the experimental results and the main findings from this study. The complete set of experimental results is publicly available online at https://rivolli.github.io/ml-binary-transformation/.

Initially, this section compares the results with multi-label baselines followed by the comparison of the most similar strategies. Next, the strategies are compared using fixed base algorithms, which is the traditional approach used in the multi-label literature. Afterwards, the base algorithms are compared by fixing the strategies. In the last set of comparisons, both strategies and base algorithms are combined without distinction. Finally, the main findings are highlighted.

5.1 Comparison with the baselines

Despite their importance for evaluating predictive performance, baselines have not been frequently used in multi-label experiments (Metz et al. 2012). As a result, there are no clear standards for selecting baselines for evaluation. Table 5 presents a comprehensive set of results for the different baselines (Sect. 4.3) used in the experiments.

The baseline$_{F1}$ obtained the highest results for all measures in data sets with high average labels' frequency and low imbalance degree. The baseline$_{HL}$, on the contrary, had its best results in data sets with low average label frequency and high imbalance degree. Regarding the baseline$_{RL}$, used to evaluate the ranking measures, the results obtained are

Table 5 Baseline values obtained for each data set and measure

Data set	Baseline$_{F1}$ ↑				Baseline$_{HL}$ ↓	Baseline$_{RL}$ ↓		Baseline$_{SA}$ ↑
	F1	F1$_m$	Prec$_m$	Rec$_m$	HL	OE	RL	SA
20NG	0.098	0.098	0.051	1.000	0.096	0.948	0.505	0.052
birds	0.288	0.096	0.059	0.267	0.149	0.694	0.316	0.087
cal500	0.478	0.156	0.112	0.282	0.165	0.116	0.212	0.000
corel5k	0.204	0.006	0.003	0.018	0.018	0.776	0.194	0.010
emotions	0.464	0.472	0.312	1.000	0.330	0.555	0.409	0.125
enron	0.463	0.057	0.042	0.095	0.078	0.464	0.141	0.088
fapesp	0.198	0.059	0.033	0.250	0.115	0.857	0.374	0.096
flags	0.699	0.528	0.427	0.714	0.328	0.211	0.220	0.097
image	0.389	0.395	0.247	1.000	0.331	0.710	0.458	0.189
langlog	0.145	0.015	0.008	0.079	0.053	0.857	0.271	0.094
mediamill	0.516	0.027	0.022	0.040	0.036	0.197	0.068	0.056
medical	0.249	0.044	0.027	0.145	0.082	0.720	0.252	0.174
msd-195	0.246	0.051	0.031	0.158	0.078	0.751	0.226	0.082
ohsumed	0.270	0.046	0.029	0.130	0.091	0.716	0.254	0.084
scene	0.302	0.303	0.179	1.000	0.272	0.779	0.473	0.168
slashdot	0.220	0.067	0.038	0.278	0.104	0.845	0.270	0.139
stackex	0.188	0.011	0.006	0.040	0.033	0.737	0.232	0.065
tmc2007	0.447	0.076	0.054	0.136	0.093	0.408	0.163	0.087
yeast	0.576	0.311	0.236	0.500	0.232	0.249	0.211	0.095
yelp8	0.494	0.284	0.203	0.500	0.260	0.411	0.296	0.080

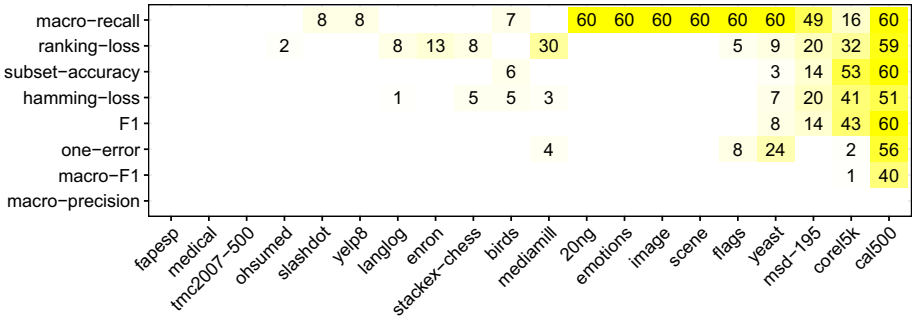

Fig. 1 Number of pairs strategy/base-algorithm that did not perform statistically significantly better than the baselines according to different evaluation measures

inversely correlated with the label cardinality, i.e. the lowest ranking-loss values were observed in data sets with high *lCard*. Finally, as the number of labels and label sets increase, the results obtained for the baseline$_{SA}$ decrease.

Figure 1 summarizes the number of strategy/base-algorithm pairs that did not perform statistically significantly better than the baselines for each data set and evaluation measure. With the exception of *macro-recall*, that can be easily maximized by predicting all labels, and some other measures in the case of the cal500 data set, at least one combination

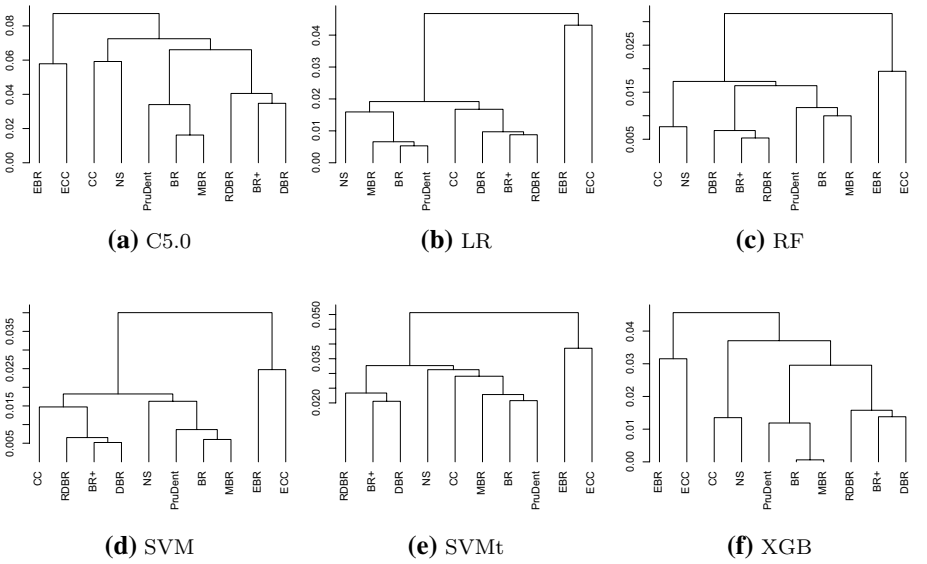

Fig. 2 Similarity of strategies according to their bipartition predictions

strategy/base-algorithm was always able to outperform the baselines for all measures and data sets. However, the considerable number of non-zero entries in Fig. 1 corroborates the claim of Metz et al. (2012) that any new strategy should be compared with others using appropriate multi-label baselines.

5.2 Similarity of strategies

How the base algorithms affect the behavior of the binary transformation strategies is one of the questions investigated in this paper. According to Table 1, it is reasonable to assume that strategies within a group/subgroup are more similar to each other than the rest. However, the transformation strategies work with a base algorithm, which is used to induce the learning models from the transformed data, and its effect over the strategies is unknown so far. Following this rationale, the similarity of strategies using different base algorithms is analyzed in two distinct ways. First, by comparing their predictions, which removes the bias of a specific evaluation measure. Second, by comparing their predictive performance statistically over distinct evaluation measures, which considers particularities of the learning process.

To compare the predictions obtained by the strategies, the Hamming distance (defined in Eq. 7) is computed for each pair of strategies. The result indicates the difference between the predictions, and therefore, the average value over all data sets and repetitions can indicate how similar or distinct any two given strategies are.

Initially, by fixing the base algorithm, the strategies were compared. For such, they were organized according to their similarity using the hierarchical clustering algorithm Averaged-Linkage (Jain and Dubes 1988). Figure 2 shows the hierarchy of strategies for each base algorithm. Similar results are observed regardless of the base algorithm, with some

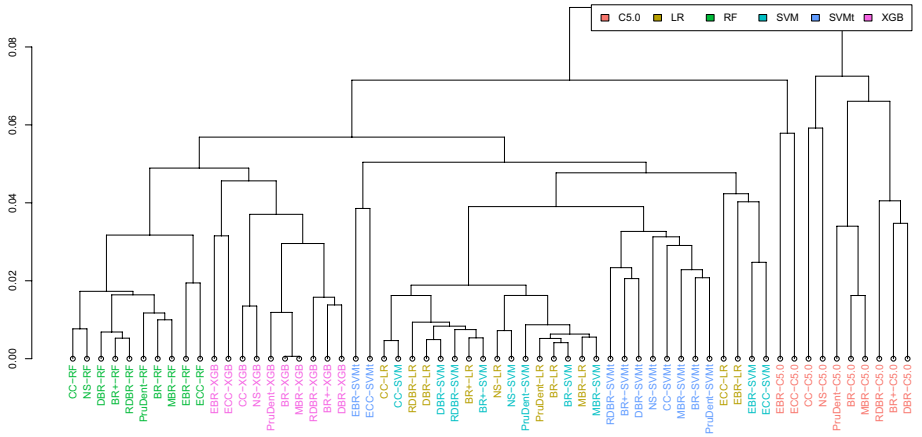

Fig. 3 Similarity of strategies and base algorithms according to their bipartition predictions

exceptions. In summary, the similarity of the predictions follows the intuition of the groups of strategies presented in Table 1.

For all base algorithms, the ensembles EBR and ECC presented the largest difference to all others. The full stacking BR+, DBR and RDBR were grouped together, following different paths, according to the base algorithm. These are the only consensus in the results. Other strategy pairs, such as the chaining CC and NS were the closest strategies only for the base algorithms C5.0, RF and XGB. Similarly, pruned stacking MBR and PruDent were not always in the same group.

Regarding the subgroups, the chaining strategies were more similar to the full stacking for some base algorithms, and to the pruned stacking for others. Pruned stacking was more related to BR than full stacking, which may indicate that the pruning approach impacted the results more than the use of stacking, for these strategies.

Looking at the base algorithms, the use of C5.0 leads to a larger difference among the results obtained by the strategies, and, on the other hand, RF leads to a higher similarity.

Next, when all the strategy/base-algorithm pairs were compared together (Fig. 3), the similarity between the base algorithms could also be compared. The base algorithms RF and XGB produced similar results, and likewise for SVM and LR. In the latter case, the similarity observed was still stronger than the former, since the same strategies using distinct base algorithms were clustered together. On the other hand, SVM and SVMt, despite being the same base algorithm using different hyperparameter values, were not so closely related as SVM and LR were.

With the exception of the ensembles and the SVM and LR base algorithms, all strategies are clustered according to the base algorithm, instead of the opposite, i.e. different variants of the same strategy grouped together. For instance, in this comparison, BR_{RF} is more similar to DBR_{RF}, a full stacking approaching, than to BR_{XGB}. This shows that, for these strategies, their differences might not be strong enough to always be apparent, regardless of the choice of base algorithm.

To identify when small differences in prediction are significant, the pairs of strategies within a group/subgroup were statistically compared. The investigated hypothesis remains that the two distributions are equal such that a high probability means that the two strategies are similar and a low value that the two strategies are indeed dissimilar as one would

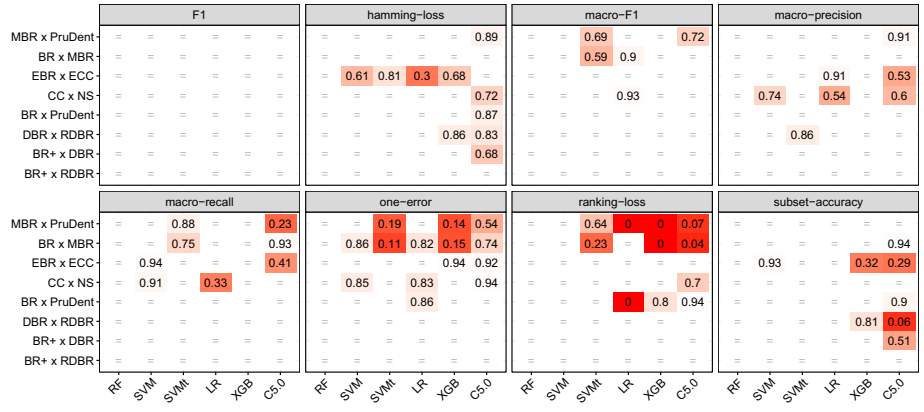

Fig. 4 Rope probabilities from the Bayesian hierarchical test in the comparison of related strategies (y axis) for different base algorithms (x axis). The symbol '=' is used for probabilities greater than 0.95

be interested in. Figure 4 presents the rope probability of different pairs of strategies. The pairs are sorted according to their average values, from the most similar to the most distinct (from the bottom to the top). Likewise, the base algorithms are sorted from left to right.

As previously observed, C5.0 was the base algorithm with the largest number of differences between strategies, whereas RF was the base algorithm with the lowest number of differences. Regardless of the evaluation measure, all pairs were considered similar to each other when RF was used. Additionally, the differences between the strategies were captured in different ways by the evaluation measures. For instance, no differences in *F1* results were observed; the ranking measures were more sensitive when comparing the pruned stacking strategies; and *hamming-loss* and *subset-accuracy* produced clear differences for the ensemble and full stacking strategies.

In summary, the results presented in this section showed that the base algorithms impact the strategies in different ways. Despite all the investigated strategies using the same paradigm (binary transformation), their small differences were captured by the evaluation measures for some of the base algorithms. By varying the base algorithm, a pair of close-related strategies can be seen as more similar, or more distinct, to each other, given a specific evaluation measure. Therefore, it can be concluded that some base algorithms are more dominant than others.

5.3 Analysis of strategies

Following the procedure used in many multi-label studies, the strategies are compared with each other by fixing the base algorithm. As distinct base algorithms are considered, the differences between them can be contrasted. Using the Bayesian hierarchical statistical test, each pair of strategies with the same base algorithm is compared with each other. Figure 5 presents the results of the paired test, varying the base algorithms. For each base algorithm, the strategy whose probability to statistically outperform the other is higher than or equal to 95% is highlighted. Similar algorithms (rope $\geq 95\%$) are represented with an "=" character and an empty value indicates inconclusive results (probabilities $< 95\%$). The pairs of strategies with similar or inclusive results for all base algorithms were removed from the chart.

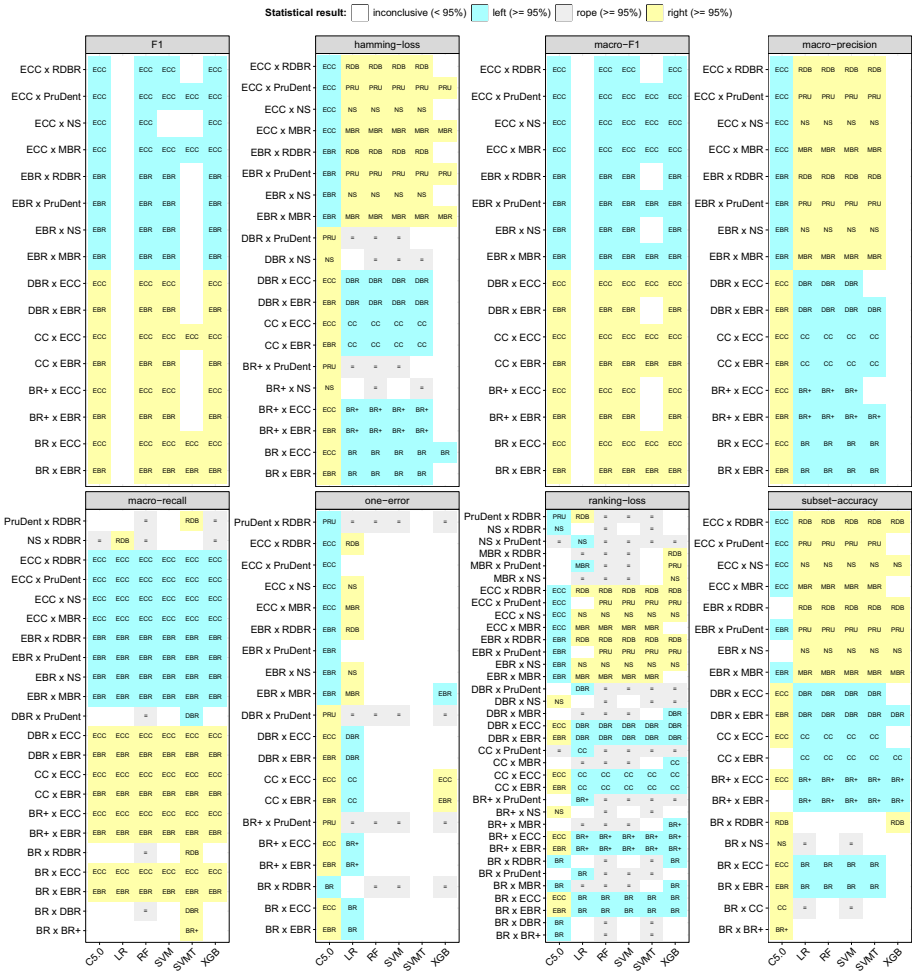

Fig. 5 Best strategy according to the results of the Bayesian hierarchical statistical test. The symbol '=' indicates they are similar with statistical significance

The main discrepancies in the results are observed in relation to the ensemble strategies and the base algorithm C5.0. For C5.0, EBR and ECC outperformed all other strategies for most evaluation measures, whereas for other base algorithms, ensembles were outperformed by different strategies. For the measures *F1*, *macro-F1* and *macro-recall* a more homogeneous result is observed across the base algorithms. In this case, the ensembles are clearly the best choice, probably due to the fact that they internally perform a thresholding calibration that allows them to obtain more balanced precision and recall results regardless of the base algorithm.

To detail the contradictions, Table 6 presents the cases where conflicting probabilities from the statistical test were found across distinct base algorithms. Probabilities indicating that the strategies are similar (rope > 50%) and inconclusive results (all probabilities

Table 6 Divergent probabilities found across the base algorithms in the comparison of the strategies

Measure	Strategies	C5.0		LR		SVMt		XGB	
		Left	Right	Left	Right	Left	Right	Left	Right
HL	CC x DBR	**0.53**	0.00					0.35	**0.59**
Rec_m	BR x NS	**0.68**	0.01			0.04	**0.79**		
	NS x PruDent					**0.53**	0.03	0.08	**0.59**
OE	CC x MBR					**0.74**	0.09	0.3	**0.50**
RL	BR+ x MBR	0.24	**0.74**					**1.00**	0.00
	BR+ x PruDent	0.01	**0.81**	**1.00**	0.00				
	DBR x MBR	0.26	**0.72**					**1.00**	0.00
	DBR x PruDent	0.01	**0.86**	**1.00**	0.00				
	MBR x PruDent	0.05	**0.89**	**1.00**	0.00			0.00	**1.00**
	MBR x RDBR	**0.84**	0.15					0.01	**0.99**
	PruDent x RDBR	**0.97**	0.00	0.00	**1.00**				

Left and right are the probabilities obtained in the Bayesian hierarchical test

$< 50\%$) were omitted from the table, which led to the elimination of the columns relative to base algorithms RF and SVM. The bold markup highlights, for each base algorithm, the highest value and the cases where the probability is greater than or equal to 95% are underlined.

Many observations showed low probabilities at least for one of the base algorithms. This indicates that the differences were not so evident according to the statistical test, even though they are still conflicting. In this sense, the most noticeable differences were observed in the ranking-loss measures, probably because the scores produced by the binary models are more sensitive to variation than the bipartitions.

Regarding the base algorithm, C5.0 shows many strongly significant differences, which reinforces the previous conclusions concerning C5.0 behaving very differently from the other base algorithms. Regarding the strategies, all observed differences are related to pairs of strategies where each comes from a different subgroup, e.g., a chaining strategy against a full stacking strategy.

In conclusion, the comparison of the transformation strategies showed different results, for some measures, according to the base algorithm used. In this particular case, all strategies use a binary transformation, which makes them very similar to each other. Given that differences were still observed, it is reasonable to assume that when different transformation strategies are evaluated, it is important to investigate distinct base algorithms.

5.4 Analysis of base algorithms

Exploring a different perspective, the base algorithms are compared by fixing the strategies. The hypothesis investigated is that for each strategy some specific base algorithms perform better than the rest. Analogous to the previous section, Fig. 6 presents the results of the paired test for base algorithms, in which all base algorithms were compared against each other for each one of the strategies. In this test, for each strategy, the algorithm whose probability to statistically outperform the other is higher than or equal to 95% is

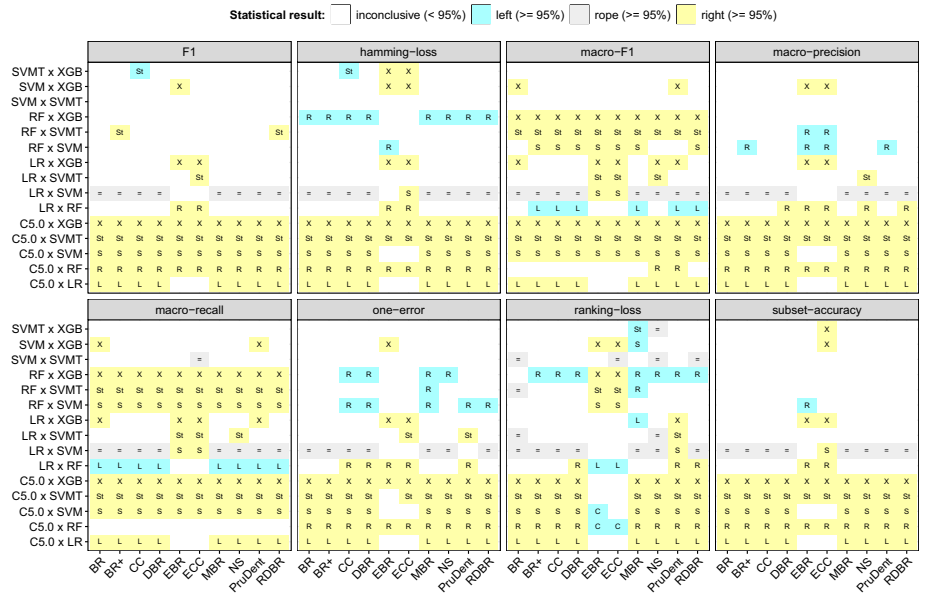

Fig. 6 Best base algorithm according to the results of the Bayesian hierarchical statistical test. The best option for each pair and strategy is indicated by the first letter of the base algorithm, such that C, L, R, S, St and X indicate C5.0, LR, RF, SVM, SVMt and XGB, respectively. The symbol '=' indicates they are similar with statistical significance

highlighted. Similar algorithms (rope $\geq 95\%$) are represented with an "=" character and an empty value indicates inconclusive results (probabilities $< 95\%$).

At a glance, RF and XGB were the dominant base algorithms, regardless of the evaluation measure used. However, they have not been used as the base algorithm in previous studies. In contrast, C5.0, followed by LR, obtained the worst results, despite their popularity in multi-label studies.

Probably due to the lack of diversity in the strategies considered, few variations concerning the best base algorithm were observed. Nevertheless, they are related to the ensembles, the most distinctive strategies among the ones investigated, as noticed in Sect. 5.2. An illustrative example that reinforces the investigated hypothesis is related to the *ranking-loss* measure. For many strategies, RF was the best base algorithm. However, for the ensembles, it was the worst. On the other hand, C5.0, which is not a good choice for many strategies, is a suitable alternative for the ensembles. This is very plausible, as ensemble-based base algorithms, similar to RF, perform better when their base learners are unstable—which is why decision tree induction algorithms (e.g., C5.0) are popular choices inside ensembles of machine learning algorithms. Since the predictions of ensemble-based base algorithms themselves reduce variance, they are not as suitable for ensembles strategies.

For some comparisons and evaluation measures, one of the base algorithms was statistically better than the other regardless of the strategy, mainly when C5.0 was involved, which typically is the worst of the two. In spite of this regularity, the results reinforce the conjecture that the performance of strategies depends on the base algorithm. In particular, the results of the ensemble strategies presented a greater variation, concerning the best

base algorithms, compared to the other strategies. However, additional tests, including a more varied set of strategies, can increase support for this claim.

Some pairs of base algorithms, in particular LR/SVM and SVM/SVMt, presented similar results, with statistical significance, for different evaluation measures. Between LR and SVM, the latter was the best option only for the ensembles, but not for all measures. Comparing SVM and its optimized version, SVMt, despite the fact the latter performed apparently better than the former in terms of *F1*, *macro-F1* and *macro-recall*, the probabilities obtained in the Bayesian test were not greater than or equal to the 95%. Regarding C5.0 and LR, the latter shows clear advantages over the former. Finally, between RF and XGB, the most dominant base algorithms according to the experimental results, the choice between one of them depends on the evaluation measure. XGB was the best option for *macro-F1* and *macro-recall*, while RF was the best for *hamming-loss*, *one-error*, and *ranking-loss*.

In summary, the results presented in this section provide some support for the claim that the choice of base algorithm can strongly influence a strategy's performance. Furthermore, some base algorithms performed better on average than others, which again can influence and even distort comparisons of multi-label learning strategies.

5.5 Combining strategies and base algorithms

The previous analyses showed that the ranking of the best strategies varies according to the base algorithm used. To further investigate this issue, all strategy/base-algorithm pairs are evaluated against each other without distinctions. In order to summarize the 60 pairs (strategy/base-algorithm), Annex A presents the ranking for each pair considering all data sets and the strategies' results using the best base algorithm. The statistical results comparing those strategies are presented in Annex B.

Considering the BR strategy as a more robust baseline, its performance is analysed in relation to the other strategies. For the measures *F1*, *macro-F1* and *macro-recall* the ensembles outperform BR with statistical significance, regardless of the base algorithm. By contrast, BR outperforms them to the measures *hamming-loss*, *macro-precision*, *ranking-loss* and *subset-accuracy*. In relation to the other strategies, there is no case in which BR is completely outperformed by other strategy and vice-versa. Specifically for *one-error* measure, BR_{RF} achieved the best ranking over all combinations and outperformed the other strategies for 4 or 5 base algorithms.

To complement these results, Table 7 presents, for all the selected pairs, the number and percentage of other pairs that were statistically outperformed with a probability greater than or equal to 95%, according to the Bayesian statistical test. The strategies are sorted from top to bottom based on the number of pairs outperformed.

None of the strategies obtained a reasonable performance over all evaluation measures. The highest results are observed for the ensembles using XGB that outperformed more than 90% of the other strategies in terms of *F1*, *macro-F1* and *macro-recall*. Consequently, they are the best ranked pairs of strategy/base-algorithm according to the number of outperformed pairs. The lack of a dominant combination for the other measures shows that all the strategies obtained a good performance for some base algorithms.

Concerning the base algorithms, the best results were obtained mainly by either RF or XGB. Both algorithms are represented in the table by all strategies. In terms of strategies,

Table 7 Selected pairs of strategy/base-algorithm and the percentage of other pairs that were statistically outperformed by them

Strategy/base-algorithm	F1 (%)	$F1_m$ (%)	$Prec_m$ (%)	Rec_m (%)	HL (%)	OE (%)	RL (%)	SA (%)
EBR_{XGB}	90	92	24	92	27	27	19	20
ECC_{XGB}	90	85	25	92	27	22	20	27
$PruDent_{RF}$	14	2	39	0	49	58	58	32
MBR_{LR}	14	25	24	27	32	20	37	25
$RDBR_{SVMt}$	32	46	25	47	29	20	37	39
DBR_{SVMt}	19	36	25	44	34	24	37	36
BR_{RF}	14	2	32	0	47	66	59	32
NS_{RF}	14	12	36	0	41	53	53	39
$BR+_{SVM}$	14	27	24	27	32	20	37	31
CC_{RF}	14	7	31	0	49	53	53	37
MBR_{XGB}	14	46	27	36	34	19	22	32
BR_{XGB}	14	46	27	36	32	20	39	31
$PruDent_{XGB}$	14	47	27	39	34	20	37	31
CC_{XGB}	14	34	27	27	27	17	37	32
NS_{XGB}	14	37	27	27	27	17	37	34
$BR+_{SVMt}$	17	37	25	41	34	24	37	36
MBR_{RF}	14	3	37	0	49	53	47	32
$RDBR_{RF}$	14	0	47	0	42	29	53	37
$BR+_{RF}$	14	3	42	0	42	27	51	37
DBR_{RF}	14	3	39	0	42	49	53	36
EBR_{SVM}	42	64	14	92	14	14	5	7
DBR_{XGB}	14	42	25	34	27	19	37	32
$RDBR_{XGB}$	14	42	27	34	31	17	37	36
$BR+_{XGB}$	14	42	27	36	29	19	37	34
EBR_{RF}	81	27	27	29	22	20	0	20
ECC_{RF}	88	27	27	32	20	19	0	22
NS_{SVMt}	14	32	32	32	31	22	37	34
CC_{SVMt}	14	36	31	34	36	22	37	36

despite being the simplest, BR presented a good performance for the *hamming-loss*, *one-error* and *ranking-loss*.

To sum up, when all strategies/base-algorithms pairs are compared, some strategies appear as dominant for some measures regardless of the choice of base algorithm, such as EBR and ECC for *macro-F1*. On the other hand, for some evaluation measures, the choice of the base algorithm dominates the results, regardless of the chosen strategies, such as RF for *ranking-loss*. Even though all strategies use binary transformation, and consequently are very similar to each other, statistical differences were observed between them. In conclusion, an empirical comparison of multiple transformation strategies together with multiple base algorithms should be considered for any future study proposing new transformations.

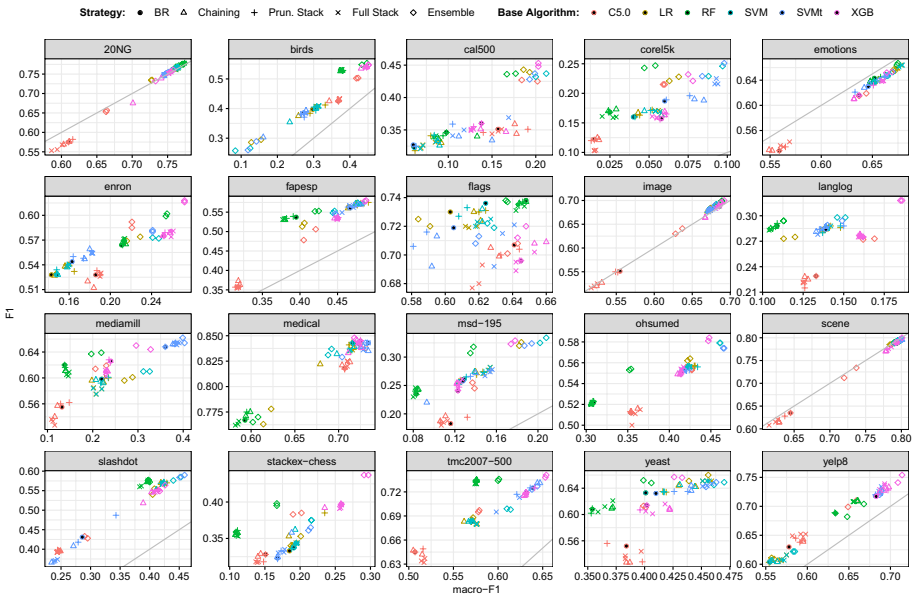

Fig. 7 Comparative results of the measures *F1* and *macro-F1* for all data sets and strategy/base-algorithm pairs

Table 8 Average label problems results over all strategy/base-algorithm pairs

Data set	MLP	WLP
flags	0.03	0.04
ohsumed	0.06	0.07
medical	0.06	0.10
yeast	0.10	0.11
fapesp	0.13	0.19
slashdot	0.15	0.20
birds	0.15	0.23
mediamill	0.17	0.20
msd-195	0.24	0.34
enron	0.29	0.44
stackex-chess	0.32	0.45
langlog	0.34	0.47
cal500	0.37	0.54
corel5k	0.55	0.73

5.6 Label problems

It can be observed in Fig. 7 that the values of *F1* are substantially higher than the values of *macro-F1* for many data sets. This occurs when the value of *F1* is very low for one or more labels. In practice, the least common labels are often behind these differences. As

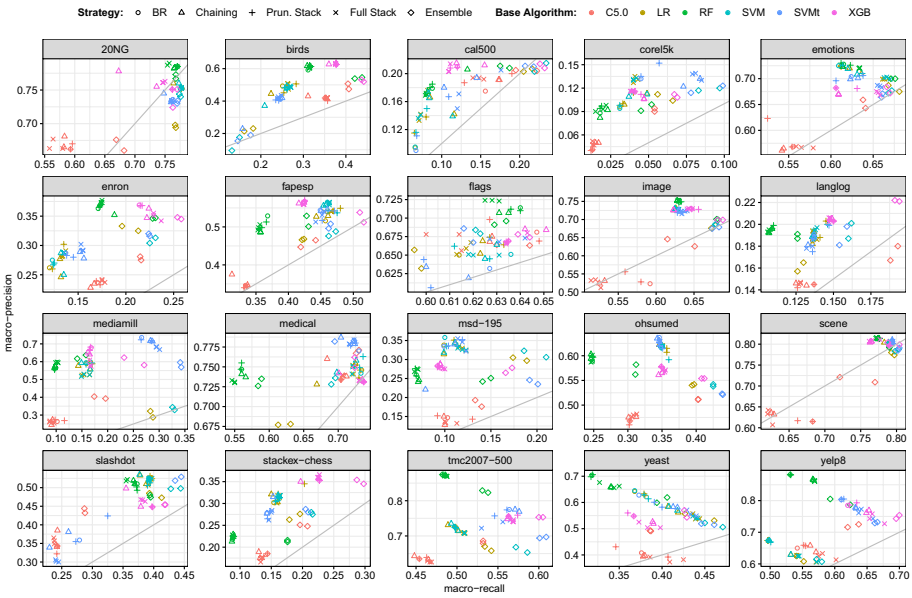

Fig. 8 Comparative results of the measures *macro-precision* and *macro-recall* for all data sets and strategy/base-algorithm pairs

the previously defined label problems MLP and WLP (Eqs. 13 and 14) provide a possible explanation, their average proportions over all strategy/base-algorithm pairs are presented in Table 8.

For the sake of clarity, the data sets without problems were removed from the table. For many data sets, the values obtained paint a clear picture, indicating that many labels were wrongly predicted or even never predicted at all. E.g., in the worst case, on average 73% of the labels from the `corel5k` ($\approx$ 159 labels) were wrongly predicted for all test instances, and 55% ($\approx$ 120 labels) were never predicted. The high values observed for many data sets indicate a problem generated by the binary transformation strategies not previously detected.

This also justifies the high *macro-precision* values in comparison with the *macro-recall* values (Fig. 8). The best results for the measures *F1*, *macro-F1* and *macro-recall* were achieved by the strategy ensembles. Since they use an internal threshold technique for selecting relevant labels, their *recall* is enhanced and, consequently, their *F1* result is also higher. Additional studies are needed to test if this behavior is mainly due to this post-processing used by the ensembles.

5.7 Summary

The main motivation for this study was to obtain a better understanding of how the base algorithm impacts the binary transformation strategies. The results presented in the

previous sections show that the choice of the base algorithm can interfere in the behaviour of binary transformation strategies. Thus, by considering distinct base algorithms, an empirical study involving transformation strategies can become less biased.

Different rankings of strategies and statistical results were obtained by using different base algorithms. This, however, is not common practice in multi-label research. Usually, transformation strategies are proposed and compared using a single base algorithm (Read et al. 2011; Madjarov et al. 2012; Montañes et al. 2014; Moyano et al. 2018). The claim that by segmenting the comparison of base algorithms more consistent results can be obtained (Moyano et al. 2018) might actually be misleading. In addition, across all assessed measures, there was not a single base algorithm that obtained the best results for all strategies. Consequently, performing a comparison of strategies using only one fixed base algorithm should be avoided.

Nevertheless, it is still valid to compare the strategies using a fixed base algorithm, since it can help with understanding the scenarios in which a strategy is improved. For instance, a clear superiority of the ensembles EBR and ECC, regardless of the evaluation measure, was observed when the base algorithm C5.0 was used. On the other hand, when using the LR and RF algorithms, ensemble strategies did not perform so well, showing that for a given base algorithm some strategies might not be suitable. Even though some base algorithms might obtain a better overall performance than others, the diversity of base algorithms is valid to determine the conditions in which each strategy is convenient. Furthermore, although predictive performance is very important, there are reasons one may consider different base classifiers. For example, decision trees provide good interpretation, logistic regression provides good probability estimates. Therefore, it is useful to consider the relative performance difference rather than simply the top performance.

Considering the large experimental scenario evaluated, the hyperparameter tuning procedure adopted was simple and did not achieve the best results for the optimized measure. The use of the SVMt base algorithm produced distinct results when compared to SVM, but when compared to others, such as RF and XGB, the SVMt results were more similar to SVM. Therefore, in this context, hyperparameter tuning can be seen secondary to base algorithm selection, provided reasonable default parameter settings can be identified for the selected base algorithm. However, we remark that this indeed depends on the model class in question; in which some models are more sensitive to initial hyperparameter settings than others. Ideally, if computational power allows for it, then the base algorithms should be tuned as part of the base-algorithm selection process, especially if the performance difference between them is not great. Of course, for large scale experimental comparisons, this may not be feasible due to the extra degree of complexity implied.

Auto-ML for MLC (de Sá et al. 2017; Wever et al. 2019) can be used to find the best combination between strategies and base algorithm. Furthermore, it can tune the hyperparameters of both of them, as well as the pipeline of the solution, in order to bring the best results for a given problem. Thus, Auto-ML tools is an answer to the question of how to give advice which multi-label classifier and base algorithm to use. However, it demands high computational resources, which may be limiting its use.

Regarding the closely-related strategies (BR and pruned stacking; chaining; full stacking; and the ensembles investigated here), their differences are shown to be subtle and circumstantial. Given the relatively small number of data sets that have been considered in empirical studies, finding characteristics of a problem that distinguishes strategies is not a

Table 9 Suggestion of binary transformation strategies to be picked in empirical experiments

Measure	Ranking of suggested strategies				
	1	2	3	4	5
F1	EBR	MBR	RDBR	BR	CC
macro-F1	EBR	RDBR	MBR	CC	BR
macro-precision	MBR	NS	RDBR	BR	ECC
macro-recall	EBR	RDBR	MBR	CC	BR
hamming-loss	PruDent	BR	CC	BR+	EBR
one-error	BR	PruDent	NS	DBR	EBR
ranking-loss	BR	NS	PruDent	DBR	ECC
subset-accuracy	RDBR	NS	PruDent	BR	ECC

The recommendation is based on criteria such as dissimilarity and the strategies' average ranking considering all base algorithms

trivial task. Thus, the choice of a strategy between those close-related might also be seen as merely a matter of convenience, potentially influenced by other performance considerations, such as memory or runtime cost.

The differences between strategies from distinct groups are very consistent for the different evaluation criteria. Therefore, for empirical studies involving binary transformation strategies in MLC, we strongly recommend the use of strategies from different groups, as well as various base algorithms. The selection between strategies in the same group is not an easy task. However, it is important to provide some guidance concerning which one to use. We decided to use the average ranking considering all base algorithms ("Appendix 1").

Table 9 summarises the experimental results, describing good strategies for different evaluation measures. In practical applications, RF and XGB should be considered as base algorithms, in addition to the usual favorites, which include C5.0, LR, and SVM. We note that if the median rank for each base algorithm or another criterion were adopted, different recommendations would probably be observed but the predicted performance obtained would not be expected to be very different.

6 Conclusion

This paper presented an extensive experimental evaluation of binary transformation strategies for multi-label classification. Different perspectives were considered in addition to the traditional approach of selecting just a single base algorithm when comparing multi-label strategies. Thus, bipartition predictions were compared, strategies were compared for fixed base algorithms, base algorithm were compared for fixed strategies, and all possible pairs of strategy and base algorithm were compared with each other.

The main conclusions to draw from this study are:

- Binary transformation strategies are strongly influenced by the base algorithm used. Consequently, empirical studies should always consider distinct and diversified base algorithms.

- RF and XGB, which showed high predictive performance across a number of strategies, should be considered in the subset of base algorithms selected to perform an empirical study in MLC.
- The investigated strategies and base algorithms always either misclassified or were unable to predict some of the labels. So far this problem has been ignored, mainly because the traditional evaluation measures are not able to capture this problem. Nevertheless, this is a problem that requires more attention in future studies.

More specific conclusions for multi-label strategies and evaluation measures include:

- Ensembles using internal threshold selection obtained good results for *F1*, *macro-F1* and *macro-recall*.
- Despite being considered a baseline in many studies, BR obtained the best predictive performance for the ranking measures, *one-error* and *ranking-loss*. In addition, BR obtained good results for the *macro-precision* and *hamming-loss* measures, depending on the choice of base algorithm.
- The full stacking strategies and the NS strategy, which uses a subset correction procedure, obtained the best results for the *subset-accuracy* measure.

Future work includes investigating the impact of the base algorithm on other transformations such as the label-powerset method. Recommendation of combinations of a strategy and a base algorithm based on a desired measure, as well as data set characteristics is another promising direction. Finally, the two types of label prediction failure, MLP and WLP, need to be researched in more depth.

Acknowledgements This work was financially supported by CNPq (Processes 305291/2017-3 and 152098/2016-0), FAPESP (Processes 2016/18615-0, 2013/07375-0 and 2012/22608-8), CAPES and Intel. The experiments were performed using the computational resources of CeMEAI-FAPESP, Proc. 13/07375-0.

Compliance with ethical standards

Conflict of Interest The authors declare that they have no conflict of interest.

Appendix 1: Best strategies/base algorithms

This section presents the strategy/base-algorithm's ranking over all data sets (Figs. 9, 10, 11, 12, 13, 14, 15 and 16) and the performance value obtained for each strategy when combined with the best base algorithm (Tables 10, 11, 12, 13, 14, 15, 16 and 17). The median ranking is used to select the base-algorithm for each strategy.

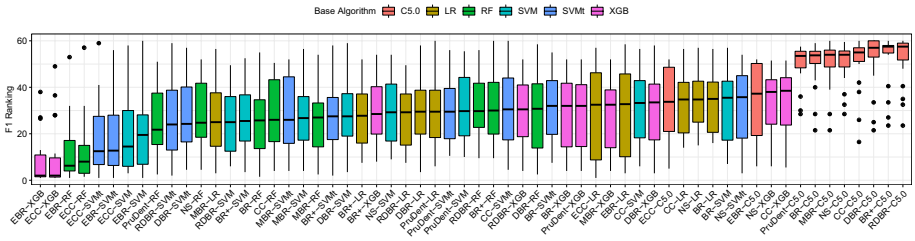

Fig. 9 Strategy/base-algorithm's rankings for the *F1* measure

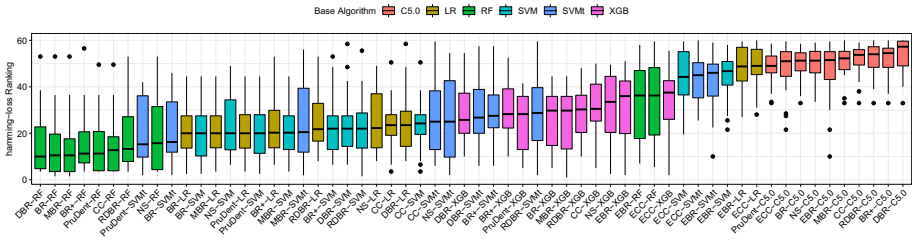

Fig. 10 Strategy/base-algorithm's rankings for the *hamming-loss* measure

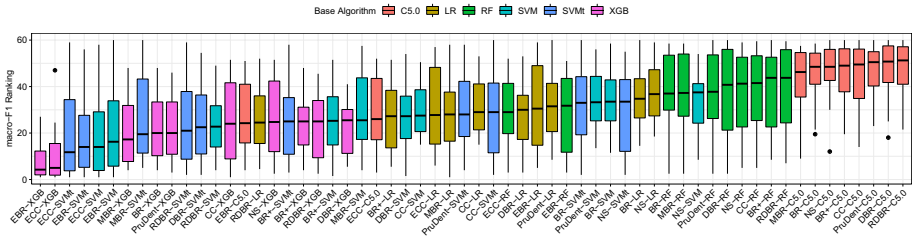

Fig. 11 Strategy/base-algorithm's rankings for the *macro-F1* measure

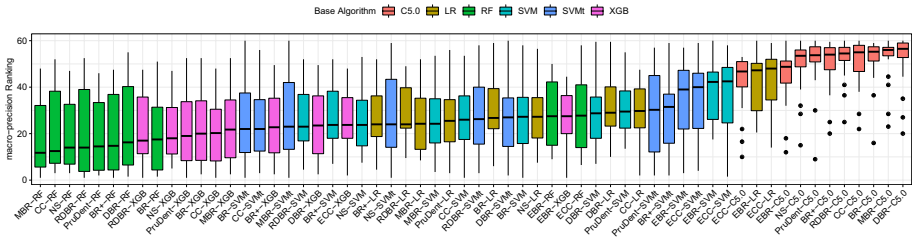

Fig. 12 Strategy/base-algorithm's rankings for the *macro-precision* measure

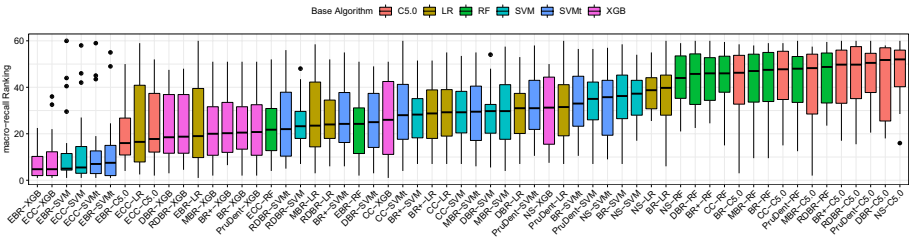

Fig. 13 Strategy/base-algorithm's rankings for the *macro-recall* measure

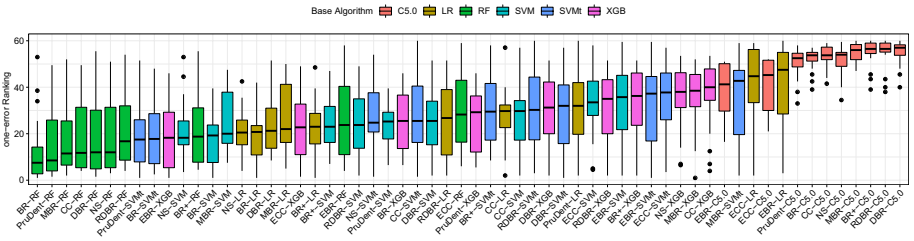

Fig. 14 Strategy/base-algorithm's rankings for the *one-error* measure

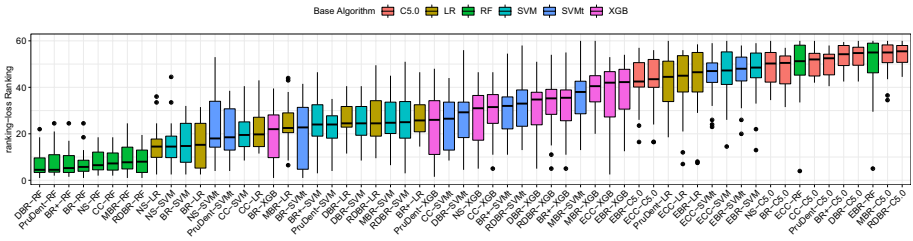

Fig. 15 Strategy/base-algorithm's rankings for the *ranking-loss* measure

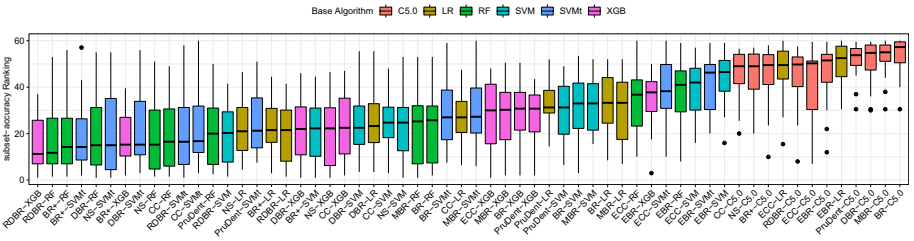

Fig. 16 Strategy/base-algorithm's rankings for the *subset-accuracy* measure

Table 10 Results of best strategies for the *F1* ↑ measure

Data set	EBR_{XGB}	ECC_{XGB}	$PruDent_{RF}$	MBR_{LR}	$RDBR_{SVMt}$	DBR_{SVMt}	BR_{RF}	NS_{RF}	$BR+_{SVM}$	CC_{RF}
20NG	0.7612	0.7323	0.7781	0.7659	0.7474	0.7477	**0.7782**	0.7735	0.7663	0.7725
	(0.003)	(0.008)	(0.002)	(0.003)	(0.004)	(0.005)	**(0.003)**	(0.002)	(0.003)	(0.003)
birds	0.5471	**0.5472**	0.5283	0.3837	0.3835	0.3862	0.5294	0.5263	0.4088	0.5277
	(0.022)	**(0.017)**	(0.018)	(0.134)	(0.111)	(0.110)	(0.023)	(0.023)	(0.119)	(0.023)
cal500	0.4476	**0.4529**	0.3414	0.3414	0.3693	0.3406	0.3458	0.3395	0.3261	0.3362
	(0.004)	**(0.004)**	(0.006)	(0.010)	(0.009)	(0.011)	(0.006)	(0.012)	(0.009)	(0.006)
corel5k	0.2211	**0.2317**	0.1680	0.1717	0.2160	0.2247	0.1675	0.1735	0.1696	0.1698
	(0.003)	**(0.003)**	(0.004)	(0.005)	(0.003)	(0.005)	(0.004)	(0.006)	(0.003)	(0.003)
emotions	0.6521	0.6528	0.6454	0.6366	0.6468	0.6473	0.6425	0.6377	**0.6640**	0.6377
	(0.010)	(0.012)	(0.013)	(0.021)	(0.019)	(0.015)	(0.022)	(0.019)	**(0.013)**	(0.019)
enron	0.6157	**0.6176**	0.5687	0.5291	0.5540	0.5549	0.5642	0.5646	0.5405	0.5670
	(0.009)	**(0.009)**	(0.015)	(0.007)	(0.012)	(0.013)	(0.013)	(0.011)	(0.009)	(0.015)
fapesp	**0.5795**	0.5778	0.5339	0.5754	0.5617	0.5672	0.5370	0.5301	0.5717	0.5301
	(0.018)	(0.023)	(0.032)	(0.027)	(0.024)	(0.033)	(0.038)	(0.035)	(0.024)	(0.035)
flags	0.7117	0.7198	0.7356	0.7310	0.7209	0.6918	**0.7377**	0.7364	0.7227	0.7371
	(0.016)	(0.018)	(0.013)	(0.013)	(0.015)	(0.029)	**(0.018)**	(0.014)	(0.007)	(0.016)
image	**0.7000**	0.6961	0.6905	0.6807	0.6756	0.6843	0.6893	0.6890	0.6812	0.6890
	(0.008)	(0.008)	(0.007)	(0.010)	(0.027)	(0.012)	(0.007)	(0.006)	(0.009)	(0.006)
langlog	**0.3181**	0.3179	0.2875	0.2861	0.2845	0.2779	0.2883	0.2846	0.2857	0.2874
	(0.008)	(0.010)	(0.015)	(0.010)	(0.010)	(0.008)	(0.010)	(0.010)	(0.008)	(0.017)
mediamill	0.6436	0.6496	0.6161	0.6005	**0.6539**	0.6514	0.6204	0.6114	0.5844	0.6108
	(0.001)	(0.001)	(0.001)	(0.001)	**(0.002)**	(0.002)	(0.001)	(0.001)	(0.002)	(0.001)
medical	**0.8481**	0.8461	0.7746	0.8425	0.8397	0.8426	0.7674	0.7711	0.8364	0.7754
	(0.010)	(0.012)	(0.010)	(0.004)	(0.016)	(0.016)	(0.013)	(0.013)	(0.005)	(0.011)
msd-195	0.3233	**0.3293**	0.2404	0.2737	0.2784	0.2669	0.2353	0.2427	0.2771	0.2420
	(0.010)	**(0.008)**	(0.009)	(0.007)	(0.010)	(0.007)	(0.009)	(0.008)	(0.005)	(0.008)

Table 10 (continued)

Data set	EBR_{XGB}	ECC_{XGB}	$PruDent_{RF}$	MBR_{LR}	$RDBR_{SVMt}$	DBR_{SVMt}	BR_{RF}	NS_{RF}	$BR+_{SVM}$	CC_{RF}
ohsumed	0.5812	**0.5842**	0.5227	0.5570	0.5524	0.5524	0.5220	0.5204	0.5562	0.5207
	(0.003)	**(0.003)**	(0.004)	(0.002)	(0.002)	(0.003)	(0.003)	(0.003)	(0.002)	(0.003)
scene	**0.8004**	0.7898	0.7908	0.7904	0.7965	0.7955	0.7893	0.7883	0.7921	0.7879
	(0.002)	(0.010)	(0.005)	(0.004)	(0.005)	(0.006)	(0.004)	(0.001)	(0.005)	(0.000)
slashdot	0.5645	0.5493	**0.5764**	0.5697	0.3688	0.3742	0.5757	0.5740	0.5668	0.5721
	(0.006)	(0.007)	**(0.007)**	(0.012)	(0.228)	(0.228)	(0.008)	(0.007)	(0.009)	(0.005)
stackex	**0.4372**	0.4366	0.3611	0.3846	0.3296	0.3355	0.3591	0.3580	0.3434	0.3546
	(0.011)	(0.011)	(0.008)	(0.013)	(0.050)	(0.039)	(0.008)	(0.007)	(0.008)	(0.009)
tmc2007	0.7381	**0.7407**	0.7341	0.6835	0.6947	0.7186	0.7340	0.7352	0.6802	0.7345
	(0.002)	**(0.002)**	(0.002)	(0.002)	(0.012)	(0.015)	(0.001)	(0.002)	(0.002)	(0.002)
yeast	0.6561	**0.6569**	0.6023	0.6342	0.6468	0.6430	0.6082	0.6105	0.6513	0.6095
	(0.004)	**(0.005)**	(0.003)	(0.003)	(0.006)	(0.004)	(0.003)	(0.005)	(0.004)	(0.005)
yelp8	0.7410	**0.7540**	0.6889	0.6039	0.7301	0.7340	0.6885	0.7087	0.6072	0.7091
	(0.004)	**(0.003)**	(0.002)	(0.003)	(0.003)	(0.004)	(0.003)	(0.003)	(0.005)	(0.003)

Table 11 Results of best strategies for the *hamming-loss* ↓ measure

Data set	PruDent$_{RF}$	BR$_{RF}$	MBR$_{RF}$	BR+$_{RF}$	DBR$_{RF}$	RDBR$_{RF}$	CC$_{RF}$	NS$_{RF}$	EBR$_{RF}$	ECC$_{RF}$
20NG	**0.0228**	**0.0228**	**0.0228**	0.0242	0.0229	0.0246	0.0234	0.0233	0.0238	0.0243
	(0.000)	**(0.000)**	**(0.000)**	(0.000)	(0.000)	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)
birds	0.0922	0.0924	0.0927	**0.0921**	**0.0921**	**0.0921**	0.0927	0.0927	0.1094	0.1107
	(0.004)	(0.003)	(0.003)	**(0.004)**	**(0.004)**	**(0.003)**	(0.004)	(0.004)	(0.003)	(0.006)
cal500	0.1686	0.1694	0.1694	0.1668	**0.1666**	0.1669	0.1683	0.1962	0.1932	0.1876
	(0.001)	(0.001)	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.003)	(0.004)	(0.003)
corel5k	**0.0167**	**0.0167**	**0.0167**	0.0169	0.0169	0.0169	**0.0167**	0.0181	0.0215	0.0218
	(0.000)	**(0.000)**	**(0.000)**	(0.000)	(0.000)	(0.000)	**(0.000)**	(0.000)	(0.000)	(0.000)
emotions	**0.1865**	**0.1865**	0.1872	0.1868	0.1873	0.1867	0.1898	0.1898	0.1880	0.1889
	(0.007)	**(0.009)**	(0.008)	(0.008)	(0.008)	(0.009)	(0.009)	(0.009)	(0.005)	(0.005)
enron	**0.0570**	0.0571	0.0572	0.0574	0.0573	0.0578	0.0572	0.0638	0.0605	0.0614
	(0.002)	(0.002)	(0.002)	(0.001)	(0.001)	(0.001)	(0.002)	(0.002)	(0.001)	(0.001)
fapesp	0.0630	**0.0625**	**0.0625**	0.0634	0.0635	0.0631	0.0635	0.0635	0.0688	0.0688
	(0.004)	**(0.005)**	**(0.004)**	(0.006)	(0.005)	(0.006)	(0.004)	(0.004)	(0.005)	(0.004)
flags	0.2366	0.2341	**0.2338**	0.2370	0.2380	0.2364	0.2350	0.2353	0.2368	0.2363
	(0.012)	(0.012)	**(0.016)**	(0.014)	(0.013)	(0.012)	(0.012)	(0.012)	(0.012)	(0.009)
image	**0.1457**	0.1461	0.1465	0.1458	0.1458	0.1462	0.1459	0.1459	0.1551	0.1547
	(0.004)	(0.003)	(0.002)	(0.004)	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)
langlog	0.0435	0.0435	**0.0434**	0.0438	0.0439	0.0438	0.0436	0.0437	0.0503	0.0502
	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
mediamill	0.0275	**0.0273**	**0.0273**	0.0280	0.0281	0.0281	0.0277	0.0277	0.0278	0.0282
	(0.000)	**(0.000)**	**(0.000)**	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
medical	0.0263	0.0271	0.0270	0.0275	0.0275	0.0272	**0.0262**	0.0267	0.0293	0.0301
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.001)
msd-195	0.0717	0.0716	0.0717	0.0723	0.0728	0.0721	**0.0715**	**0.0715**	0.0906	0.0929
	(0.001)	(0.000)	(0.001)	(0.001)	(0.001)	(0.001)	**(0.000)**	**(0.000)**	(0.001)	(0.001)

Table 11 (continued)

Data set	PruDent$_{RF}$	BR$_{RF}$	MBR$_{RF}$	BR+$_{RF}$	DBR$_{RF}$	RDBR$_{RF}$	CC$_{RF}$	NS$_{RF}$	EBR$_{RF}$	ECC$_{RF}$
ohsumed	**0.0576**	0.0577	**0.0576**	0.0578	0.0577	0.0577	0.0578	0.0578	0.0615	0.0616
	(0.000)	(0.000)	**(0.000)**	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
scene	**0.0747**	0.0751	0.0748	0.0751	0.0751	0.0755	0.0753	0.0752	0.0782	0.0778
	(0.001)	(0.001)	(0.002)	(0.001)	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
slashdot	0.0530	0.0531	0.0530	0.0547	**0.0528**	0.0545	0.0533	0.0531	0.0583	0.0591
	(0.001)	(0.001)	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.001)	(0.002)	(0.002)
stackex	**0.0257**	0.0258	**0.0257**	0.0259	0.0259	0.0258	0.0259	0.0258	0.0354	0.0356
	(0.000)	(0.000)	**(0.000)**	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.001)	(0.000)
tmc2007	0.0462	0.0462	**0.0461**	0.0465	0.0466	0.0466	0.0462	**0.0461**	0.0502	0.0504
	(0.000)	(0.000)	**(0.000)**	(0.000)	(0.000)	(0.000)	(0.000)	**(0.000)**	(0.000)	(0.000)
yeast	0.1908	0.1902	**0.1901**	0.1921	0.1979	0.1941	0.1919	0.1918	0.1920	0.1951
	(0.002)	(0.002)	**(0.002)**	(0.002)	(0.002)	(0.004)	(0.002)	(0.002)	(0.002)	(0.003)
yelp8	0.1426	0.1421	0.1421	0.1393	0.1394	0.1393	**0.1369**	0.1370	0.1616	0.1571
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.007)

Table 12 Results of best strategies for *macro-F1* ↑ measure

Data set	EBR_{XGB}	ECC_{XGB}	MBR_{XGB}	BR_{XGB}	$PruDent_{XGB}$	$RDBR_{SVMt}$	DBR_{SVMt}	CC_{XGB}	NS_{XGB}	$BR+_{SVMt}$
20NG	**0.7582**	0.7333	0.7546	0.7544	0.7544	0.7450	0.7452	0.7008	0.7409	0.7451
	(0.002)	(0.005)	(0.003)	(0.003)	(0.003)	(0.004)	(0.005)	(0.004)	(0.003)	(0.005)
birds	0.4511	**0.4512**	0.4459	0.4459	0.4459	0.2766	0.2788	0.4465	0.4307	0.2699
	(0.029)	**(0.032)**	(0.035)	(0.035)	(0.035)	(0.124)	(0.125)	(0.035)	(0.037)	(0.125)
cal500	**0.2035**	0.2031	0.1378	0.1380	0.1288	0.1688	0.1049	0.1369	0.1661	0.1185
	(0.007)	(0.008)	(0.005)	(0.005)	(0.004)	(0.011)	(0.008)	(0.005)	(0.010)	(0.006)
corel5k	0.0780	0.0751	0.0608	0.0577	0.0572	0.0917	0.0932	0.0598	0.0605	**0.0940**
	(0.002)	(0.004)	(0.003)	(0.002)	(0.003)	(0.004)	(0.006)	(0.004)	(0.004)	**(0.008)**
emotions	0.6644	**0.6694**	0.6369	0.6369	0.6514	0.6626	0.6655	0.6322	0.6324	0.6634
	(0.009)	**(0.009)**	(0.010)	(0.010)	(0.011)	(0.014)	(0.009)	(0.008)	(0.008)	(0.011)
enron	**0.2725**	0.2722	0.2523	0.2524	0.2535	0.1817	0.1820	0.2578	0.2508	0.1826
	(0.006)	(0.006)	(0.008)	(0.008)	(0.008)	(0.009)	(0.009)	(0.006)	(0.008)	(0.009)
fapesp	**0.4884**	0.4863	0.4498	0.4498	0.4513	0.4712	0.4663	0.4500	0.4462	0.4559
	(0.025)	(0.028)	(0.029)	(0.029)	(0.027)	(0.022)	(0.038)	(0.030)	(0.028)	(0.041)
flags	0.6431	0.6483	0.6449	0.6449	0.6435	0.6415	0.6293	**0.6601**	0.6531	0.6335
	(0.030)	(0.032)	(0.028)	(0.028)	(0.026)	(0.042)	(0.031)	**(0.026)**	(0.028)	(0.031)
image	**0.6923**	0.6885	0.6851	0.6851	0.6891	0.6714	0.6795	0.6666	0.6666	0.6765
	(0.008)	(0.009)	(0.008)	(0.008)	(0.009)	(0.027)	(0.012)	(0.009)	(0.009)	(0.019)
langlog	0.1854	**0.1865**	0.1616	0.1616	0.1616	0.1373	0.1342	0.1603	0.1604	0.1375
	(0.015)	**(0.016)**	(0.014)	(0.014)	(0.014)	(0.009)	(0.010)	(0.013)	(0.013)	(0.010)
mediamill	0.3277	0.2960	0.2355	0.2396	0.2387	**0.3913**	0.3865	0.2331	0.2306	0.3874
	(0.004)	(0.006)	(0.004)	(0.005)	(0.005)	**(0.003)**	(0.004)	(0.004)	(0.005)	(0.005)
medical	0.7185	0.7198	0.7253	0.7253	0.7253	0.7255	**0.7338**	0.7262	0.7107	0.7284
	(0.021)	(0.022)	(0.024)	(0.024)	(0.024)	(0.017)	**(0.017)**	(0.022)	(0.023)	(0.021)
msd-195	0.1744	**0.1779**	0.1234	0.1234	0.1241	0.1555	0.1486	0.1216	0.1220	0.1539
	(0.009)	**(0.008)**	(0.009)	(0.009)	(0.008)	(0.012)	(0.008)	(0.009)	(0.009)	(0.006)

Table 12 (continued)

Data set	EBR_{XGB}	ECC_{XGB}	MBR_{XGB}	BR_{XGB}	$PruDent_{XGB}$	$RDBR_{SVMt}$	DBR_{SVMt}	CC_{XGB}	NS_{XGB}	$BR+_{SVMt}$
ohsumed	0.4465	**0.4479**	0.4137	0.4139	0.4139	0.4177	0.4180	0.4129	0.4095	0.4180
	(0.003)	**(0.004)**	(0.002)	(0.002)	(0.002)	(0.004)	(0.003)	(0.004)	(0.003)	(0.004)
scene	**0.8009**	0.7945	0.7920	0.7920	0.7920	0.8005	0.7989	0.7777	0.7795	0.7980
	(0.004)	(0.013)	(0.010)	(0.010)	(0.010)	(0.006)	(0.004)	(0.001)	(0.002)	(0.005)
slashdot	**0.4258**	0.4174	0.4148	0.4148	0.4148	0.2385	0.2443	0.3893	0.4031	0.2399
	(0.008)	(0.015)	(0.008)	(0.008)	(0.008)	(0.227)	(0.230)	(0.011)	(0.007)	(0.228)
stackex	**0.2968**	0.2907	0.2580	0.2586	0.2586	0.1687	0.1778	0.2577	0.2349	0.1767
	(0.009)	(0.011)	(0.010)	(0.010)	(0.010)	(0.052)	(0.038)	(0.009)	(0.011)	(0.051)
tmc2007	0.6526	**0.6543**	0.6406	0.6323	0.6323	0.5978	0.6319	0.6286	0.6313	0.6237
	(0.011)	**(0.012)**	(0.014)	(0.013)	(0.013)	(0.008)	(0.023)	(0.011)	(0.013)	(0.028)
yeast	0.4328	0.4259	0.4019	0.4019	0.3961	**0.4589**	0.4512	0.4221	0.4223	0.4546
	(0.004)	(0.009)	(0.006)	(0.006)	(0.006)	**(0.007)**	(0.005)	(0.004)	(0.004)	(0.008)
yelp8	0.7087	**0.7141**	0.6970	0.6832	0.6846	0.6896	0.6893	0.6921	0.6982	0.6864
	(0.005)	**(0.004)**	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)	(0.004)	(0.005)	(0.004)

Table 13 Results of best strategies for *macro-precision* ↑ measure

Data set	MBR_{RF}	CC_{RF}	$RDBR_{RF}$	NS_{RF}	$BR+_{RF}$	$PruDent_{RF}$	DBR_{RF}	BR_{RF}	ECC_{XGB}	EBR_{XGB}
20NG	0.7851	0.7824	0.7871	0.7827	**0.7895**	0.7851	0.7852	0.7849	0.7236	0.7511
	(0.002)	(0.003)	(0.009)	(0.002)	**(0.008)**	(0.003)	(0.003)	(0.003)	(0.014)	(0.003)
birds	**0.6179**	0.6161	0.6043	0.6167	0.6047	0.6131	0.6046	0.5916	0.5311	0.5203
	(0.078)	(0.054)	(0.089)	(0.056)	(0.087)	(0.075)	(0.087)	(0.080)	(0.060)	(0.053)
cal500	0.1852	0.1745	0.1718	0.2061	0.1690	0.1797	0.1695	0.1791	0.2084	**0.2136**
	(0.012)	(0.019)	(0.021)	(0.019)	(0.019)	(0.014)	(0.015)	(0.016)	(0.012)	**(0.008)**
corel5k	0.0974	0.0821	0.0897	0.0915	0.0976	0.0937	0.0929	0.0882	**0.1121**	0.1082
	(0.010)	(0.011)	(0.016)	(0.011)	(0.009)	(0.012)	(0.006)	(0.009)	**(0.012)**	(0.007)
emotions	**0.7299**	0.7232	0.7205	0.7232	0.7209	0.7258	0.7203	0.7281	0.6843	0.6724
	(0.011)	(0.013)	(0.011)	(0.013)	(0.010)	(0.010)	(0.011)	(0.010)	(0.011)	(0.015)
enron	0.3651	0.3697	**0.3769**	0.3524	0.3725	0.3696	0.3733	0.3625	0.3450	0.3484
	(0.017)	(0.024)	**(0.019)**	(0.024)	(0.018)	(0.019)	(0.017)	(0.019)	(0.017)	(0.019)
fapesp	0.5127	0.4859	0.5057	0.4859	0.4980	0.4984	0.4922	0.5291	0.5119	**0.5378**
	(0.082)	(0.069)	(0.088)	(0.069)	(0.084)	(0.072)	(0.081)	(0.087)	(0.028)	**(0.039)**
flags	0.7101	0.7065	0.7227	0.7069	0.7235	0.7117	**0.7238**	0.7145	0.6743	0.6785
	(0.064)	(0.057)	(0.062)	(0.057)	(0.059)	(0.059)	**(0.059)**	(0.063)	(0.031)	(0.044)
image	0.7472	0.7519	0.7518	0.7520	**0.7533**	0.7486	0.7504	0.7482	0.6990	0.6987
	(0.005)	(0.007)	(0.007)	(0.007)	**(0.008)**	(0.008)	(0.008)	(0.007)	(0.010)	(0.008)
langlog	0.1988	0.1961	0.1953	0.1918	0.1935	0.1964	0.1935	0.1964	0.2206	**0.2222**
	(0.018)	(0.027)	(0.015)	(0.014)	(0.015)	(0.020)	(0.015)	(0.018)	(0.031)	**(0.024)**
mediamill	0.5550	0.5732	0.5949	0.5745	0.5944	0.5601	0.5985	0.5599	**0.6225**	0.5805
	(0.013)	(0.019)	(0.015)	(0.014)	(0.016)	(0.011)	(0.020)	(0.014)	**(0.022)**	(0.016)
medical	0.7471	0.7370	0.7312	0.7405	0.7329	0.7554	0.7303	0.7468	**0.7624**	0.7490
	(0.028)	(0.037)	(0.036)	(0.033)	(0.035)	(0.023)	(0.033)	(0.030)	**(0.038)**	(0.036)
msd-195	0.2572	0.2406	0.2735	0.2464	0.2756	0.2757	0.2681	0.2605	**0.2783**	0.2647
	(0.033)	(0.028)	(0.040)	(0.028)	(0.037)	(0.033)	(0.032)	(0.026)	**(0.023)**	(0.030)

Table 13 (continued)

Data set	MBR_{RF}	CC_{RF}	$RDBR_{RF}$	NS_{RF}	$BR+_{RF}$	$PruDent_{RF}$	DBR_{RF}	BR_{RF}	ECC_{XGB}	EBR_{XGB}
ohsumed	0.5991	0.5941	0.6007	**0.6028**	0.5866	0.5974	0.5894	0.5894	0.5538	0.5540
	(0.039)	(0.037)	(0.035)	**(0.026)**	(0.022)	(0.025)	(0.027)	(0.017)	(0.014)	(0.012)
scene	0.8130	0.8124	0.8136	0.8133	**0.8159**	0.8136	0.8145	0.8129	0.8038	0.7990
	(0.004)	(0.003)	(0.003)	(0.003)	**(0.004)**	(0.004)	(0.002)	(0.001)	(0.014)	(0.005)
slashdot	0.4932	0.5187	**0.5216**	0.5100	0.5184	0.5016	0.5113	0.5099	0.4540	0.4541
	(0.027)	(0.036)	**(0.052)**	(0.029)	(0.052)	(0.028)	(0.041)	(0.041)	(0.037)	(0.016)
stackex	0.2196	0.2132	0.2211	0.2290	0.2259	0.2235	0.2277	0.2180	**0.3540**	0.3453
	(0.029)	(0.017)	(0.025)	(0.023)	(0.025)	(0.023)	(0.020)	(0.033)	**(0.019)**	(0.018)
tmc2007	0.8738	0.8710	0.8697	0.8716	0.8696	0.8737	0.8691	**0.8740**	0.7526	0.7527
	(0.018)	(0.018)	(0.018)	(0.017)	(0.018)	(0.016)	(0.017)	**(0.017)**	(0.010)	(0.012)
yeast	**0.7062**	0.6583	0.6760	0.6575	0.6616	0.6982	0.6565	0.7040	0.5704	0.5427
	(0.024)	(0.025)	(0.030)	(0.025)	(0.030)	(0.025)	(0.025)	(0.027)	(0.026)	(0.017)
yelp8	**0.8820**	0.8649	0.8613	0.8673	0.8626	0.8808	0.8622	**0.8820**	0.7537	0.7429
	(0.006)	(0.004)	(0.004)	(0.005)	(0.004)	(0.004)	(0.004)	**(0.005)**	(0.006)	(0.015)

Table 14 Results of best strategies for *macro-recall* ↑ measure

Data set	EBR_{SVM}	ECC_{XGB}	DBR_{XGB}	$RDBR_{XGB}$	MBR_{XGB}	BR_{XGB}	$BR+_{XGB}$	$PruDent_{XGB}$	CC_{XGB}	NS_{XGB}
20NG	**0.7740**	0.7634	0.7505	0.7459	0.7619	0.7615	0.7468	0.7615	0.6734	0.7364
	(0.003)	(0.016)	(0.003)	(0.002)	(0.003)	(0.003)	(0.003)	(0.003)	(0.007)	(0.004)
birds	0.1323	**0.4369**	0.3761	0.3761	0.3770	0.3770	0.3761	0.3770	0.3771	0.3605
	(0.046)	**(0.038)**	(0.027)	(0.027)	(0.026)	(0.026)	(0.027)	(0.026)	(0.026)	(0.028)
cal500	0.2186	**0.2318**	0.1089	0.1113	0.1200	0.1201	0.1085	0.1109	0.1190	0.1548
	(0.015)	**(0.009)**	(0.005)	(0.006)	(0.005)	(0.005)	(0.005)	(0.004)	(0.005)	(0.010)
corel5k	**0.0826**	0.0672	0.0394	0.0378	0.0450	0.0413	0.0400	0.0407	0.0469	0.0471
	(0.003)	(0.004)	(0.003)	(0.003)	(0.002)	(0.002)	(0.003)	(0.002)	(0.004)	(0.003)
emotions	**0.6680**	0.6655	0.6576	0.6560	0.6064	0.6064	0.6542	0.6309	0.6091	0.6089
	(0.024)	(0.018)	(0.025)	(0.025)	(0.020)	(0.020)	(0.023)	(0.021)	(0.014)	(0.013)
enron	0.2309	**0.2583**	0.2370	0.2289	0.2146	0.2145	0.2288	0.2167	0.2239	0.2220
	(0.010)	**(0.006)**	(0.005)	(0.004)	(0.007)	(0.007)	(0.006)	(0.007)	(0.007)	(0.007)
fapesp	0.4732	**0.5155**	0.4227	0.4245	0.4245	0.4245	0.4226	0.4254	0.4239	0.4163
	(0.024)	**(0.025)**	(0.024)	(0.023)	(0.027)	(0.027)	(0.025)	(0.026)	(0.026)	(0.024)
flags	0.6227	0.6440	0.6342	0.6429	0.6347	0.6347	0.6340	0.6338	**0.6505**	0.6418
	(0.028)	(0.033)	(0.023)	(0.028)	(0.031)	(0.031)	(0.024)	(0.025)	**(0.025)**	(0.027)
image	0.6777	**0.6808**	0.6476	0.6409	0.6473	0.6473	0.6426	0.6557	0.6232	0.6232
	(0.009)	**(0.011)**	(0.008)	(0.009)	(0.008)	(0.008)	(0.009)	(0.011)	(0.010)	(0.010)
langlog	0.1604	**0.1941**	0.1469	0.1476	0.1495	0.1495	0.1475	0.1495	0.1486	0.1485
	(0.006)	**(0.015)**	(0.012)	(0.011)	(0.012)	(0.012)	(0.012)	(0.012)	(0.012)	(0.012)
mediamill	**0.3256**	0.2315	0.1683	0.1648	0.1653	0.1677	0.1660	0.1666	0.1649	0.1621
	(0.004)	(0.004)	(0.003)	(0.003)	(0.002)	(0.003)	(0.002)	(0.003)	(0.003)	(0.003)
medical	0.6959	0.7236	0.7323	0.7301	0.7372	**0.7373**	0.7306	**0.7373**	0.7351	0.7011
	(0.012)	(0.021)	(0.027)	(0.029)	(0.026)	**(0.026)**	(0.026)	**(0.026)**	(0.026)	(0.026)
msd-195	**0.2102**	0.1734	0.0937	0.1000	0.0924	0.0924	0.0977	0.0941	0.0948	0.0952
	(0.004)	(0.007)	(0.007)	(0.008)	(0.006)	(0.006)	(0.007)	(0.005)	(0.006)	(0.006)

Table 14 (continued)

Data set	EBR_{SVM}	ECC_{XGB}	DBR_{XGB}	$RDBR_{XGB}$	MBR_{XGB}	BR_{XGB}	$BR+_{XGB}$	$PruDent_{XGB}$	CC_{XGB}	NS_{XGB}
ohsumed	**0.4245**	0.4125	0.3552	0.3536	0.3490	0.3491	0.3512	0.3491	0.3502	0.3453
	(0.005)	(0.004)	(0.002)	(0.002)	(0.003)	(0.003)	(0.002)	(0.003)	(0.002)	(0.003)
scene	**0.7978**	0.7883	0.7766	0.7659	0.7895	0.7895	0.7619	0.7895	0.7633	0.7645
	(0.013)	(0.010)	(0.004)	(0.005)	(0.007)	(0.007)	(0.002)	(0.007)	(0.003)	(0.001)
slashdot	**0.4443**	0.4194	0.3862	0.3820	0.3967	0.3975	0.3792	0.3975	0.3555	0.3793
	(0.012)	(0.019)	(0.008)	(0.007)	(0.008)	(0.008)	(0.008)	(0.008)	(0.010)	(0.008)
stackex	0.2162	**0.2873**	0.2239	0.2267	0.2268	0.2274	0.2259	0.2274	0.2255	0.1967
	(0.024)	**(0.014)**	(0.011)	(0.011)	(0.013)	(0.013)	(0.012)	(0.013)	(0.011)	(0.009)
tmc2007	0.5865	**0.6048**	0.5690	0.5688	0.5737	0.5632	0.5680	0.5632	0.5622	0.5633
	(0.007)	**(0.012)**	(0.011)	(0.012)	(0.014)	(0.013)	(0.011)	(0.013)	(0.011)	(0.013)
yeast	**0.4504**	0.4258	0.3835	0.3732	0.3666	0.3666	0.3984	0.3597	0.3887	0.3902
	(0.014)	(0.009)	(0.006)	(0.010)	(0.006)	(0.006)	(0.009)	(0.005)	(0.004)	(0.004)
yelp8	0.5719	**0.7002**	0.6768	0.6583	0.6342	0.6204	0.6536	0.6234	0.6487	0.6530
	(0.015)	**(0.005)**	(0.003)	(0.004)	(0.003)	(0.003)	(0.005)	(0.002)	(0.003)	(0.004)

Table 15 Results of best strategies for *one-error* ↓ measure

Data set	BR_{RF}	$PruDent_{RF}$	DBR_{RF}	MBR_{RF}	NS_{RF}	CC_{RF}	$RDBR_{RF}$	$BR+_{RF}$	EBR_{XGB}	ECC_{XGB}
20NG	0.2161	**0.2160**	0.2165	**0.2160**	0.2211	0.2221	0.2333	0.2297	0.2354	0.2676
	(0.003)	**(0.002)**	(0.002)	**(0.002)**	(0.002)	(0.003)	(0.007)	(0.006)	(0.003)	(0.008)
birds	0.3270	0.3284	**0.3218**	0.3272	0.3284	0.3284	0.3230	0.3224	0.3329	0.3319
	(0.034)	(0.026)	**(0.031)**	(0.031)	(0.031)	(0.031)	(0.029)	(0.030)	(0.038)	(0.040)
cal500	**0.1382**	0.1562	0.1418	0.1529	0.1406	0.1410	0.1422	0.1414	0.3015	0.2601
	(0.018)	(0.017)	(0.018)	(0.021)	(0.015)	(0.016)	(0.017)	(0.017)	(0.023)	(0.014)
corel5k	0.6393	0.6379	0.6560	0.6384	0.6375	**0.6371**	0.6547	0.6553	0.6756	0.6769
	(0.008)	(0.008)	(0.007)	(0.008)	(0.007)	**(0.005)**	(0.006)	(0.006)	(0.012)	(0.007)
emotions	**0.2626**	0.2649	0.2667	0.2659	0.2805	0.2805	0.2650	0.2667	0.2890	0.2873
	(0.015)	(0.010)	(0.016)	(0.012)	(0.014)	(0.014)	(0.017)	(0.017)	(0.016)	(0.012)
enron	0.2128	**0.2117**	0.2121	0.2175	0.2135	0.2141	0.2199	0.2136	0.2299	0.2362
	(0.011)	**(0.012)**	(0.011)	(0.010)	(0.008)	(0.009)	(0.011)	(0.010)	(0.014)	(0.012)
fapesp	0.3854	0.3894	0.3942	0.3854	0.3942	0.3942	0.3910	0.3934	0.3904	**0.3808**
	(0.048)	(0.041)	(0.052)	(0.039)	(0.042)	(0.042)	(0.052)	(0.052)	(0.023)	**(0.034)**
flags	0.2009	**0.1979**	0.2009	0.2039	0.2042	0.2042	0.2071	0.2103	0.2421	0.2154
	(0.025)	**(0.018)**	(0.027)	(0.029)	(0.019)	(0.019)	(0.026)	(0.018)	(0.035)	(0.018)
image	**0.2457**	0.2460	0.2471	0.2478	0.2469	0.2469	0.2480	0.2473	0.2525	0.2581
	(0.007)	(0.009)	(0.008)	(0.007)	(0.007)	(0.007)	(0.006)	(0.007)	(0.012)	(0.010)
langlog	0.6714	0.6715	0.6779	0.6694	0.6752	0.6724	0.6775	0.6774	0.6555	**0.6546**
	(0.010)	(0.015)	(0.013)	(0.013)	(0.011)	(0.019)	(0.013)	(0.013)	(0.011)	**(0.013)**
mediamill	**0.1015**	0.1057	0.1175	0.1032	0.1058	0.1059	0.1093	0.1110	0.1416	0.1605
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.002)	(0.003)	(0.001)	(0.001)	(0.003)
medical	0.1856	0.1791	0.1909	0.1831	0.1824	0.1786	0.1877	0.1915	**0.1337**	0.1362
	(0.015)	(0.011)	(0.015)	(0.012)	(0.015)	(0.012)	(0.009)	(0.013)	**(0.011)**	(0.012)
msd-195	0.6241	0.6258	0.6466	0.6258	0.6220	0.6220	0.6331	0.6361	**0.6043**	0.6127
	(0.011)	(0.011)	(0.009)	(0.009)	(0.009)	(0.009)	(0.014)	(0.013)	**(0.016)**	(0.014)

Table 15 (continued)

Data set	BR_{RF}	$PruDent_{RF}$	DBR_{RF}	MBR_{RF}	NS_{RF}	CC_{RF}	$RDBR_{RF}$	$BR+_{RF}$	EBR_{XGB}	ECC_{XGB}
ohsumed	0.3402	0.3397	0.3412	0.3414	0.3429	0.3422	0.3415	0.3432	**0.3233**	0.3234
	(0.002)	(0.003)	(0.003)	(0.003)	(0.003)	(0.002)	(0.002)	(0.002)	**(0.004)**	(0.002)
scene	0.1903	0.1897	0.1911	0.1902	0.1924	0.1919	0.1928	0.1915	**0.1861**	0.1982
	(0.006)	(0.006)	(0.006)	(0.007)	(0.000)	(0.000)	(0.002)	(0.000)	**(0.000)**	(0.009)
slashdot	0.3886	0.3887	**0.3873**	0.3885	0.3892	0.3917	0.4026	0.4045	0.4120	0.4277
	(0.008)	(0.007)	**(0.005)**	(0.005)	(0.008)	(0.006)	(0.008)	(0.008)	(0.008)	(0.006)
stackex	0.4684	0.4657	0.4718	0.4668	0.4676	0.4718	0.4707	0.4737	**0.4430**	0.4445
	(0.012)	(0.010)	(0.008)	(0.012)	(0.009)	(0.011)	(0.009)	(0.009)	**(0.016)**	(0.011)
tmc2007	**0.1482**	0.1489	0.1521	0.1695	0.1503	0.1512	0.1527	0.1530	0.1802	0.1763
	(0.001)	(0.001)	(0.001)	(0.003)	(0.001)	(0.002)	(0.002)	(0.001)	(0.003)	(0.002)
yeast	0.2304	0.2489	0.2459	0.2467	0.2525	0.2525	0.2434	0.2455	**0.2249**	0.2307
	(0.005)	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)	(0.004)	(0.004)	**(0.007)**	(0.006)
yelp8	0.1659	0.1695	0.1707	0.1703	0.1622	0.1622	0.1699	0.1698	0.1616	**0.1588**
	(0.005)	(0.004)	(0.004)	(0.003)	(0.004)	(0.005)	(0.004)	(0.004)	(0.004)	**(0.005)**

Table 16 Results of best strategies for *ranking-loss* ↓ measure

Data set	DBR_{RF}	$PruDent_{RF}$	BR_{RF}	$BR+_{RF}$	CC_{RF}	NS_{RF}	MBR_{RF}	$RDBR_{RF}$	EBR_{XGB}	ECC_{XGB}
20NG	0.0378	0.0377	**0.0376**	0.0408	0.0379	0.0379	0.0389	0.0407	0.0745	0.0882
	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.001)	(0.000)	(0.001)	(0.001)	(0.005)
birds	0.1205	0.1212	0.1226	**0.1204**	0.1234	0.1234	0.1217	**0.1204**	0.1909	0.1875
	(0.011)	(0.011)	(0.011)	**(0.011)**	(0.010)	(0.010)	(0.012)	**(0.010)**	(0.009)	(0.010)
cal500	0.2185	0.2287	**0.2180**	0.2187	0.2204	0.2202	0.2258	0.2210	0.2589	0.2548
	(0.003)	(0.003)	**(0.003)**	(0.003)	(0.003)	(0.003)	(0.003)	(0.002)	(0.002)	(0.003)
corel5k	0.1521	0.1534	0.1532	**0.1514**	0.1516	0.1519	0.1612	0.1535	0.2389	0.2377
	(0.001)	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.002)
emotions	**0.1455**	0.1460	0.1475	0.1465	0.1490	0.1490	0.1476	0.1468	0.1689	0.1664
	(0.006)	(0.006)	(0.008)	(0.006)	(0.007)	(0.007)	(0.006)	(0.007)	(0.012)	(0.009)
enron	**0.0832**	**0.0832**	0.0835	0.0834	0.0834	0.0833	**0.0832**	0.0834	0.1430	0.1368
	(0.001)	**(0.001)**	(0.001)	(0.001)	(0.002)	(0.001)	**(0.001)**	(0.001)	(0.003)	(0.004)
fapesp	0.1002	0.1022	0.1028	**0.1000**	0.1026	0.1026	0.1020	0.1001	0.1462	0.1350
	(0.009)	(0.010)	(0.008)	**(0.009)**	(0.009)	(0.009)	(0.009)	(0.009)	(0.016)	(0.013)
flags	0.1903	0.1876	0.1831	0.1899	0.1826	**0.1825**	0.1890	0.1900	0.2056	0.1980
	(0.014)	(0.013)	(0.013)	(0.014)	(0.013)	**(0.014)**	(0.018)	(0.014)	(0.015)	(0.012)
image	**0.1313**	0.1323	0.1321	**0.1313**	0.1323	0.1323	0.1328	0.1315	0.1532	0.1565
	(0.004)	(0.007)	(0.006)	**(0.004)**	(0.005)	(0.005)	(0.005)	(0.004)	(0.007)	(0.007)
langlog	0.1688	0.1681	0.1684	0.1687	0.1696	0.1695	**0.1664**	0.1685	0.2500	0.2530
	(0.003)	(0.003)	(0.003)	(0.003)	(0.004)	(0.005)	**(0.005)**	(0.003)	(0.008)	(0.007)
mediamill	0.0417	0.0377	**0.0338**	0.0418	0.0400	0.0400	0.0545	0.0435	0.1634	0.1584
	(0.000)	(0.000)	**(0.000)**	(0.000)	(0.000)	(0.000)	(0.001)	(0.001)	(0.001)	(0.002)
medical	0.0224	0.0227	0.0228	0.0228	0.0219	0.0218	**0.0217**	0.0222	0.0406	0.0413
	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)	**(0.003)**	(0.003)	(0.007)	(0.007)
msd-195	0.1499	0.1494	0.1520	0.1490	0.1487	0.1487	0.1522	**0.1469**	0.2695	0.2628
	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	**(0.002)**	(0.006)	(0.005)

Table 16 (continued)

Data set	DBR_{RF}	$PruDent_{RF}$	BR_{RF}	$BR+_{RF}$	CC_{RF}	NS_{RF}	MBR_{RF}	$RDBR_{RF}$	EBR_{XGB}	ECC_{XGB}
ohsumed	0.0849	0.0849	0.0851	0.0853	0.0853	0.0853	**0.0845**	0.0853	0.1741	0.1720
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	**(0.001)**	(0.001)	(0.002)	(0.002)
scene	0.0593	0.0593	**0.0589**	0.0593	0.0592	0.0598	0.0593	0.0605	0.0910	0.0862
	(0.003)	(0.002)	**(0.003)**	(0.001)	(0.002)	(0.002)	(0.003)	(0.001)	(0.003)	(0.005)
slashdot	**0.1091**	0.1110	0.1116	0.1134	0.1102	0.1106	0.1127	0.1123	0.1795	0.1805
	(0.002)	(0.002)	(0.001)	(0.001)	(0.002)	(0.002)	(0.002)	(0.001)	(0.005)	(0.004)
stackex	**0.1086**	0.1099	0.1107	0.1092	0.1110	0.1106	0.1130	0.1105	0.2366	0.2351
	(0.002)	(0.003)	(0.002)	(0.002)	(0.003)	(0.002)	(0.004)	(0.002)	(0.012)	(0.010)
tmc2007	0.0315	0.0312	**0.0311**	0.0315	0.0315	0.0314	0.0324	0.0315	0.0764	0.0723
	(0.000)	(0.000)	**(0.001)**	(0.001)	(0.001)	(0.000)	(0.001)	(0.000)	(0.001)	(0.001)
yeast	0.1715	0.1683	**0.1603**	0.1685	0.1682	0.1682	0.1701	0.1689	0.1678	0.1668
	(0.002)	(0.002)	**(0.002)**	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)
yelp8	0.1178	0.1227	0.1130	0.1178	0.1177	0.1172	0.1122	0.1217	0.1187	**0.1088**
	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.003)	(0.002)	**(0.002)**

Table 17 Results of best strategies for *subset-accuracy* ↑ measure

Data set	$RDBR_{RF}$	$BR+_{SVMt}$	NS_{SVMt}	DBR_{RF}	CC_{SVMt}	$PruDent_{RF}$	MBR_{RF}	BR_{RF}	ECC_{XGB}	EBR_{XGB}
20NG	0.7476	0.6963	0.7164	**0.7630**	0.6945	0.7627	0.7628	0.7628	0.6543	0.7180
	(0.005)	(0.008)	(0.006)	**(0.003)**	(0.009)	(0.002)	(0.002)	(0.003)	(0.017)	(0.003)
birds	**0.2852**	0.1758	0.1300	**0.2852**	0.1766	0.2804	0.2840	0.2830	0.2195	0.2196
	(0.026)	(0.067)	(0.052)	**(0.026)**	(0.064)	(0.022)	(0.026)	(0.022)	(0.028)	(0.043)
cal500	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
	(0.000)	**(0.000)**	**(0.000)**	**(0.000)**	**(0.000)**	**(0.000)**	**(0.000)**	**(0.000)**	**(0.000)**	**(0.000)**
corel5k	0.0045	0.0231	**0.0260**	0.0040	0.0249	0.0038	0.0044	0.0041	0.0089	0.0082
	(0.001)	(0.002)	**(0.004)**	(0.001)	(0.002)	(0.001)	(0.001)	(0.001)	(0.002)	(0.001)
emotions	0.3253	**0.3346**	0.3045	0.3220	0.3139	0.3160	0.3079	0.3116	0.3083	0.2871
	(0.041)	**(0.023)**	(0.024)	(0.038)	(0.029)	(0.035)	(0.037)	(0.037)	(0.020)	(0.018)
enron	0.1523	0.1519	0.1558	0.1495	0.1456	0.1407	0.1404	0.1388	**0.1563**	0.1465
	(0.016)	(0.013)	(0.012)	(0.015)	(0.012)	(0.016)	(0.015)	(0.014)	**(0.012)**	(0.016)
fapesp	0.3993	0.4176	**0.4351**	0.3960	0.4310	0.3969	0.4049	0.3977	0.3403	0.3481
	(0.025)	(0.035)	**(0.030)**	(0.024)	(0.021)	(0.016)	(0.016)	(0.016)	(0.026)	(0.028)
flags	**0.2506**	0.2207	0.1577	0.2381	0.1608	0.2144	0.2196	0.2113	0.2146	0.1835
	(0.022)	(0.023)	(0.049)	(0.027)	(0.081)	(0.029)	(0.026)	(0.031)	(0.037)	(0.035)
image	0.5566	0.5485	0.5445	**0.5572**	0.5458	0.5564	0.5543	0.5552	0.5148	0.5084
	(0.006)	(0.016)	(0.016)	**(0.009)**	(0.015)	(0.008)	(0.005)	(0.008)	(0.012)	(0.011)
langlog	0.2187	0.2130	0.2115	0.2179	0.2137	0.2199	**0.2240**	0.2222	0.1851	0.1863
	(0.012)	(0.010)	(0.011)	(0.012)	(0.013)	(0.016)	**(0.013)**	(0.010)	(0.005)	(0.011)
mediamill	0.1702	**0.2261**	0.2220	0.1622	0.2212	0.1513	0.1458	0.1460	0.1483	0.0991
	(0.002)	**(0.003)**	(0.003)	(0.002)	(0.003)	(0.001)	(0.002)	(0.002)	(0.002)	(0.002)
medical	0.6669	0.7449	0.7549	0.6644	0.7445	0.6749	0.6672	0.6669	**0.7561**	0.7540
	(0.012)	(0.026)	(0.017)	(0.015)	(0.021)	(0.015)	(0.015)	(0.013)	**(0.013)**	(0.011)
msd-195	0.1514	**0.1588**	0.1165	0.1325	0.1542	0.1373	0.1264	0.1286	0.0693	0.0619
	(0.008)	**(0.009)**	(0.051)	(0.008)	(0.008)	(0.009)	(0.009)	(0.009)	(0.006)	(0.005)

Table 17 (continued)

Data set	$RDBR_{RF}$	$BR+_{SVMt}$	NS_{SVMt}	DBR_{RF}	CC_{SVMt}	$PruDent_{RF}$	MBR_{RF}	BR_{RF}	ECC_{XGB}	EBR_{XGB}
ohsumed	0.2997	0.3086	0.3074	0.2994	**0.3089**	0.2988	0.2990	0.2980	0.2775	0.2753
	(0.004)	(0.003)	(0.004)	(0.004)	**(0.003)**	(0.004)	(0.005)	(0.004)	(0.004)	(0.005)
scene	0.7453	0.7319	0.7387	**0.7466**	0.7333	0.7454	0.7449	0.7428	0.7316	0.7266
	(0.001)	(0.006)	(0.008)	**(0.004)**	(0.006)	(0.005)	(0.005)	(0.004)	(0.012)	(0.003)
slashdot	0.4813	0.3141	0.3468	**0.4943**	0.3068	0.4901	0.4903	0.4892	0.4059	0.4345
	(0.009)	(0.191)	(0.169)	**(0.007)**	(0.168)	(0.008)	(0.006)	(0.008)	(0.014)	(0.006)
stackex	0.1324	0.1204	**0.1356**	0.1296	0.1161	0.1349	0.1336	0.1341	0.1209	0.1175
	(0.007)	(0.019)	**(0.012)**	(0.007)	(0.019)	(0.007)	(0.007)	(0.007)	(0.009)	(0.010)
tmc2007	**0.4486**	0.3722	0.4036	0.4456	0.3992	0.4373	0.4394	0.4373	0.3782	0.3649
	(0.003)	(0.038)	(0.031)	(0.004)	(0.024)	(0.004)	(0.004)	(0.003)	(0.002)	(0.004)
yeast	0.2092	0.2388	**0.2539**	0.1765	0.2264	0.1822	0.1623	0.1638	0.2295	0.1870
	(0.008)	(0.008)	**(0.009)**	(0.005)	(0.010)	(0.010)	(0.007)	(0.008)	(0.007)	(0.008)
yelp8	0.4065	0.4231	**0.4260**	0.4049	0.4235	0.3756	0.3746	0.3742	0.4174	0.3752
	(0.004)	(0.004)	**(0.006)**	(0.005)	(0.007)	(0.005)	(0.003)	(0.004)	(0.007)	(0.011)

Appendix 2: Statistical results

From the previous results, the best pairs of strategies/base-algorithms were statistically compared against the other pairs using the Bayesian statistical test. Tables 18, 19, 20, 21, 22, 23, 24 and 25 report the pairs that the considered strategies/base-algorithms statistically outperform with a probability greater than or equal to 95%.

Table 18 Bayesian statistical results for the *F1* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
EBR_{XGB}	*	*	*	*	124	12	*	*	*	*
ECC_{XGB}	*	*	*	*	124	12	*	*	*	*
$PruDent_{RF}$	1	1	1	1	1	1	1	1		
MBR_{LR}	1	1	1	1	1	1	1	1		
$RDBR_{SVMt}$	13	13	136	13	1	1	13	16	13	13
DBR_{SVMt}	1	1	136	1	1	1	1	13		
BR_{RF}	1	1	1	1	1	1	1	1		
NS_{RF}	1	1	1	1	1	1	1	1		
$BR+_{SVM}$	1	1	1	1	1	1	1	1		
CC_{RF}	1	1	1	1	1	1	1	1		

The cells' content indicates the base algorithms from the columns

Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: *All*

Table 19 Bayesian Statistical results for the *hamming-loss* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
$PruDent_{RF}$	16	16	16	156	*	*	16	16	16	16
BR_{RF}	16	16	16	16	*	*	16	16	16	16
MBR_{RF}	16	16	16	156	*	*	16	16	16	16
$BR+_{RF}$	1	16	16	16	*	*	1	16	1	16
DBR_{RF}	1	16	16	16	*	*	1	16	1	16
$RDBR_{RF}$	1	16	16	16	*	*	1	16	1	16
CC_{RF}	16	16	16	156	*	*	16	16	16	16
NS_{RF}	1	16	16	16	*	*	1	16	1	1
EBR_{RF}	1	1	1	1	124	12	1	1	1	1
ECC_{RF}	1	1	1	1	12	12	1	1	1	1

The cells' content indicates the base algorithms from the columns

Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: *All*

Table 20 Bayesian Statistical results for the *macro-F1* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
EBR_{XGB}	*	*	*	*	123	123	*	*	*	*
ECC_{XGB}	*	12346	12346	12346	123	123	*	*	*	12346
MBR_{XGB}	1234	13	123	13	13	13	13	1234	1234	13
BR_{XGB}	1234	13	1234	13	13	1	13	1234	1234	13
$PruDent_{XGB}$	1234	13	1234	123	13	1	13	1234	1234	13
$RDBR_{SVMt}$	1234	13	13	13	12	1	1234	1234	1234	13
DBR_{SVMt}	1234	13	13	13	13	1234	134	13		
CC_{XGB}	123	13	13	13	1	1	13	123	13	13
NS_{XGB}	1234	13	13	13	1	1	13	123	134	13
$BR+_{SVMt}$	1234	13	13	13	13	1234	1234	13		

The cells' content indicates the base algorithms from the columns

Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: *All*

Table 21 Bayesian Statistical results for the *macro-precision* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
MBR_{RF}	1	1	1	12	*	12345	1	12	14	1
CC_{RF}	1	1	1	1	12345	12345	1	1	1	1
$RDBR_{RF}$	12	124	12	124	*	12345	1	124	14	1
NS_{RF}	1	1	1	1	*	*	1	12	1	1
$BR+_{RF}$	12	1	12	12	*	12345	1	124	14	1
$PruDent_{RF}$	12	1	1	12	*	12345	1	12	14	1
DBR_{RF}	12	1	1	12	*	12345	1	12	14	1
BR_{RF}	1	1	1	1	*	12345	1	1	1	1
ECC_{XGB}	1	1	1	1	1245	124	1	1	1	1
EBR_{XGB}	1	1	1	1	124	124	1	1	1	1

The cells' content indicates the base algorithms from the columns

Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: *All*

Table 22 Bayesian Statistical results for the *macro-recall* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
EBR_{SVM}	*	*	*	*	123	123	*	*	*	*
ECC_{XGB}	*	*	*	*	123	123	*	*	*	*
DBR_{XGB}	1234	13	13	13	13	123	134	13		
$RDBR_{XGB}$	1234	13	13	13	13	123	134	13		
MBR_{XGB}	1234	13	13	13	13	1234	134	13		
BR_{XGB}	1234	13	13	13	13	1234	134	13		
$BR+_{XGB}$	1234	13	13	13	13	1234	134	13		
$PruDent_{XGB}$	1234	13	134	13	13	1234	1234	13		
CC_{XGB}	13	13	13	13	13	13	13	13		
NS_{XGB}	13	13	13	13	13	13	13	13		

The cells' content indicates the base algorithms from the columns

Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: *All*

Table 23 Bayesian Statistical results for the *one-error* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
BR_{RF}	1	1246	12456	1246	1245	1246	12456	1246	124	12456
$PruDent_{RF}$	1	1246	1246	1246	1245	124	12456	16	124	1246
DBR_{RF}	1	1246	1246	1246	124	124	146	16	12	124
MBR_{RF}	1	1246	1246	1246	124	124	1456	16	12	1246
NS_{RF}	1	1246	1246	1246	124	124	1456	16	12	1246
CC_{RF}	1	1246	1246	1246	124	124	1456	16	12	1246
$RDBR_{RF}$	1	1	146	1	12	12	16	16	12	1
$BR+_{RF}$	1	1	16	1	12	12	16	16	12	1
EBR_{XGB}	1	1	16	1	124	12	16	1	12	1
ECC_{XGB}	1	1	16	1	12	12	1	1	1	1

The cells' content indicates the base algorithms from the columns

Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: *All*

Table 24 Bayesian Statistical results for the *ranking-loss* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
DBR_{RF}	1	126	16	16	*	*	156	16	126	126
$PruDent_{RF}$	1	1246	16	126	*	*	156	16	126	1246
BR_{RF}	1	1246	16	1246	*	*	156	16	126	1246
$BR+_{RF}$	1	16	16	16	*	*	156	16	126	126
CC_{RF}	1	126	16	16	*	*	156	16	126	126
NS_{RF}	1	126	16	16	*	*	156	16	126	126
MBR_{RF}	1	16	16	16	*	*	156	16	12	16
$RDBR_{RF}$	1	126	16	16	*	*	156	16	126	126
EBR_{XGB}	1	1	1	1	34	3	1	1	1	1
ECC_{XGB}	1	1	1	1	34	34	1	1	1	1

The cells' content indicates the base algorithms from the columns

Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: *All*

Table 25 Bayesian Statistical results for the *subset-accuracy* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
$RDBR_{RF}$	16	1	1	1	*	*	16	1	1	1
$BR+_{SVMt}$	16	1	1	1	*	*	1	1	1	1
NS_{SVMt}	1	1	1	1	*	*	1	1	1	1
DBR_{RF}	16	1	1	1	*	*	1	1	1	1
CC_{SVMt}	16	1	1	1	*	*	1	1	1	1
$PruDent_{RF}$	1	1	1	1	*	12345	1	1	1	1
MBR_{RF}	1	1	1	1	*	12345	1	1	1	1
BR_{RF}	1	1	1	1	*	12345	1	1	1	1
ECC_{XGB}	1	1	1	1	1245	1245	1	1	1	1
EBR_{XGB}	1	1	1	1	12	12	1	1	1	1

The cells' content indicates the base algorithms from the columns

Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: *All*

References

Alali, A., & Kubat, M. (2015). PruDent: A pruned and confident stacking approach for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, *27*(9), 2480–2493. https://doi.org/10.1109/TKDE.2015.2416731.

Benavoli, A., Corani, G., Demsar, J., & Zaffalon, M. (2017). Time for a change: A tutorial for comparing multiple classifiers through bayesian analysis. *Journal of Machine Learning Research*, *18*, 77:1–77:36.

Bernardini, F. C., Benito, E., & Meza, M. (2014). Cardinality and density measures and their influence to multi-label learning methods. *Journal of the Brazilian Society on Computational Intelligence*, *12*(1), 53–71.

Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, *37*(9), 1757–1771. https://doi.org/10.1016/j.patcog.2004.03.009.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324.

Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., Hadley, S. F., 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* (pp. 1–8). https://doi.org/10.1109/MLSP.2013.6661934.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27. https://doi.org/10.1145/1961189.1961199.

Charte, F., Rivera, A. J., del Jesus, M. J., & Herrera, F. (2015). QUINTA: A question tagging assistant to improve the answering ratio in electronic forums. In *IEEE international conference on computer as a tool, IEEE* (pp. 1–6). https://doi.org/10.1109/EUROCON.2015.7313677.

Charte, F., & Charte, F. D. (2015). Working with multilabel datasets in R: The mldr Package. *The R Journal*, *7*(2), 149–162.

Charte, F., Rivera, A. J., Charte, D., del Jesús, M. J., & Herrera, F. (2018). Tips, guidelines and tools for managing multi-label datasets: The mldr.datasets R package and the cometa data repository. *Neurocomputing*, *289*, 68–85. https://doi.org/10.1016/j.neucom.2018.02.011.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM international conference on knowledge discovery and data mining* (pp. 785–794). https://doi.org/10.1145/2939672.2939785.

Cherman, E. A., Metz, J., & Monard, M. C. (2012). Incorporating label dependency into the binary relevance framework for multi-label classification. *Expert Systems with Applications*, *39*(2), 1647–1655. https://doi.org/10.1016/j.eswa.2011.06.056.

Cherman, E. A., Spolaôr, N., Valverde-Rebaza, J., & Monard, M. C. (2014). Lazy multi-label learning algorithms based on mutuality strategies. *Journal of Intelligent & Robotic Systems*,. https://doi.org/10.1007/s10846-014-0144-4.

de Carvalho, A. C. P. L. F., & Freitas, A. A. (2009). A tutorial on multi-label classification techniques. In A. Abraham, A. E. Hassanien, & V. Snášel (Eds.), *Foundations of computational intelligence* (pp. 177–195). Berlin: Springer. https://doi.org/10.1007/978-3-642-01536-6_8.

de Sá, A. G. C., Freitas, A. A., & Pappa, G. L. (2018). Automated selection and configuration of multi-label classification algorithms with grammar-based genetic programming. In A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, D. Whitley (Eds.), *Parallel Problem Solving from Nature - PPSN XV–15th international conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part II, Springer, Lecture Notes in Computer Science* (Vol. 11102, pp. 308–320). https://doi.org/10.1007/978-3-319-99259-4_25.

de Sá, A. G. C., Pappa, G. L., & Freitas, A. A. (2017). Towards a method for automatically selecting and configuring multi-label classification algorithms. In *Proceedings of the genetic and evolutionary computation conference companion* (pp. 1125–1132) https://doi.org/10.1145/3067695.3082053.

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, 7th European conference on computer vision, Copenhagen, Denmark, May 28–31, 2002, Proceedings, Part IV, Lecture Notes in Computer Science* (Vol. 2353, pp. 97–112). Berlin: Springer. https://doi.org/10.1007/3-540-47979-1_7.

Elisseeff, A., & Weston, J. (2001). A kernel method for multi-labeled classification. In *Proceedings of the neural information processing systems* (pp. 681–687).

Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models. Analytical methods for social research*. New York: Cambridge University Press.

Gibaja, E., & Ventura, S. (2014). Multi-label learning: A review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(6), 411–444. https://doi.org/10.1002/widm.1139.

Gibaja, E., & Ventura, S. (2015). A tutorial on multilabel learning. *ACM Computing Surveys*, *47*(3), 1–38. https://doi.org/10.1145/2716262.

Godbole, S., & Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia conference*, (pp. 22–30) https://doi.org/10.1007/978-3-540-24775-3_5.

Gonçalves, E. C., Plastino, A., & Freitas, A. A. (2013). A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In *Proceedings of the international conference on tools with artificial intelligence* (pp. 469–476). https://doi.org/10.1109/ICTAI.2013.76.

Jackson, P., & Moulinier, I. (2002). *Natural language processing for online applications: Text retrieval, extraction & categorization*. Amsterdam: John Benjamins.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice-Hall Inc.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the 10th European Conference on Machine Learning*, *1398*, 137–142.

Klimt, B., & Yang, Y. (2004). The Enron Corpus: A new dataset for email classification research. In *Proceedings of the 15th European conference on Machine Learning* (pp. 217–226) https://doi.org/10.1007/978-3-540-30115-8_22.

Lang, K. (1995). Newsweeder: Learning to filter Netnews. In *Proceedings of the twelfth international conference on machine learning*, (pp. 331–339).

Li, Y. k., & Zhang, M. L. (2014). Enhancing binary relevance for multi-label learning with controlled label correlations exploitation. In *13th Pacific Rim International Conference on Artificial Intelligence* (pp. 91–103). https://doi.org/10.1007/978-3-319-13560-1_8.

Liu, S. M., & Chen, J. (2015). An empirical study of empty prediction of multi-label classification. *Expert Syst Appl*, *42*(13), 5567–5579. https://doi.org/10.1016/j.eswa.2015.01.024.

Luaces, O., Díez, J., Barranquero, J., del Coz, J. J., & Bahamonde, A. (2012). Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence*, *1*(4), 303–313.

Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, *45*(9), 3084–3104. https://doi.org/10.1016/j.patcog.2012.03.004.

Mantovani, R. G., Rossi, A. L. D., Vanschoren, J., Bischl, B., & Carvalho, A. C. P. L. F. (2015). To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning. In *2015 International Joint Conference on Neural Networks, IEEE*, (pp. 1–8). https://doi.org/10.1109/IJCNN.2015.7280644.

Metz, J., de Abreu, L. F., Cherman, E. A., & Monard, M. C. (2012). On the estimation of predictive evaluation measure baselines for multi-label learning. In *13th Ibero-American Conference on Artificial Intelligence* (pp. 189–198).

Montañes, E., Senge, R., Barranquero, J., Quevedo, J. R., Coz, J Jd, & Hüllermeier, E. (2014). Dependent binary relevance models for multi-label classification. *Pattern Recognition*, *47*(3), 1494–1508. https://doi.org/10.1016/j.patcog.2013.09.029.

Moyano, J. M., Galindo, E. L. G., Cios, K. J., & Ventura, S. (2018). Review of ensembles of multi-label classifiers: Models, experimental study and prospects. *Information Fusion*, *44*, 33–45. https://doi.org/10.1016/j.inffus.2017.12.001.

Pereira, R. B., Plastino, A., Zadrozny, B., & Merschmann, L. H. (2018). Correlation analysis of performance measures for multi-label classification. *Information Processing & Management*, *54*(3), 359–369. https://doi.org/10.1016/j.ipm.2018.01.002.

Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D. J., Johnson, N., Cohen, K. B., & Duch, W. (2007). A shared task involving multi-label classification of clinical free text. In *Proceedings of the workshop on biological, translational, and clinical language processing, association for computational linguistics* (pp. 97–104).

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Raez, A. M., Lopez, L. A. U., Steinberger, R. (2004). Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. In *Advances in Natural Language Processing* (pp. 1–12). https://doi.org/10.1007/978-3-540-30228-5_1.

Rauber, T. W., Mello, L. H., Rocha, V. F., Luchi, D., & Varejão, F. M. (2014). Recursive dependent binary relevance model for multi-label classification. In A. L. Bazzan, K. Pichara (Eds), *Advances in artificial intelligence—IBERAMIA 2014* (pp. 206–217). https://doi.org/10.1007/978-3-319-12027-0_17.

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classifier chains for multi-label classification. *Proceedings of the European conference, Bled, Slovenia*, *5782*, 254–269.

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, *85*(3), 333–359.

Rivolli, A., & de Carvalho, A. C. P. L. F. (2018). The utiml Package: Multi-label Classification in R. *The R Journal*https://journal.r-project.org/archive/2018/RJ-2018-041/index.html.

Rivolli, A., Soares, C., & de Carvalho, A. C. P. L. F. (2018). Enhancing multilabel classification for food truck recommendation. *Expert Systems*,. https://doi.org/10.1111/exsy.12304.

Schapire, E. R., & Singer, Y. (1999). Improved boosting algorithm using confidence-rated predictions. *Machine Learning*, *37*(3), 297–336. https://doi.org/10.1023/A:1007614523901.

Sechidis, K., Tsoumakas, G., & Vlahavas, I. (2011). On the stratification of multi-label data. In D. Gunopulos, T. Hofmann, D. Malerba, Vazirgiannis M. (Eds.), *Machine learning and knowledge discovery in databases* (pp. 145–158). https://doi.org/10.1007/978-3-642-23808-6_10.

Senge, R., del Coz, J. J., & Hüllermeier, E. (2013). Rectifying classifier chains for multi-label classification. In *Proceedings of the Workshop of Lernen, Wissen & Adaptivität, Bamberg, Germany* (pp. 162–169).

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) https://doi.org/10.1145/1180639.1180727.

Srivastava, A. N., & Zane-Ulman, B. (2005). Discovering recurring anomalies in text reports regarding complex space systems. In *IEEE aerospace conference* (pp. 3853–3862). https://doi.org/10.1109/AERO.2005.1559692.

Trohidis, K., Tsoumakas, G., Kalliris, G., & Vlahavas, I. (2011). Multi-label classification of music by emotion. *Journal on Audio, Speech, and Music Processing*, *2011*(1), 4. https://doi.org/10.1186/1687-4722-2011-426793.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases, workshop on mining multidimensional data* (pp. 30–44).

Tsoumakas, G., Loza Mencía, E., Katakis, I., Park, S. H., & Fürnkranz, J. (2009). On the combination of two decompositive multi-label classification methods. In *Proceedings of the European conference on machine learning and principles and practice of knowledge discovery, workshop on preference learning* (pp. 114–129).

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, *3*(3), 1–13.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Mining multi-label data. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook, Chap 34* (2nd ed., pp. 667–685). Berlin: Springer. https://doi.org/10.1007/978-0-387-09823-4_34.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2011a). Random k-labelsets for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, *23*(7), 1079–1089.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2011b). Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, *23*(7), 1079–1089. https://doi.org/10.1109/TKDE.2010.164.

Turnbull, D., Barrington, L., Torres, D., & Lanckriet, G. (2008). Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, *16*(2), 467–476. https://doi.org/10.1109/TASL.2007.913750.

Wever, M., Mohr, F., & Hüllermeier, E. (2018). Automated multi-label classification based on ML-plan. arXiv:1811.04060.

Wever, M. D., Mohr, F., Tornede, A., & Hüllermeier, E. (2019). Automating multi-label classification extending ml-plan. In *6th ICML Workshop on Automated Machine Learning*.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, *1*(1–2), 69–90. https://doi.org/10.1023/A:1009982220290.

Zhang, M. L., & Wu, L. (2015). Lift: Multi-Label learning with label-specific features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(1), 107–120. https://doi.org/10.1109/TPAMI.2014.2339815.

Zhang, M. L., & Zhou, Z. H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *26*(8), 1819–1837. https://doi.org/10.1109/TKDE.2013.39.

Zhou, Z., & Zhang, M. (2006). Multi-instance multi-label learning with application to scene classification. In B. Schölkopf, J. C. Platt, & T. Hofmann (Eds.), *Advances in neural information processing systems 19, Proceedings of the twentieth annual conference on neural information processing systems, Vancouver, British Columbia*, December 4–7, 2006, (pp. 1609–1616). Cambridge: MIT Press.

Zhou, T., Tao, D., & Wu, X. (2012). Compressed labeling on distilled labelsets for multi-label learning. *Machine Learning*, *88*(1–2), 69–126.

Zufferey, D., Hofer, T., Hennebert, J., Schumacher, M., Ingold, R., & Bromuri, S. (2015). Performance comparison of multi-label learning algorithms on clinical data for chronic diseases. *Computers in Biology and Medicine*, *65*, 34–43. https://doi.org/10.1016/j.compbiomed.2015.07.017.

Affiliations

Adriano Rivolli[1] · Jesse Read[2] · Carlos Soares[3] · Bernhard Pfahringer[4] · André C. P. L. F. de Carvalho[5]

Jesse Read
jesse.read@polytechnique.edu

Carlos Soares
csoares@fe.up.pt

Bernhard Pfahringer
bernhard@waikato.ac.nz

André C. P. L. F. de Carvalho
andre@icmc.usp.br

[1] Department of Computer Science, Technological University of Paraná, Cornélio Procópio, PR, Brazil

[2] Laboratoire d'Informatique (LIX), École Polytechnique, Palaiseau, France

[3] Fraunhofer AICOS and LIAAD-INESC TEC, University of Porto, Porto, Portugal

[4] University of Waikato, Hamilton, New Zealand

[5] Institute of Mathematics and Computer Sciences, University of São Paulo, São Carlos, SP, Brazil