LiBRe: Label-Wise Selection of Base Learners in Binary Relevance for Multi-label Classification

Marcel Wever[1](✉), Alexander Tornede[1], Felix Mohr[2], and Eyke Hüllermeier[1]

[1] Heinz Nixdorf Institut, Paderborn University, Paderborn, Germany
{marcel.wever,alexander.tornede,eyke}@upb.de
[2] Universidad de La Sabana, Chia, Cundinamarca, Colombia
felix.mohr@unisabana.edu.co

Abstract. In multi-label classification (MLC), each instance is associated with a set of class labels, in contrast to standard classification, where an instance is assigned a single label. Binary relevance (BR) learning, which reduces a multi-label to a set of binary classification problems, one per label, is arguably the most straight-forward approach to MLC. In spite of its simplicity, BR proved to be competitive to more sophisticated MLC methods, and still achieves state-of-the-art performance for many loss functions. Somewhat surprisingly, the optimal choice of the base learner for tackling the binary classification problems has received very little attention so far. Taking advantage of the label independence assumption inherent to BR, we propose a label-wise base learner selection method optimizing label-wise macro averaged performance measures. In an extensive experimental evaluation, we find that or approach, called LiBRe, can significantly improve generalization performance.

Keywords: Multi-label classification · Algorithm selection · Binary relevance

1 Introduction

By relaxing the assumption of mutual exclusiveness of classes, the setting of *multi-label classification* (MLC) generalizes standard (binary or multinomial) classification—subsequently also referred to as single-label classification (SLC). MLC has received a lot of attention in the recent machine learning literature [23, 29]. The motivation for allowing an instance to be associated with several classes simultaneously originated in the field of text categorization [19], but nowadays multi-label methods are used in applications as diverse as image processing [4,26] and video annotation [14], music classification [18], and bioinformatics [2].

Common approaches to MLC either adapt existing algorithms (*algorithm adaptation*) to the MLC setting, e.g., the structure and the training procedure for neural networks, or reduce the original MLC problem to one or multiple SLC problems (*problem transformation*). The most intuitive and straight-forward

© The Author(s) 2020
M. R. Berthold et al. (Eds.): IDA 2020, LNCS 12080, pp. 561–573, 2020.
https://doi.org/10.1007/978-3-030-44584-3_44

problem transformation is to decompose the original task into several binary classification tasks, one per label. More specifically, each task consists of training a classifier that predicts whether or not a specific label is relevant for a query instance. This approach is called *binary relevance* (BR) learning [3]. Beyond BR, many more sophisticated strategies have been developed, most of them trying to exploit correlations and interdependencies between labels [28]. In fact, BR is often criticized for ignoring such dependencies, implicitly assuming that the relevance of one label is (statistically) independent of the relevance of another label. In spite of this, or perhaps just because of this simplification, BR proved to achieve state-of-the-art performance, especially for so-called decomposable loss functions, for which its optimality can even be corroborated theoretically [7,9].

Techniques for reducing MLC to SLC problems involve the choice of a base learner for solving the latter. Somewhat surprisingly, this choice is often neglected, despite having an important influence on generalization performance [10–12,15]. Even in more extensive studies [10,12], a base learner is fixed a priori in a more or less arbitrary way. Broader studies considering multiple base learners, such as [6,22], are relatively rare and rather limited in terms of the number of base learners considered. Only recently, greater attention to the choice of the base learner has been paid in the field of automated machine learning (AutoML) [17,24,25], where the base learner is considered as an important "hyper-parameter" to tune. Indeed, while optimizing the selection of base learners is laborious and computationally expensive in general, which could be one reason for why it has been tackled with reservation, AutoML now offers new possibilities in this direction.

Motivated by these opportunities, and building on recent AutoML methodology, we investigate the idea of base learner selection for BR in a more systematic way. Instead of only choosing a single base learner to be used for all labels simultaneously, we even allow for selecting an individual learner for each label (i.e., each binary classification task) separately. In an extensive experimental study, we find that customizing BR in a label-wise manner can significantly improve generalization performance.

2 Multi-label Classification

The setting of *multi-label classification* (MLC) allows an instance to belong to several classes simultaneously. Consequently, several class labels can be assigned to an instance at the same time. For example, a single image could be tagged with labels `Sun` and `Beach` and `Sea` and `Yacht`.

2.1 Problem Setting

To formalize this learning problem, let $\mathcal{X}$ denote an instance space and $\mathcal{L} = \{\lambda_1, \ldots, \lambda_m\}$ a finite set of m class labels. An instance $\boldsymbol{x} \in \mathcal{X}$ is then (non-deterministically) associated with a subset of class labels $L \in 2^{\mathcal{L}}$. The subset L is often called the set of relevant labels, while its complement $\mathcal{L} \setminus L$ is considered irrelevant for $\boldsymbol{x}$. Furthermore, a set L of relevant labels can be identified by a binary vector $\boldsymbol{y} = (y_1, \ldots, y_m)$ where $y_i = 1$ if $\lambda_i \in L$ and $y_i = 0$ otherwise (i.e., if $\lambda_i \in \mathcal{L} \setminus L$). The set of all label combinations is denoted by $\mathcal{Y} = \{0, 1\}^m$.

Generally speaking, a multi-label classifier $\boldsymbol{h}$ is a mapping $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$ returning, for a given instance $\boldsymbol{x} \in \mathcal{X}$, a prediction in the form of a vector

$$\boldsymbol{h}(\boldsymbol{x}) = \big(h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})\big).$$

The MLC task can be stated as follows: Given a finite set of observations as training data $\mathcal{D}_{\text{train}} := (X_{\text{train}}, Y_{\text{train}}) = \big\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\big\}_{i=1}^{N} \subset \mathcal{X}^N \times \mathcal{Y}^N$, the goal is to learn a classifier $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$ that generalizes well beyond these observations in the sense of minimizing the risk with respect to a specific loss function.

2.2 Loss Functions

A wide spectrum of loss functions has been proposed for MLC, many of which are generalizations or adaptations of losses for single-label classification. In general, these loss functions can be divided into two major categories: instance-wise and label-wise. While the latter first compute a loss for each label and then aggregate the values obtained across the labels, e.g., by taking the mean, instance-wise loss functions first compute a loss for each instance and subsequently aggregate the losses over all instances in the test data. As an obvious advantage of label-wise loss functions, note that they can be optimized by optimizing a standard SLC loss for each label separately. In other words, label-wise losses naturally harmonize with label-wise decomposition techniques such as BR. Since this allows for a simpler selection of the base learner per label, we focus on two such loss functions in the following. For additional details on MLC and loss functions, especially instance-wise losses, we refer to [23,29].

Let $\mathcal{D}_{\text{test}} := (X_{\text{test}}, Y_{\text{test}}) = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{S} \subset \mathcal{X}^S \times \mathcal{Y}^S$ be a test set of size S. Further, let $H = (\boldsymbol{h}(\boldsymbol{x}_1), \ldots, \boldsymbol{h}(\boldsymbol{x}_S)) \subset \mathcal{Y}^S$. Then, the Hamming loss, which can be seen as a generalized form of the error rate, is defined[1] as

$$\mathcal{L}_H(Y_{\text{test}}, H) := \frac{1}{m} \sum_{j=1}^{m} \frac{1}{S} \sum_{i=1}^{S} [\![y_{i,j} \neq h_j(\boldsymbol{x}_i))]\!] \quad . \tag{1}$$

Moreover, the label-wise macro-averaged F-measure (which is actually a measure of accuracy, not a loss function, and thus to be maximized) is given by

$$\mathrm{F}(Y_{\text{test}}, H) := \frac{1}{m} \sum_{j=1}^{m} \frac{2 \sum_{i=1}^{S} y_{i,j} h_j(\boldsymbol{x}_i)}{\sum_{i=1}^{S} y_{i,j} + \sum_{i=1}^{S} h_j(\boldsymbol{x}_i)} \quad . \tag{2}$$

Obviously, to optimize the measures (1) and (2), it is sufficient to optimize each label individually, which corresponds to optimizing the inner term of the (first) sum.

[1] $[\![\cdot]\!]$ is the indicator function.

2.3 Binary Relevance

As already said, binary relevance learning decomposes the MLC task into several binary classification tasks, one for each label. For every such task, a single-label classifier, such as an SVM, random forest, or logistic regression, is trained. More specifically, a classifier for the j^{th} label is trained on the dataset $\{(\boldsymbol{x}_i, y_{i,j})\}_{i=1}^{N}$. Formally, BR induces a multi-label predictor

$$\mathbf{BR}_{\boldsymbol{b}} : \mathcal{X} \longrightarrow \mathcal{Y}, \quad \boldsymbol{x} \mapsto \big(b_1(\boldsymbol{x}), b_2(\boldsymbol{x}), \ldots, b_m(\boldsymbol{x})\big) \ ,$$

where $b_j : \mathcal{X} \longrightarrow \{0, 1\}$ represents the prediction of the base learner for the j^{th} label.

3 Related Work

Binary relevance has been subject to modifications in various directions, an excellent overview of which is provided in a recent survey [28]. Extensions of BR mainly focus on its inability to exploit label correlations, due to treating all labels independently of each other. Three types of approaches have been proposed to overcome this problem. The first is to use *classifier chains* [15]. In this approach, one first defines a total order among the m labels and then trains binary classifiers in this order. The input of the classifier for the i^{th} label is the original data plus the predictions of *all classifiers* for labels preceding this label in the chain. Similarly, in addition to the binary classifiers for the m labels, *stacking* uses a second layer of m meta-classifiers, one for each label, which take as input the original data augmented by the predictions of *all* base learners [11,21]. A third approach seeks to capture the dependencies in a Bayesian network, and to learn such a network from the data [1,20]. One can then use probabilistic inference to compute the probability for each possible prediction.

Another line of research looks at how the problem of imbalanced classes can be addressed using BR. Class imbalance constitutes an important challenge in multi-label classification in general, since most labels are usually irrelevant for an instance, i.e., the overwhelming majority of labels in a binary task is negative. Using BR, the imbalance can be "repaired" in a label-wise manner, using techniques for standard binary classification, such as sampling [5] or thresholding the decision boundary [13]. An approach taking dependencies among labels into account (and hence applied prior to splitting the problem) is presented in [27].

To the best of our knowledge, this is the first approach in which the base learner used for the different labels is subject to optimization itself. In fact, except for AutoML tools, we are not even aware of an approach optimizing a single base learner applied to all labels. In all the above approaches, the choice of the base learners is an external decision and not part of the learning problem itself.

4 Label-Wise Selection of Base Learners

As already stated before, while various attempts at improving binary relevance learning by capturing label dependencies have been made, the choice of the base learner for tackling the underlying binary problems—as another potential source of improvement—has attracted much less attention in the literature so far. If considered at all, this choice has been restricted to the selection of a *single* learner, which is applied to all m binary problems simultaneously.

We proceed from a portfolio of base learners

$$\mathcal{A} := \big\{ a \mid a : (\mathcal{X}^n \times \{0,1\}^n) \longrightarrow (\mathcal{X} \longrightarrow \{0,1\}) \big\} \, .$$

Then, given training data $\mathcal{D}_{\text{train}} = (X_{\text{train}}, Y_{\text{train}})$, the objective is to find the base learner a for which BR performs presumably best on test data $\mathcal{D}_{\text{test}} = (X_{\text{test}}, Y_{\text{test}})$ with respect to some loss function $\mathcal{L}$:

$$\arg\min_{a \in \mathcal{A}} \; \mathcal{L}\big(Y_{\text{test}}, \mathbf{BR}_{\boldsymbol{b}}(X_{\text{test}})\big), \text{ with } b_j := a\left(X_{\text{train}}, Y^{(j)}_{\text{train}}\right) \quad , \tag{3}$$

where $Y^{(i)}_{\text{train}}$ denotes the j^{th} column of the label matrix Y_{train}.

Moreover, we propose to leverage the independence assumption underlying BR to select a different base learner for each of the labels, and refer to this variant as LiBRe. We are thus interested in solving the following problem:

$$\arg\min_{\boldsymbol{a} \in \mathcal{A}^m} \mathcal{L}\big(Y_{\text{test}}, \mathbf{BR}_{\boldsymbol{b}}(X_{\text{test}})\big), \text{ with } b_j := a_j\left(X_{\text{train}}, Y^{(j)}_{\text{train}}\right) \quad . \tag{4}$$

Compared to (3), we thus significantly increase flexibility. In fact, by taking advantage of the different behavior of the respective base learners, and the ability to model the relationship between features and a class label differently for each binary problem, one may expect to improve the overall performance of BR. On the other side, the BR learner as a whole is now equipped with many degrees of freedom, namely the choice of the base learners, which can be seen as "hyper-parameters" of LiBRe. Since this may easily lead to undesirable effects such as over-fitting of the training data, an improvement in terms of generalization performance (approximated by the performance on the test data) is by no means self-evident. From this point of view, the restriction to a single base learner in (3) can also be seen as a sort of regularization. Such kind of regulation can indeed be justified for various reasons. In most cases, for example, the binary problems are indeed not completely different but share important characteristics.

Computationally, (4) may appear more expensive than choosing a single base learner jointly for all the labels, at least at first sight. However, the complexity in terms of the number of base learners to be evaluated remains exactly the same. In fact, just like in (3), we need to fit a BR model for every base learner exactly once. The only difference is that, instead of picking one of the base learners for all labels in the end, LiBRe assembles the base learners performing best for the respective labels (recall that we head for label-wise decomposable performance measures).

5 Experimental Evaluation

This section presents an empirical evaluation of LiBRe, comparing it to the use of a single base learner as a baseline. We first describe the experimental setup (Sect. 5.1), specify the baseline with the single best base learner (Sect. 5.2), and define the oracle performance (Sect. 5.3) for an upper bound. Finally, the experimental results are presented in Sect. 5.4.

5.1 Experimental Setup

For the evaluation, we considered a total of 24 MLC datasets. These datasets stem from various domains, such as text, audio, image classification, and biology, and range from small datasets with only a few instances and labels to larger datasets with thousands of instances and hundreds of labels. A detailed overview is given in Table 1, where, in addition to the number of instances (#I) and number of labels (#L), statistics regarding the label-to-instance ratio (L2IR), the percentage of unique label combinations (ULC), and the average label cardinality (card.) are given.

The train and validation folds were derived by conducting a nested 2-fold cross validation, i.e., to assess the test performance we have an outer loop of 2-fold cross validation. To tune the thresholds and select the base learner, we again split the training fold of the outer loop into train and validation sets by 2-fold cross validation. The entire process is repeated 5 times with different random seeds for the cross validation. Throughout this study, we trained and evaluated a total of 14,400 instances of BR and 649,800 base learners accordingly.

Furthermore, we consider two performance measures, namely the Hamming loss $\mathcal{L}_H$ and the macro-averaged label-wise F-measure as defined in (1) and (2), respectively. A binary prediction is obtained by thresholding the prediction of an underlying scoring classifier, which produces values in the unit interval (the higher the value, the more likely a label is considered relevant). The thresholds $\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_m)$ are optimized by a grid search considering values for $\tau_i \in [0, 1]$ and a step size of 0.01. When optimizing the thresholds, we either allow for label-wise optimization or constrain the threshold to be the same for all labels (uniform τ), i.e., $\tau_i = \tau_j$ for all $i, j \in \{1, \dots, m\}$.

In order to determine significance of results, we apply a Wilcoxon signed rank test with a threshold for the p-value of 0.05. Significant improvements of LiBRe are marked by • and significant degradations by ◦.

We executed the single BR evaluation runs, i.e., training and evaluating either on the validation or test split, on up to 300 nodes in parallel, each of them equipped with 8 CPU cores and 32 GB of RAM, and a timeout of 6 h. Due to the limitation of the memory and the runtime, some of the evaluations failed due to memory overflows or timeouts.

The implementation is based on the Java machine learning library WEKA [8] and an extension for multi-label classification called MEKA [16]. In our study, we consider a total of 20 base learners from WEKA: BayesNet (BN), DecisionStump (DS), IBk, J48, JRip (JR), KStar (KS), LMT, Logistic (L), MultilayerPerceptron

Table 1. The datasets used in this study. Furthermore, the number of instances (#I), the number of labels (#L), the label-to-instance ratio (L2IR), the percentage of unique label combinations (ULC), and the label cardinality (card.) are given.

Dataset	#I	#L	L2IR	ULC	card.	Dataset	#I	#L	L2IR	ULC	card.
arts1	7484	26	0.0035	0.08	1.65	bibtex	7395	159	0.0215	0.39	2.40
birds	645	19	0.0295	0.21	1.01	bookmarks	87856	208	0.0024	0.21	2.03
business1	11214	30	0.0027	0.02	1.60	computers1	12444	33	0.0027	0.03	1.51
education1	12030	33	0.0027	0.04	1.46	emotions	593	6	0.0101	0.05	1.87
enron-f	1702	53	0.0311	0.44	3.38	entertainment1	12730	21	0.0016	0.03	1.41
flags	194	12	0.0619	0.53	4.12	genbase	662	27	0.0408	0.05	1.25
health1	9205	32	0.0035	0.04	1.64	llog-f	1460	75	0.0514	0.21	1.18
mediamill	43907	101	0.0023	0.15	4.38	medical	978	45	0.0460	0.10	1.25
recreation1	12828	22	0.0017	0.04	1.43	reference1	8027	33	0.0041	0.03	1.17
scene	2407	6	0.0025	0.01	1.07	science1	6428	40	0.0062	0.07	1.45
social1	12111	39	0.0032	0.03	1.28	society1	14512	27	0.0019	0.07	1.67
tmc2007	28596	22	0.0008	0.05	2.16	yeast	2417	14	0.0058	0.08	4.24

(MlP), NaiveBayes (NB), NaiveBayesMultinomial (NBM), OneR (1R), PART (P), REPTree (REP), RandomForest (RF), RandomTree (RT), SMO, SimpleLogistic (SL), VotedPerceptron (VP), ZeroR (0R). All the data and source code is made available via GitHub (https://github.com/mwever/LiBRe).

5.2 Single Best Base Learner

To figure out how much we can benefit from selecting a base learner for each label individually, and whether this flexibility is beneficial at all, we define the single best base learner, subsequently referred to as SBB, as a baseline. In principle, SBB is nothing but a grid search over the portfolio of base learners (3).

When considering a base learner a, it is chosen to be employed as a base learner for every label. After training and validating the performance, we pick the base learner that performs best overall. This baseline thus gives an upper bound on the performance of what can be achieved when the base learner is not chosen for each label individually. As simple and straight-forward as it is, this baseline represents what is currently possible in implementations of MLC libraries, and already goes beyond what is most commonly done in the literature.

5.3 Optimistic Versus Validated Optimization

In addition to the results obtained by selecting the base learner(s) according to the validation performance (obtained in the inner loop of the nested cross validation), we consider optimistic performance estimates, which are obtained as follows: After having trained the base learners on the training data, we select the presumably best one, not on the basis of their performance on validation data, but based on their actual test performance (as observed in the outer loop

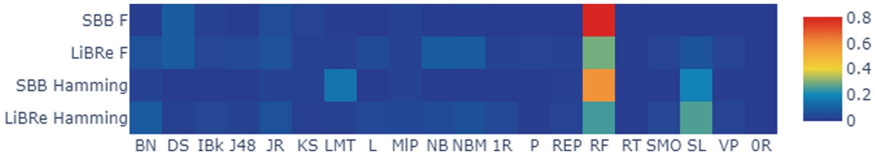

Fig. 1. The heat map shows the average share of each base learner being employed for a label with respect to the optimized performance measure: Hamming ($\mathcal{L}_H$) or the label-wise macro averaged F-measure (F).

of the nested cross-validation). Intuitively, this can be understood as a kind of "oracle" performance: Given a set of candidate predictors to choose from, the oracle anticipates which of them will perform best on the test data.

Although these performances should be treated with caution, and will certainly tend to overestimate the true generalization performance of a classifier, they can give some information about the potential of the optimization. More specifically, these optimistic performance estimates suggest an upper bound on what can be obtained by the nested optimization routine.

5.4 Results

In Fig. 1, the average share of a base learner per label is shown. From this heatmap, it becomes obvious that for the SBB baseline only a subset of base learners plays a role. However, one can also notice that the distribution of the shares varies when different performance measures are optimized. Furthermore, although random forest (RF) achieves significant shares of 0.8 for the Hamming loss and around 0.6 for the F-measure, it is not best on all the datasets. To put it differently, one still needs to optimize the base learner per dataset. This is especially true, when different performance measures are of interest.

In the case of LiBRe, it is clearly recognizable how the shares are distributed over the base learners, in contrast to SBB. For example, the shares of RF decrease to 0.29 for F-measure and to 0.25 for Hamming, respectively. Moreover, base learners that did not even play any role in SBB are now gaining in importance and are selected quite often. Although there are significant differences in the frequency of base learners being picked, there is not a single base learner in the portfolio that was never selected.

In Table 2, the results for optimizing Hamming loss are presented. The optimistic performance estimates already indicate that there is not much room for improvement. This comes at no surprise, since the datasets are already pretty much saturated, i.e., the loss is already close to 0 for most of the datasets. While LiBRe performs competitively to SBB for the setting with uniform τ, SBB compares favourably to LiBRe in the case where the thresholds can be tuned in a label-wise manner. Apparently, the additional degrees of freedom make LiBRe more prone to over-fitting, especially on smaller datasets.

In contrast to the previous results, for the optimization of the F-measure, the optimistic performance estimates already give a promising outlook on the

Table 2. Results obtained for minimizing $\mathcal{L}_H$ optimistically resp. with validation performances. Thresholds are optimized either jointly for all the labels (uniform τ) or label-wise. Best performances per setting and dataset are highlighted in bold. Significant improvements of LiBRe are marked by a • and degradations by ○.

Dataset	Optimistic uniform τ		Validated uniform τ		Optimistic label-wise τ		Validated label-wise τ	
	LiBRe	SBB	LiBRe	SBB	LiBRe	SBB	LiBRe	SBB
arts1	**0.0515**	0.0536	**0.0531**	0.0538	**0.0504**	0.0513	0.0526	**0.0525**
bibtex	**0.0118**	0.0126	**0.0126**	0.0127	**0.0115**	0.0120	0.0151	**0.0139**
birds	**0.0357**	0.0397	0.0476	**0.0420** ○	**0.0329**	0.0352	0.0470	**0.0422** ○
bookmarks	**0.0085**	0.0087	**0.0086**	0.0087 •	**0.0085**	0.0086	**0.0105**	0.0114 •
business1	**0.0233**	0.0248	**0.0241**	0.0249 •	**0.0218**	0.0223	**0.0227**	0.0228
computers1	**0.0313**	0.0334	**0.0329**	0.0335	**0.0301**	0.0306	0.0323	**0.0312**
education1	**0.0352**	0.0365	**0.0359**	0.0369 •	**0.0340**	0.0344	0.0354	**0.0349** ○
emotions	**0.1762**	0.1800	0.1926	**0.1856** ○	**0.1684**	0.1712	0.1961	**0.1875** ○
enron-f	**0.0447**	0.0474	0.0481	**0.0477**	**0.0437**	0.0445	0.0485	**0.0469** ○
entertainment1	**0.0432**	0.0466	**0.0440**	0.0469 •	**0.0414**	0.0434	**0.0430**	0.0443 •
flags	**0.1732**	0.1979	0.2134	**0.2088**	**0.1635**	0.1799	**0.2105**	0.2158
genbase	**7.0E-4**	0.0014	0.0069	**0.0016** ○	**6.0E-4**	7.0E-4	0.0070	**0.0023** ○
health1	**0.0305**	0.0344	**0.0313**	0.0347 •	**0.0282**	0.0297	0.0303	**0.0302**
llog-f	**0.0149**	0.0153	0.0202	**0.0157** ○	**0.0145**	0.0149	0.0230	**0.0178** ○
mediamill	**0.0268**	0.0270	0.0271	**0.0270**	**0.0261**	0.0262	0.0265	**0.0265**
medical	**0.0084**	0.0103	0.0115	**0.0109**	**0.0078**	0.0093	0.0136	**0.0116**
recreation1	**0.0459**	0.0472	**0.0472**	0.0473	**0.0446**	0.0453	0.0468	**0.0462**
reference1	**0.0244**	0.0264	**0.0267**	0.0268	**0.0230**	0.0245	0.0255	**0.0251**
scene	**0.0781**	0.0788	0.0817	**0.0794** ○	**0.0757**	0.0762	0.0816	**0.0800** ○
science1	**0.0281**	0.0311	**0.0311**	0.0317	**0.0269**	0.0291	0.0304	**0.0302**
social1	**0.0197**	0.0208	0.0227	**0.0210**	**0.0188**	0.0196	0.0223	**0.0200**
society1	**0.0474**	0.0495	**0.0479**	0.0496 •	**0.0444**	0.0455	**0.0455**	0.0461 •
tmc2007	**0.0601**	0.0611	**0.0600**	0.0611 •	**0.0590**	0.0611	0.0613	**0.0611**
yeast	**0.1914**	0.1926	0.2002	**0.1930** ○	**0.1886**	0.1890	0.1940	**0.1929** ○

potential for improving the generalization performance through the label-wise selection of the base learners. More precisely, they indicate that performance gains of up to 11% points are possible. Independent of the threshold optimization variant, LiBRe outperforms the SBB baseline, yielding the best performance on two third of the considered datasets, 13 improvements of which are significant in the case of uniform τ, and 11 in the case of label-wise τ. Significant degradations of LiBRe compared to SBB can only be observed for 2 respectively 3 datasets. Hence, for the F-measure, LiBRe compares favorably to the SBB baseline.

In summary, we conclude that LiBRe does indeed yield performance improvements. However, increasing the flexibility of BR also makes it more prone to over-fitting. Furthermore, these results were obtained by conducting a nested 2-fold cross validation. While keeping the computational costs of this evaluation reasonable, this implies that, for the purpose of validation, the base learners were trained on only one fourth of the original dataset. Therefore, considering nested 5-fold or 10-fold cross validation could help to reduce the observed over-fitting.

Table 3. Results for maximizing the F-measure optimistically resp. with validation performances. Thresholds are optimized either jointly for all the labels (uniform τ) or label-wise. Best performances per setting and dataset are highlighted in bold. Significant improvements of LiBRe are marked by a ● and degradations by ○.

Dataset	Optimistic uniform τ		Validated uniform τ		Optimistic label-wise τ		Validated label-wise τ	
	LiBRe	SBB	LiBRe	SBB	LiBRe	SBB	LiBRe	SBB
arts1	**0.3445**	0.2749	**0.3018**	0.2684 ●	**0.3680**	0.3211	**0.3184**	0.3001 ●
bibtex	**0.4020**	0.3027	**0.3391**	0.2998 ●	**0.4194**	0.3516	**0.3378**	0.3041 ●
birds	**0.5404**	0.4424	0.3707	**0.3961** ○	**0.5832**	0.5310	0.3843	**0.3981** ○
bookmarks	**0.2495**	0.2244	**0.2347**	0.2239 ●	**0.2646**	0.2516	**0.2435**	0.2416
business1	**0.3692**	0.2854	**0.2970**	0.2659 ●	**0.3874**	0.3197	**0.3006**	0.2790 ●
computers1	**0.3646**	0.2861	**0.3099**	0.2810 ●	**0.3833**	0.3486	**0.3224**	0.3190
education1	**0.3346**	0.2468	**0.2594**	0.2437 ●	**0.3591**	0.3022	**0.2652**	0.2612
emotions	**0.7068**	0.6946	0.6670	**0.6779**	**0.7186**	0.7135	0.6761	**0.6859** ○
enron-f	**0.2870**	0.2192	0.2056	**0.2096**	**0.3138**	0.2773	**0.2077**	0.2069
entertainment1	**0.4470**	0.3673	**0.3929**	0.3500 ●	**0.4639**	0.4049	**0.3950**	0.3774 ●
flags	**0.6280**	0.5634	**0.5230**	0.5098	**0.6474**	0.5981	**0.5150**	0.5145
genbase	**0.8126**	0.7798	0.6039	**0.7421** ○	**0.8141**	0.8119	0.6201	**0.6390**
health1	**0.4203**	0.3259	**0.3486**	0.3208 ●	**0.4312**	0.3582	**0.3464**	0.3225 ●
llog-f	**0.1569**	0.0808	**0.0730**	0.0689	**0.1834**	0.1264	**0.0744**	0.0741
mediamill	**0.3766**	0.3499	0.3481	**0.3483**	**0.4010**	0.3898	0.3543	**0.3600** ○
medical	**0.4960**	0.3852	0.3560	**0.3639**	**0.5251**	0.4523	**0.3547**	0.3208 ●
recreation1	**0.4964**	0.4224	**0.4669**	0.4160 ●	**0.5093**	0.4675	**0.4670**	0.4494 ●
reference1	**0.3185**	0.2254	**0.2477**	0.2021 ●	**0.3393**	0.2860	**0.2587**	0.2418 ●
scene	**0.7831**	0.7816	0.7734	**0.7776**	**0.7909**	0.7897	0.7759	**0.7812**
science1	**0.3824**	0.2724	**0.2928**	0.2637 ●	**0.4033**	0.3240	**0.3036**	0.2662 ●
social1	**0.3629**	0.3073	0.3046	**0.3060**	**0.3737**	0.3119	**0.3103**	0.2769 ●
society1	**0.3437**	0.2807	**0.3180**	0.2688 ●	**0.3597**	0.3382	0.3215	**0.3238**
tmc2007	**0.5659**	0.5342	**0.5467**	0.5342	**0.5782**	0.5525	**0.5656**	0.5484 ●
yeast	**0.4970**	0.4750	**0.4800**	0.4731 ●	**0.5145**	0.5084	0.4922	**0.4947**

6 Conclusion

In this paper, we have not only demonstrated the potential of binary relevance to optimize label-wise macro averaged measures, but also the importance of the base learner as a hyper-parameter for each label. Especially for the case of optimizing for F1 macro-averaged over the labels, we could achieve significant performance improvements by choosing a proper base learner in a label-wise manner. Compared to selecting the best single base learner, choosing the base learner for each label individually comes at no additional cost in terms of base learner evaluations. Moreover, the label-wise selection of base learners can be realized by a straight-forward grid search.

As the label-wise choice of a base learner has already led to considerable performance gains, we plan to examine to what extent the optimization of the hyper-parameters of those base learners can lead to further improvements. Furthermore, we want to increase the efficiency of the tuning by replacing the grid search with a heuristic approach.

Another direction of future work concerns the avoidance of over-fitting effects due to an overly excessive flexibility of LiBRe. As already explained, the restriction to a single base learner can be seen as a kind of regularization, which, however, appears to be too strong, at least according to our results. On the other side, the full flexibility of LiBRe does not always pay off either. An interesting compromise could be to restrict the number of different base learners used by LiBRe to a suitable value $k \in \{1, \ldots, m\}$. Technically, this comes down to finding the $\arg\min$ in (4), not over $\boldsymbol{a} \in \mathcal{A}^m$, but over $\{\boldsymbol{a} \in \mathcal{A}^m \mid \#\{a_1, \ldots, a_m\} \leq k\}$.

Acknowledgement. This work was supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing" (SFB 901/3 project no. 160364472). The authors also gratefully acknowledge support of this project through computing time provided by the Paderborn Center for Parallel Computing (PC^2).

References

1. Antonucci, A., Corani, G., Mauá, D.D., Gabaglio, S.: An ensemble of Bayesian networks for multilabel classification. In: IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013, pp. 1220–1225 (2013)
2. Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G.: Hierarchical multi-label prediction of gene function. Bioinformatics **22**(7), 830–836 (2006). https://doi.org/10.1093/bioinformatics/btk048
3. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recogn. **37**(9), 1757–1771 (2004). https://doi.org/10.1016/j.patcog.2004.03.009
4. Cabral, R.S., la Torre, F.D., Costeira, J.P., Bernardino, A.: Matrix completion for multi-label image classification. In: 25th Annual Conference on Neural Information Processing Systems 2011, Advances in Neural Information Processing Systems, Granada, Spain, vol. 24, pp. 190–198 (2011)
5. Charte, F., Rivera, A.J., del Jesús, M.J., Herrera, F.: Addressing imbalance in multilabel classification: measures and random resampling algorithms. Neurocomputing **163**, 3–16 (2015). https://doi.org/10.1016/j.neucom.2014.08.091
6. Cherman, E.A., Metz, J., Monard, M.C.: Incorporating label dependency into the binary relevance framework for multi-label classification. Exp. Syst. Appl. **39**(2), 1647–1655 (2012). https://doi.org/10.1016/j.eswa.2011.06.056
7. Dembczynski, K., Waegeman, W., Cheng, W., Hüllermeier, E.: On label dependence and loss minimization in multi-label classification. Mach. Learn. **88**(1–2), 5–45 (2012). https://doi.org/10.1007/s10994-012-5285-8
8. Frank, E., Hall, M.A., Witten, I.H.: The Weka workbench. Online appendix. In: Frank, E., Hall, M.A., Witten, I.H. (eds.) Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Cambridge (2016)
9. Luaces, O., Díez, J., Barranquero, J., del Coz, J.J., Bahamonde, A.: Binary relevance efficacy for multilabel classification. Prog. AI **1**(4), 303–313 (2012). https://doi.org/10.1007/s13748-012-0030-x
10. Madjarov, G., Kocev, D., Gjorgjevikj, D., Dzeroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recogn. **45**(9), 3084–3104 (2012). https://doi.org/10.1016/j.patcog.2012.03.004

11. Montañés, E., Senge, R., Barranquero, J., Quevedo, J.R., del Coz, J.J., Hüllermeier, E.: Dependent binary relevance models for multi-label classification. Pattern Recogn. **47**(3), 1494–1508 (2014). https://doi.org/10.1016/j.patcog.2013.09.029
12. Moyano, J.M., Galindo, E.L.G., Cios, K.J., Ventura, S.: Review of ensembles of multi-label classifiers: models, experimental study and prospects. Inf. Fusion **44**, 33–45 (2018). https://doi.org/10.1016/j.inffus.2017.12.001
13. Pillai, I., Fumera, G., Roli, F.: Threshold optimisation for multi-label classifiers. Pattern Recogn. **46**(7), 2055–2065 (2013). https://doi.org/10.1016/j.patcog.2013.01.012
14. Qi, G., Hua, X., Rui, Y., Tang, J., Mei, T., Zhang, H.: Correlative multi-label video annotation. In: Proceedings of the 15th International Conference on Multimedia 2007, Augsburg, Germany, 24–29 September 2007, pp. 17–26 (2007). https://doi.org/10.1145/1291233.1291245
15. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Mach. Learn. **85**(3), 333–359 (2011). https://doi.org/10.1007/s10994-011-5256-5
16. Read, J., Reutemann, P., Pfahringer, B., Holmes, G.: MEKA: a multi-label/multi-target extension to Weka. J. Mach. Learn. Res. **17**(21), 667–671 (2016)
17. de Sá, A.G.C., Freitas, A.A., Pappa, G.L.: Automated selection and configuration of multi-label classification algorithms with grammar-based genetic programming. Parallel Prob. Solving Nat. - PPSN XV **2018**, 308–320 (2018). https://doi.org/10.1007/978-3-319-99259-4_25
18. Sanden, C., Zhang, J.Z.: Enhancing multi-label music genre classification through ensemble techniques. In: Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, pp. 705–714 (2011). https://doi.org/10.1145/2009916.2010011
19. Schapire, R.E., Singer, Y.: BoosTexter: a boosting-based system for text categorization. Mach. Learn. **39**(2/3), 135–168 (2000). https://doi.org/10.1023/A:1007649029923
20. Sucar, L.E., Bielza, C., Morales, E.F., Hernandez-Leal, P., Zaragoza, J.H., Larrañaga, P.: Multi-label classification with bayesian network-based chain classifiers. Pattern Recogn. Lett. **41**, 14–22 (2014). https://doi.org/10.1016/j.patrec.2013.11.007
21. Tahir, M.A., Kittler, J., Bouridane, A.: Multi-label classification using stacked spectral kernel discriminant analysis. Neurocomputing **171**, 127–137 (2016). https://doi.org/10.1016/j.neucom.2015.06.023
22. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. IJDWM **3**(3), 1–13 (2007). https://doi.org/10.4018/jdwm.2007070101
23. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer, Boston (2010). https://doi.org/10.1007/978-0-387-09823-4_34
24. Wever, M., Mohr, F., Hüllermeier, E.: Automated multi-label classification based on ML-Plan. CoRR abs/1811.04060 (2018)
25. Wever, M.D., Mohr, F., Tornede, A., Hüllermeier, E.: Automating multi-label classification extending ML-Plan (2019)
26. Xue, X., Zhang, W., Zhang, J., Wu, B., Fan, J., Lu, Y.: Correlative multi-label multi-instance image annotation. In: IEEE International Conference on Computer Vision, pp. 651–658 (2011). https://doi.org/10.1109/ICCV.2011.6126300
27. Zhang, M., Li, Y., Liu, X.: Towards class-imbalance aware multi-label learning. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, 2015, pp. 4041–4047 (2015)

28. Zhang, M.-L., Li, Y.-K., Liu, X.-Y., Geng, X.: Binary relevance for multi-label learning: an overview. Frontiers Comput. Sci. **12**(2), 191–202 (2018). https://doi.org/10.1007/s11704-017-7031-7
29. Zhang, M., Zhou, Z.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2014). https://doi.org/10.1109/TKDE.2013.39

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.