Towards Class-Imbalance Aware Multi-Label Learning

Min-Ling Zhang, *Senior Member, IEEE*, Yu-Kun Li, Hao Yang, and Xu-Ying Liu

***Abstract*—Multi-label learning deals with training examples each represented by a single instance while associated with multiple class labels. Due to the exponential number of possible label sets to be considered by the predictive model, it is commonly assumed that label correlations should be well exploited to design an effective multi-label learning approach. On the other hand, *class-imbalance* stands as an intrinsic property of multi-label data which significantly affects the generalization performance of the multi-label predictive model. For each class label, the number of training examples with positive labeling assignment is generally much less than those with negative labeling assignment. To deal with the class-imbalance issue for multi-label learning, a simple yet effective class-imbalance aware learning strategy called cross-coupling aggregation (COCOA) is proposed in this article. Specifically, COCOA works by leveraging the exploitation of label correlations as well as the exploration of class-imbalance simultaneously. For each class label, a number of multiclass imbalance learners are induced by randomly coupling with other labels, whose predictions on the unseen instance are aggregated to determine the corresponding labeling relevancy. Extensive experiments on 18 benchmark datasets clearly validate the effectiveness of COCOA against state-of-the-art multi-label learning approaches especially in terms of imbalance-specific evaluation metrics.**

***Index Terms*—Class-imbalance, cross-coupling aggregation (COCOA), machine learning, multi-label learning.**

I. INTRODUCTION

IN MULTI-LABEL learning, each real-world object is represented by a single instance while associated with multiple class labels [21], [64]. Formally, let $\mathcal{X} = \mathbb{R}^d$ denote the instance space of d-dimensional feature vectors and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ denote the label space consisting of q class labels. The task of multi-label learning is to induce a predictive model $h : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ from the multi-label training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq N\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional instance and $Y_i \subseteq \mathcal{Y}$ is the set of relevant labels associated with $\boldsymbol{x}_i$. Given an unseen instance $\boldsymbol{x}$, $h(\boldsymbol{x})$ returns the set of relevant labels for the unseen instance.

In the past decade, multi-label learning techniques have been widely applied to model real-world objects with rich semantics, such as text [27], [33], [39], [48]; image [6], [15], [59], [66], [69]; audio [5], [37]; video [26], [56]; gene [7], [41], [55]; etc. To learn from multi-label data, the key challenge lies in the huge output space which contains an exponential number (2^q) of possible label sets for prediction. Therefore, a common practice for designing effective multi-label learning approaches is trying to exploit correlations among class labels to facilitate the learning procedure [21], [64]. Roughly speaking, existing approaches can be grouped into three categories based on the *order of correlations* being considered, that is, first-order approaches considering independence among class labels, second-order approaches considering correlations between a pair of class labels, and high-order approaches considering correlations among all class labels or subsets of class labels.

Nonetheless, the intrinsic property of *class-imbalance* needs to be taken into full consideration as well for learning from multi-label data [18]. Specifically, for each class label $y_j \in \mathcal{Y}$, let $\mathcal{D}_j^+ = \{(\boldsymbol{x}_i, +1) \mid y_j \in Y_i, 1 \leq i \leq N\}$ and $\mathcal{D}_j^- = \{(\boldsymbol{x}_i, -1) \mid y_j \notin Y_i, 1 \leq i \leq N\}$ denote the set of examples with *positive* labeling and *negative* labeling w.r.t. y_j, respectively. Correspondingly, class skewness between $\mathcal{D}_j^+$ and $\mathcal{D}_j^-$ can be measured by the *imbalance ratio*

$$\mathrm{ImR}_j = \frac{\max\left(\left|\mathcal{D}_j^+\right|, \left|\mathcal{D}_j^-\right|\right)}{\min\left(\left|\mathcal{D}_j^+\right|, \left|\mathcal{D}_j^-\right|\right)}.$$

As an intrinsic property of multi-label data, ImR_j would be high in most cases.[1] For instance, among the 18 benchmark multi-label datasets used in this article (Table II), the `average` imbalance ratio across the label space (i.e., $(1/q) \sum_{j=1}^{q} \mathrm{ImR}_j$) ranges from 2.1 to 32.2 (with 13 of them greater than 5.0), and the `maximum` imbalance ratio across the label space (i.e., $\max_{1\leq j\leq q} \mathrm{ImR}_j$) ranges from 3.0 to 50.0 (with 15 of them greater than 10.0).

It is well known that class imbalance acts as a major threat to compromise the training procedure of machine learning techniques, which would lead to performance degradation for existing multi-label learning approaches [24], [64]. Therefore,

Manuscript received February 4, 2020; revised July 17, 2020; accepted September 20, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1004300; in part by the National Science Foundation of China under Grant 61573104; in part by the China University S&T Innovation Plan Guided by the Ministry of Education; and in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization. This article was recommended by Associate Editor S. Ventura. *(Corresponding author: Min-Ling Zhang.)*

Min-Ling Zhang, Hao Yang, and Xu-Ying Liu are with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and also with the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China (e-mail: zhangml@seu.edu.cn; yang_h@seu.edu.cn; liuxy@seu.edu.cn).

Yu-Kun Li is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and also with the Business Group of Natural Language Processing, Baidu Inc., Beijing 100085, China (e-mail: liyukun01@baidu.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.3027509

[1] Generally, $|\mathcal{D}_j^+| < |\mathcal{D}_j^-|$ holds.

the desirable multi-label training procedure should be aware of the exploitation of label correlations as well as the exploration of class imbalance. In light of this consideration, a simple yet effective class-imbalance aware learning strategy called COCOA, that is, *cross-coupling aggregation*, is proposed to learning from multi-label data. For each class label, COCOA generates a number of multiclass imbalance learners each induced by coupling one randomly chosen class label with the current label. After that, the relevancy of each class label w.r.t. the unseen instance is determined by aggregating predictions yielded by the multiclass imbalance learners. Comprehensive experiments on 18 benchmark datasets show that COCOA achieves highly competitive performance against state-of-the-art comparing algorithms, especially in terms of evaluation metrics specific to the class-imbalance scenario.

The remainder of this article is organized as follows. Section II reviews existing works related to COCOA. Section III presents the technical details of the proposed approach. Section IV reports the experimental results of comparative studies. Finally, Section V concludes and indicates several issues for future work.

II. RELATED WORK

The goal of multi-label learning is to learn a mapping from the instance space to the power set of the label space. Due to the exponential number of possible label subsets to be predicted, it is essential to exploit label correlations for predictive model induction. Roughly speaking, existing approaches to learning from multi-label data can be categorized based on the order of correlations being considered.

First-order approaches work in a label-by-label style by ignoring the co-existence of other labels. Specifically, one binary classification model is induced to make prediction w.r.t. each class label independently [3], [62], [63]. Second-order approaches work by considering pairwise relations between labels. For instance, it is natural to consider the ranking relation that relevant labels should have larger modeling output than irrelevant labels [16], [19], [25], or the interaction relation that label pairs with a high co-occurrence rate would have strong label correlation [20]. High-order approaches work by considering relations among a number of labels. For instance, the multi-label predictive model can be trained by exploiting the assumption that high-order relations exist among all labels [28], [47] or a subset of labels [53] in the label space. More detailed discussions on existing multi-label learning algorithms can be found in recent review literature [21], [64], [68].

To address the class-imbalance issue in multi-label learning, one can take the *transformation* strategy by applying existing class-imbalance learning techniques to the binary or multiclass learning problems transformed from the multi-label learning problem. Binary relevance [62] serves as the most straightforward solution where the original multi-label learning problem is decomposed into a number of independent binary learning problems, one per class label. Given the decomposed binary learning problems, the skewness between the majority class and minority class can be directly handled by employing popular binary imbalance learning techniques. For undersampling techniques, the training examples from the majority class can be undersampled to form new binary training set by random sampling [52] or exploiting the Tomek link [42]. For oversampling techniques, the training examples from the minority class can be oversampled to form a new binary training set by random sampling [8], nearest neighbor informed oversampling [49], or synthetic instance generation [9], [31], [32], [36]. Furthermore, oversampling techniques can be combined with instance editing mechanism to decouple highly imbalanced labels for class-imbalance multi-label learning [10].

To fulfill the transformation strategy, label powerset [64] serves as another straightforward solution which transforms the original multi-label learning problem into a multiclass problem by treating any distinct label combination appearing in the training set as a new class. After that, the skewness among the transformed classes can be directly handled by employing off-the-shelf multiclass imbalance learning techniques [1], [30], [35], [40], [57], [61]. Although label correlations have been explicitly addressed through the transformation process, the number of transformed classes (upper bounded by $\min(N, 2^q)$) would be prohibitively large for any multiclass learner to work well.

Other than the transformation strategy, one can also employ the *adaptation* strategy of endowing multi-label learning algorithms with the ability of handling class imbalance via tailored adaptations. For instance, the classification threshold can be determined based on a held-out validation set [17] or optimized with extra learning procedures [44], [46]. Rather than only tuning the thresholding parameter, a more sophisticated solution is to train the multi-label predictive model by directly optimizing imbalance-aware metric such as F-measure [13], [43], [45]. Furthermore, it is also feasible to customize algorithmic choices of a specific classification model for class-imbalance multi-label learning, such as designing oblique splitting function for decision trees [12], calibrating regularization hyperparameter for support vector machines [22], or adjusting hyperedge weights for hypernetwork models [51].

Most works on class-imbalance multi-label learning assume complete labeling information for the training examples, that is, all the relevant labels for each training example are available for model induction. However, the process of acquiring labels for training examples is generally costly, especially under a multi-label learning scenario where multiple labels need to be annotated for the training example. Therefore, it is of practical importance to consider the issue of missing labels for class-imbalance multi-label learning, which can be tackled by imposing label consistency via submodular minimization [29], [60] or label regularization via accelerated proximal gradient [4]. On the other hand, the task of learning a multi-label predictive model may take place under the streaming scenario where training examples arrive incrementally. The issue of concept drift and class imbalance for multi-label stream learning can be tackled by maintaining two windows for the positive and negative examples of each label, respectively [50].

It is worth noting that COCOA makes use of ensemble learning to aggregate the predictions of a number of (i.e., K) randomly generated imbalance learners, each induced by pairwise cross-coupling between the current label and one randomly chosen class label. There have been multi-label learning methods which also utilize ensemble learning [65], [67] to deal with their inherent random factors, such as ensembling chaining classifier with random order [34], [47] or ensembling multiclass learner derived from random k-labelsets [38], [53]. It is worth noting that although pairwise cross-coupling only considers second-order correlations among labels, the overall label correlations exploited by COCOA are actually high order as controlled by the parameter K. Specifically, COCOA fulfills high-order label correlations by imposing random pairwise cross-coupling for K times instead of combining all K coupling labels simultaneously, as the latter strategy may lead to severe class imbalance due to the combinatorial effects.

III. COCOA APPROACH

Following the notations in Section I, the task of multi-label learning is to induce a multi-label predictive model $f : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ from the multi-label training examples $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq N)\}$. Generally, this task is accomplished by learning a set of q real-valued functions $f_j : \mathcal{X} \rightarrow \mathbb{R}$ $(1 \leq j \leq q)$, where $f_j(\boldsymbol{x})$ returns the *confidence* of associating class label y_j with an instance $\boldsymbol{x} \in \mathcal{X}$. Along with the thresholding function $t_j : \mathcal{X} \rightarrow \mathbb{R}$, the set of relevant labels for $\boldsymbol{x}$ is predicted as

$$h(\boldsymbol{x}) = \{y_j \mid f_j(\boldsymbol{x}) > t_j(\boldsymbol{x}),\ 1 \leq j \leq q\}. \tag{1}$$

An intuitive way to induce $f_j(\cdot)$ is to learn from the binary training set $\mathcal{D}_j$ derived from $\mathcal{D}$ for the jth class label y_j [62]

$$\mathcal{D}_j = \{(\boldsymbol{x}_i, \phi(Y_i, y_j)) \mid 1 \leq i \leq N\}$$
$$\text{where} \quad \phi(Y_i, y_j) = \begin{cases} +1, & \text{if } y_j \in Y_i \\ -1, & \text{otherwise.} \end{cases} \tag{2}$$

Correspondingly, the derived binary training set consists of positive training examples ($\mathcal{D}_j^+$) and negative training examples ($\mathcal{D}_j^-$), that is, $\mathcal{D}_j = \mathcal{D}_j^+ \bigcup \mathcal{D}_j^-$. To account for the skewness between $\mathcal{D}_j^+$ and $\mathcal{D}_j^-$, one straightforward solution is to apply some *binary-class imbalance* learner $\mathcal{B}$ on $\mathcal{D}_j$ to induce a binary classifier g_j, that is, $g_j \leftarrow \mathcal{B}(\mathcal{D}_j)$. Then, the real-valued function $f_j(\cdot)$ can be instantiated as $f_j(\boldsymbol{x}) = g_j(+1 \mid \boldsymbol{x})$, where $g_j(+1 \mid \boldsymbol{x})$ denotes the predictive confidence that $\boldsymbol{x}$ should be regarded as a positive example w.r.t. y_j.

Although it is feasible to explore the class-imbalance issue following the above intuitive way, the predictive model $f_j(\cdot)$ for each class label y_j is actually built in an independent manner. To exploit label correlations for model induction, COCOA proposes to considering correlations between one random class label y_k $(k \neq j)$ with y_j via cross-coupling. Specifically, given the label pair (y_j, y_k), a multiclass training set $\mathcal{D}_{jk}$ can be derived from $\mathcal{D}$

$$\mathcal{D}_{jk} = \{(\boldsymbol{x}_i, \psi(Y_i, y_j, y_k)) \mid 1 \leq i \leq N\}$$
$$\text{where} \quad \psi(Y_i, y_j, y_k) = \begin{cases} 0, & \text{if } y_j \notin Y_i \text{ and } y_k \notin Y_i \\ +1, & \text{if } y_j \notin Y_i \text{ and } y_k \in Y_i \\ +2, & \text{if } y_j \in Y_i \text{ and } y_k \notin Y_i \\ +3, & \text{if } y_j \in Y_i \text{ and } y_k \in Y_i. \end{cases} \tag{3}$$

Here, the class label $\psi(Y_i, y_j, y_k)$ for the derived four-class learning problem is determined by the joint assignment of y_j and y_k w.r.t. Y_i.

Note that although the exploitation of label correlations can be enabled by making use of $\mathcal{D}_{jk}$ in the learning process, the issue of class imbalance becomes more pronounced by jointly considering y_j and y_k. Without loss of generality, suppose that positive examples $\mathcal{D}_j^+$ (or $\mathcal{D}_k^+$) correspond to the *minority* class in the binary training set $\mathcal{D}_j$ (or $\mathcal{D}_k$). Accordingly, for the four-class training set $\mathcal{D}_{jk}$, the first class $(\psi(Y_i, y_k, y_k) = 0)$ and the fourth class $(\psi(Y_i, y_k, y_k) = +3)$ would contain the largest and the smallest number of examples. In constrast to the imbalance ratios ImR_j and ImR_k in binary training sets $\mathcal{D}_j$ and $\mathcal{D}_k$, the imbalance ratio between the largest class and the smallest class in $\mathcal{D}_{jk}$ would roughly increase to $\text{ImR}_j \cdot \text{ImR}_k$.

To deal with this potential problem, COCOA employs a simple strategy of transforming the four-class dataset $\mathcal{D}_{jk}$ into a tri-class dataset $\mathcal{D}_{jk}^{\textbf{tri}}$ by merging the third class and the fourth class (both with positive assignment for y_j)

$$\mathcal{D}_{jk}^{\textbf{tri}} = \{(\boldsymbol{x}_i, \psi^{\textbf{tri}}(Y_i, y_j, y_k)) \mid 1 \leq i \leq N\}$$
$$\text{where} \quad \psi^{\textbf{tri}}(Y_i, y_j, y_k) = \begin{cases} 0, & \text{if } y_j \notin Y_i \text{ and } y_k \notin Y_i \\ +1, & \text{if } y_j \notin Y_i \text{ and } y_k \in Y_i \\ +2, & \text{if } y_j \in Y_i. \end{cases} \tag{4}$$

Here, for the newly merged class $(\psi^{\textbf{tri}}(Y_i, y_j, y_k) = +2)$, its imbalance ratios w.r.t. the first class $(\psi^{\textbf{tri}}(Y_i, y_j, y_k) = 0)$ and the second class $(\psi^{\textbf{tri}}(Y_i, y_j, y_k) = +1)$ would roughly be $[(\text{ImR}_j \cdot \text{ImR}_k)/(1 + \text{ImR}_k)]$ and $[\text{ImR}_j/(1 + \text{ImR}_k)]$, which is much smaller than the worst-case imbalance ratio $\text{ImR}_j \cdot \text{ImR}_k$ in the four-class training set.

Based on some *multiclass imbalance* learner $\mathcal{M}$, one multiclass classifier g_{jk} can be induced by applying $\mathcal{M}$ on $\mathcal{D}_{jk}^{\textbf{tri}}$, that is, $g_{jk} \leftarrow \mathcal{M}(\mathcal{D}_{jk}^{\textbf{tri}})$. Correspondingly, let $g_{jk}(+2 \mid \boldsymbol{x})$ denote the predictive confidence that $\boldsymbol{x}$ should have positive assignment w.r.t. y_j (regardless of $\boldsymbol{x}$ having positive or negative assignment w.r.t. y_k). For each class label y_j, COCOA draws a random subset of K class labels $\mathcal{I}_K \subset \mathcal{Y} \backslash \{y_j\}$ for pairwise cross-coupling. The real-valued function $f_j(\cdot)$ is then instantiated by aggregating the predictive confidences of K multiclass imbalance learners

$$f_j(\boldsymbol{x}) = \sum_{y_k \in \mathcal{I}_K} g_{jk}(+2 \mid \boldsymbol{x}). \tag{5}$$

Furthermore, COCOA chooses to set the thresholding function $t_j(\cdot)$ as a constant function $t_j(\boldsymbol{x}) = a_j$, where any example $\boldsymbol{x}$ is predicted to be positive for y_j if $f_j(\boldsymbol{x}) > a_j$ and negative otherwise. Here, the "goodness" of a_j can be evaluated based on certain metric which measures how well f_j classifies examples in $\mathcal{D}_j$ by using a_j as the bipartition threshold. Specifically, COCOA employs the F-measure metric (i.e., harmonic mean of precision and recall) which is popular for evaluating the performance of binary classifier, especially for the case of skewed class distribution.

Let $F(f_j, a, \mathcal{D}_j)$ denote the F-measure value achieved by applying $\{f_j, a\}$ over the binary training set $\mathcal{D}_j$, that is

$$F(f_j, a, \mathcal{D}_j) = \frac{2 \cdot P(f_j, a, \mathcal{D}_j) \cdot R(f_j, a, \mathcal{D}_j)}{P(f_j, a, \mathcal{D}_j) + R(f_j, a, \mathcal{D}_j)} \tag{6}$$

TABLE I
PSEUDOCODE OF COCOA

```
Inputs:
$\mathcal{D}$:   the multi-label training set $\{(\boldsymbol{x}_i, Y_i) \mid 1 \le i \le N\}$
      $(\boldsymbol{x}_i \in \mathcal{X}, Y_i \subseteq \mathcal{Y}, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{y_1, y_2, \ldots, y_q\})$
$\mathcal{M}$:   the multi-class imbalance learner
$K$:   the number of coupling class labels
$\boldsymbol{x}$:   the test example ($\boldsymbol{x} \in \mathcal{X}$)
Outputs:
$Y$:   the predicted label set for $\boldsymbol{x}$
Process:
1:  for $j = 1$ to $q$ do
2:      Form the binary training set $\mathcal{D}_j$ according to Eq.(2);
3:      Draw a random subset $\mathcal{I}_K \subset \mathcal{Y} \setminus \{y_j\}$ containing $K$ class
        labels;
4:      for $y_k \in \mathcal{I}_K$ do
5:          Form the tri-class training set $\mathcal{D}_{jk}^{\mathbf{tri}}$ according to Eq.(4);
6:          $g_{jk} \leftarrow \mathcal{M}(\mathcal{D}_{jk}^{\mathbf{tri}})$;
7:      end for
8:      Set the real-valued function $f_j(\cdot)$ according to Eq.(5);
9:      Set the constant thresholding function $t_j(\cdot)$ with constant
        $a_j$ being determined according to Eqs.(6) and (7);
10: end for
11: Return $Y = h(\boldsymbol{x})$ according to Eq.(1);
```

where

$$P(f_j, a, \mathcal{D}_j) = \frac{\sum_{i=1}^{N} [\![f_j(\boldsymbol{x}_i) > a_j]\!] \cdot [\![y_j \in Y_i]\!]}{\sum_{i=1}^{N} [\![f_j(\boldsymbol{x}_i) > a_j]\!]}$$

$$R(f_j, a, \mathcal{D}_j) = \frac{\sum_{i=1}^{N} [\![f_j(\boldsymbol{x}_i) > a_j]\!] \cdot [\![y_j \in Y_i]\!]}{\sum_{i=1}^{N} [\![y_j \in Y_i]\!]}.$$

Here, $[\![\pi]\!]$ returns 1 if predicate π holds and 0 otherwise. The thresholding constant a_j is determined by maximizing the corresponding F-measure

$$a_j = \arg\max_{a \in \mathbb{R}} F(f_j, a, \mathcal{D}_j). \tag{7}$$

The complete procedure of COCOA is summarized in Table I. For each class label $y_j \in \mathcal{Y}$, a total of K multiclass imbalance classifiers (steps 3–7) are induced by manipulating the multi-label training set $\mathcal{D}$ via random cross-coupling. After that, the predictive model for y_j is produced by calibrating the aggregated predictive confidences of the induced multiclass classifiers w.r.t. the thresholding value (steps 8 and 9). Finally, the predicted label set for the test example is obtained by querying the predictive models of all class labels (step 11).[2]

IV. EXPERIMENTS

In this section, the effectiveness of COCOA is thoroughly investigated via extensive experimental studies. First, the experimental setup, including datasets, comparing algorithms, and evaluation metrics, are introduced. Second, detailed experimental results as well as statistical comparisons are reported. Third, several properties of the proposed COCOA approach are further analyzed.

A. Experimental Setup

1) Datasets: To comprehensively evaluate the performance of COCOA, a total of 18 benchmark multi-label datasets have been collected for experimental studies. For each multi-label dataset $\mathcal{S}$, we use $|\mathcal{S}|$, $L(\mathcal{S})$, $\dim(\mathcal{S})$, and $F(\mathcal{S})$ to represent its number of examples, number of class labels, number of features, and feature type, respectively. In addition, several multi-label statistics [47] are further used to characterize the properties of $\mathcal{S}$.

1) $LCard(\mathcal{S}) = (1/|\mathcal{S}|) \sum_{(\boldsymbol{x},Y)\in\mathcal{S}} |Y|$*: Label cardinality* which measures the average number of relevant labels per example.
2) $LDen(\mathcal{S}) = LCard(\mathcal{S})/L(\mathcal{S})$*: Label density* which normalizes label cardinality by the total number of class labels.
3) $DL(\mathcal{S}) = |\{Y|(\boldsymbol{x}, Y) \in \mathcal{S}\}|$: *Distinct label sets* which measures the number of distinct relevant label sets.
4) $PDL(\mathcal{S}) = DL(\mathcal{S})/|\mathcal{S}|$*: The proportion of distinct label sets* which normalizes distinct label sets by the number of examples.

As discussed in Section I, let ImR_j represent the imbalance ratio on the jth class label ($1 \le j \le q$). The level of class imbalance on $\mathcal{S}$ can be characterized by the `average` imbalance ratio $(1/q) \sum_{j=1}^{q} \text{ImR}_j$, the `minimum` imbalance ratio $\min_{1\le j\le q} \text{ImR}_j$, and the `maximum` imbalance ratio $\max_{1\le j\le q} \text{ImR}_j$ across the label space.[3]

Table II summarizes the characteristics of the experimental datasets, which are roughly ordered according to $|\mathcal{S}|$. As shown in Table II, the 18 datasets exhibit diversified properties in terms of different multi-label statistics. In addition, these datasets cover a broad range of scenarios, including text (`Medical`, `Enron`, `Rcv1`, `Bibtex`, `Eurlex-sm`, `Tmc2007`),[4] audio (`CAL500`, `Emotions`, `Birds`), image (`Scene`, `Corel5k`), video (`Mediamill`), biology (`Yeast`), etc.

2) Comparing Algorithms: Two series of comparing algorithms are employed in this article for experimental studies. First, the performance of COCOA is compared against several approaches which are capable of dealing with the class-imbalance issue in multi-label data.

1) THRSEL *[17]:* The multi-label learning problem is decomposed into q binary learning problems, and the classification threshold is tuned by maximizing F-measure over held-out validation set.
2) IRUS *[52]:* The multi-label learning problem is decomposed into q binary learning problems, and the majority class in each binary problem is randomly *undersampled* to form the new binary training set. The random undersampling procedure is repeated multiple times to derive an ensemble of binary classifiers to yield a composite decision boundary between the majority class and the minority class.

[2] The code package for COCOA is publicly available at http://palm.seu.edu.cn/zhangml/files/COCOA.rar.

[3] As a common practice in class-imbalance studies [24], the case of *extreme imbalance* is not considered in this article. Specifically, any class label with rare appearance (less than 20 positive examples) or with overly high imbalance ratio ($\text{ImR}_j \ge 50$) is excluded from the label space.

[4] Dimensionality reduction is performed on text datasets by retaining features with high document frequency.

TABLE II
CHARACTERISTICS OF THE BENCHMARK MULTI-LABEL DATASETS

Data set	$\lvert\mathcal{S}\rvert$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$F(\mathcal{S})$	$LCard(\mathcal{S})$	$LDen(\mathcal{S})$	$DL(\mathcal{S})$	$PDL(\mathcal{S})$	Imbalance Ratio		
									min	max	avg
CAL500	502	68	124	numeric	25.058	0.202	502	1.000	1.040	24.390	3.846
Emotions	593	72	6	numeric	1.869	0.311	27	0.046	1.247	3.003	2.146
Birds	645	260	12	numeric	0.891	0.074	93	0.144	5.262	31.250	15.493
Medical	978	144	14	numeric	1.075	0.077	42	0.043	2.674	43.478	11.236
LLOG	1460	100	18	nominal	0.851	0.047	109	0.075	7.538	46.097	24.981
Enron	1702	50	24	nominal	3.113	0.130	547	0.321	1.000	43.478	5.348
Image	2000	294	5	numeric	1.236	0.247	20	0.010	2.448	3.890	3.116
Scene	2407	294	6	numeric	1.074	0.179	15	0.006	3.521	5.618	4.566
Yeast	2417	103	13	numeric	4.233	0.325	189	0.078	1.328	12.500	2.778
Slashdot	3782	53	14	nominal	1.134	0.081	118	0.031	5.464	35.714	10.989
Corel5k	5000	499	44	nominal	2.214	0.050	1037	0.207	3.460	50.000	17.857
Rcv1-s1	6000	472	42	numeric	2.458	0.059	574	0.096	3.342	49.000	24.966
Rcv1-s2	6000	472	39	numeric	2.170	0.056	489	0.082	3.216	47.780	26.370
Rcv1-s3	6000	472	39	numeric	2.150	0.055	488	0.081	3.205	49.000	26.647
Bibtex	7395	183	26	nominal	0.934	0.036	377	0.051	6.097	47.974	32.245
Eurlex-sm	19348	250	27	numeric	1.492	0.055	497	0.026	3.509	47.619	16.393
Tmc2007	28596	500	15	nominal	2.100	0.140	637	0.022	1.447	34.483	5.848
Mediamill	43907	120	29	numeric	4.010	0.138	3540	0.079	1.748	45.455	7.092

3) SMOTE-EN*:* The multi-label learning problem is decomposed into q binary learning problems, and the minority class in each binary problem is *oversampled* via the SMOTE method [11] to form the new binary training set. Considering that COCOA utilizes ensemble learning in its learning process, its ensemble version SMOTE-EN is employed for comparative study.

4) RML *[43]:* Other than integrating binary decomposition with threshold selection or undersampling/oversampling, another way to handle class imbalance is to design a learning system which can directly optimize imbalance-specific metric. Here, the RML approach is employed as another comparing algorithm, which maximizes macro-averaging F-measure on multi-label data via convex relaxation.

Second, the performance of COCOA is compared against several well-established multi-label learning algorithms [64], including first-order approach binary relevance (BR) [62], second-order approach calibrated label ranking (CLR) [19], and high-order approaches ensemble of classifier chains (ECC) [47] and random k-labelsets (RAKEL) [53].

All the comparing algorithms are instantiated with the following configurations.

1) For IRUS and SMOTE-EN, decision tree is used as the base learner due to its popularity in class-imbalance studies [24]. Specifically, J48 decision tree (C4.5 implementation in the widely used Weka platform) is adopted as their base learner [23].
2) For RML, the original implementation provided in the literature is used.
3) For the second series of algorithms (BR, CLR, ECC, and RAKEL), their implementations provided by the MULAN multi-label learning library (upon Weka platform) with suggested parameter configurations [54] are adopted.
4) For COCOA, the multiclass imbalance learners $\mathcal{M}$ are implemented in Weka using J48 decision tree with undersampling [23], and the number of coupling class labels is set as $K = \min(q-1, 10)$.

Furthermore, for comparing algorithms incorporating ensemble strategy (IRUS, SMOTE-EN, and ECC), their ensemble size is set to be 100 to yield competitive performance.

3) Evaluation Metrics: Given the multi-label dataset $\mathcal{S}$, let $f_j(\cdot)$ and $t_j(\cdot)$ denote the real-valued function and thresholding function for each class label y_j ($1 \le j \le q$). Under class-imbalance scenarios, *F-measure* and *area under the ROC curve* (AUC) are the mostly used evaluation metrics which can provide more insights on the classification performance than conventional metrics such as accuracy [24]. In this article, the multi-label classification performance is accordingly evaluated by *macro-averaging* the metric values across all class labels [64].

1) Macro-averaging F-measure (F_{macro})

$$F_{\text{macro}} = \frac{1}{q}\sum_{j=1}^{q} F_j, \text{ where}$$

$$F_j = \frac{2P_j \cdot R_j}{P_j + R_j}$$

$$P_j = \frac{\sum_{(\boldsymbol{x},Y)\in\mathcal{S}} [\![f_j(\boldsymbol{x}) > t_j(\boldsymbol{x})]\!] \cdot [\![y_j \in Y]\!]}{\sum_{(\boldsymbol{x},Y)\in\mathcal{S}} [\![f_j(\boldsymbol{x}) > t_j(\boldsymbol{x})]\!]}$$

$$R_j = \frac{\sum_{(\boldsymbol{x},Y)\in\mathcal{S}} [\![f_j(\boldsymbol{x}) > t_j(\boldsymbol{x})]\!] \cdot [\![y_j \in Y]\!]}{\sum_{(\boldsymbol{x},Y)\in\mathcal{S}} [\![y_j \in Y]\!]}.$$

2) Macro-averaging AUC ($\text{AUC}_{\text{macro}}$)

$$\text{AUC}_{\text{macro}} = \frac{1}{q}\sum_{j=1}^{q} \text{AUC}_j, \text{ where}$$

$$\text{AUC}_j = \frac{\left|\left(\boldsymbol{x}', \boldsymbol{x}''\right) \mid f_j\left(\boldsymbol{x}'\right) > f_j\left(\boldsymbol{x}''\right),\ \left(\boldsymbol{x}', \boldsymbol{x}''\right) \in \mathcal{Z}_j^+ \times \mathcal{Z}_j^-\right|}{\left|\mathcal{Z}_j^+\right| \cdot \left|\mathcal{Z}_j^-\right|}$$

TABLE III

PERFORMANCE OF EACH COMPARING ALGORITHM (MEAN±STD. DEVIATION) IN TERMS OF *Macro-Averaging F-Measure* (F_{MACRO}; THE LARGER THE VALUE OF F_{MACRO}, THE BETTER THE PERFORMANCE). FURTHERMORE, ON EACH DATASET, THE PERFORMANCE OF THE TOP-RANKED ALGORITHM AND THE RUNNER-UP ALGORITHM ARE MARKED WITH ● AND ∗, RESPECTIVELY. FOR EACH COMPARING ALGORITHM, ITS AVERAGE RANK ACROSS ALL DATASETS IS ALSO SUMMARIZED AT THE BOTTOM LINE

Data set	Algorithm								
	COCOA	THRSEL	IRUS	SMOTE-EN	RML	BR	CLR	ECC	RAKEL
CAL500	.210±.007	.252±.006∗	.277±.004●	.235±.003	.209±.008	.169±.007	.081±.007	.092±.004	.193±.003
Emotions	.666±.019●	.560±.021	.622±.014	.575±.023	.645±.016∗	.550±.020	.595±.017	.638±.020	.613±.018
Birds	.443±.031●	.311±.022	.269±.012	.389±.032∗	.325±.022	.332±.025	.299±.028	.253±.028	.338±.035
Medical	.759±.013●	.733±.014∗	.537±.069	.700±.017	.707±.015	.718±.014	.724±.017	.733±.017∗	.672±.014
LLOG	.082±.010	.096±.004∗	.124±.003●	.095±.013	.095±.019	.031±.005	.024±.005	.022±.003	.023±.005
Enron	.342±.010●	.291±.006	.293±.005	.266±.009	.307±.028∗	.246±.011	.244±.007	.268±.009	.267±.012
Image	.639±.019●	.524±.011	.573±.004	.545±.006	.512±.013	.523±.011	.545±.011	.615±.017∗	.613±.013
Scene	.728±.011●	.626±.012	.632±.006	.623±.006	.684±.013	.626±.012	.631±.013	.716±.005∗	.686±.008
Yeast	.461±.011∗	.427±.008	.426±.008	.436±.005	.471±.014●	.409±.006	.413±.010	.389±.006	.420±.005
Slashdot	.374±.007●	.335±.015	.257±.011	.366±.007∗	.343±.029	.291±.018	.290±.018	.304±.015	.296±.015
Corel5k	.196±.004∗	.146±.009	.105±.003	.125±.004	.215±.009●	.089±.004	.049±.004	.054±.004	.084±.005
Rcv1-s1	.364±.007∗	.292±.006	.252±.004	.313±.004	.387±.020●	.285±.007	.227±.007	.192±.003	.272±.007
Rcv1-s2	.342±.008∗	.275±.006	.234±.005	.305±.004	.363±.029●	.272±.005	.226±.006	.173±.004	.263±.005
Rcv1-s3	.339±.008∗	.275±.010	.225±.002	.302±.006	.371±.006●	.271±.011	.211±.008	.163±.005	.257±.006
Bibtex	.318±.011∗	.303±.011	.253±.003	.283±.005	.326±.010●	.263±.008	.265±.009	.212±.009	.252±.012
Eurlex-sm	.703±.004●	.581±.006	.360±.003	.552±.003	.605±.006	.580±.006	.599±.006	.608±.007	.632±.008∗
Tmc2007	.668±.004●	.615±.003	.455±.002	.566±.002	.568±.039	.607±.002	.623±.003	.643±.003∗	.643±.004∗
Mediamill	.455±.005●	.346±.003	.278±.001	.338±.001	.268±.019	.318±.003	.268±.004	.260±.001	.378±.002∗
Average Rank	1.72	4.39	5.89	4.78	3.56	6.36	6.83	6.28	5.19

$$\mathcal{Z}_j^+ = \{\boldsymbol{x} \mid (\boldsymbol{x}, Y) \in \mathcal{S},\ y_j \in Y\}$$
$$\mathcal{Z}_j^- = \{\boldsymbol{x} \mid (\boldsymbol{x}, Y) \in \mathcal{S},\ y_j \notin Y\}.$$

Furthermore, two widely used canonical multi-label evaluation metrics, *ranking loss* and *average precision* [21], [64], are also employed for performance evaluation.[5]

1) *Ranking Loss (RL):*

$$RL = \frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x},Y)\in\mathcal{S}} \frac{1}{|Y||\bar{Y}|} \big|\big\{(y_{j_1}, y_{j_2}) \mid f_{j_1}(\boldsymbol{x}) \le f_{j_2}(\boldsymbol{x})\ (y_{j_1}, y_{j_2}) \in Y \times \bar{Y}\big\}\big|$$

where $\bar{Y} = \mathcal{Y} \setminus Y$ is the complementary set of Yin $\mathcal{Y}$.

2) *Average Precision (AP):*

$$AP = \frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x},Y)\in\mathcal{S}} \frac{1}{|Y|} \times \sum_{y_{j_1}\in Y} \frac{\big|\{y_{j_2} \mid \text{rank}(\boldsymbol{x}, y_{j_2}) \le \text{rank}(\boldsymbol{x}, y_{j_1}),\ y_{j_2} \in Y\big|}{\text{rank}(\boldsymbol{x}, y_{j_1})}$$

where $\text{rank}(\boldsymbol{x}, y_j)$ returns the rank of y_j when all labels in $\mathcal{Y}$ are sorted in descending order based on $\{f_j(\cdot) \mid 1 \le j \le q\}$.

B. Experimental Results

Tables III–VI report the detailed experimental results of the comparing algorithms in terms of each evaluation metric.[6] For each dataset, 50% examples are randomly sampled without replacement to form the training set, and the remaining 50% examples are used to form the test set. The random train/test splits are repeated for ten times and the mean metric value as well as the standard deviation are recorded. In each table, the performance of the top-ranked algorithm and the runner-up algorithm is marked with ● and ∗, respectively, and the average rank across all datasets are also summarized for each comparing algorithm.

To systematically analyze the relative performance among the comparing algorithms, the widely used *Friedman test* [14] is employed which serves as a favorable statistical test for comparisons among *multiple algorithms* over *a number of datasets*. Given k comparing algorithms and T datasets, let r_i^j denote the rank of the jth algorithm on the ith dataset where mean ranks are shared in case of ties. Furthermore, let $R_j = (1/T)\sum_{i=1}^{T} r_i^j$ denote the average rank for the jth algorithm. Then, under the null hypothesis of all algorithms having "equal" performance, the following Friedman statistic F_F will be distributed according to the F-distribution with $k-1$ numerator degrees of freedom and $(k-1)(T-1)$ denominator degrees of freedom:

$$F_F = \frac{(T-1)\chi_F^2}{T(k-1) - \chi_F^2}$$

[5] For brevity, experimental results in terms of other multi-label evaluation metrics are not reported in this article while similar observations can be obtained as well.

[6] As the RML approach [43] does not yield real-valued outputs on each class label, its performance is only evaluated in terms of macro-averaging F-measure with categorical classification results.

TABLE IV
PERFORMANCE OF EACH COMPARING ALGORITHM (MEAN±STD. DEVIATION) IN TERMS OF *Macro-Averaging AUC* ($\text{AUC}_{\text{MACRO}}$; THE LARGER THE VALUE OF $\text{AUC}_{\text{MACRO}}$, THE BETTER THE PERFORMANCE). FURTHERMORE, ON EACH DATASET, THE PERFORMANCE OF THE TOP-RANKED ALGORITHM AND THE RUNNER-UP ALGORITHM ARE MARKED WITH ● AND ∗, RESPECTIVELY. FOR EACH COMPARING ALGORITHM, ITS AVERAGE RANK ACROSS ALL DATASETS IS ALSO SUMMARIZED AT THE BOTTOM LINE

Data set	Algorithm							
	COCOA	THRSEL	IRUS	SMOTE-EN	BR	CLR	ECC	RAKEL
CAL500	.558±.005∗	.509±.004	.545±.004	.512±.003	.509±.004	.561±.004●	.557±.008	.528±.005
Emotions	.844±.010∗	.687±.013	.802±.008	.698±.013	.687±.013	.796±.010	.850±.009●	.797±.015
Birds	.855±.015●	.673±.029	.843±.009	.692±.020	.673±.029	.737±.018	.850±.018∗	.737±.024
Medical	.964±.007●	.869±.014	.955±.009∗	.873±.029	.869±.014	.955±.007∗	.952±.010	.856±.010
LLOG	.663±.005	.518±.005	.676±.010●	.561±.013	.518±.005	.612±.009	.673±.012∗	.514±.004
Enron	.752±.006●	.597±.006	.738±.005	.619±.011	.597±.006	.720±.004	.750±.004∗	.650±.006
Image	.864±.008∗	.681±.011	.823±.007	.698±.011	.681±.011	.799±.008	.867±.008●	.813±.009
Scene	.943±.003∗	.761±.015	.920±.004	.777±.012	.761±.015	.894±.005	.944±.003●	.892±.004
Yeast	.711±.006●	.576±.007	.658±.006	.582±.006	.576±.007	.650±.004	.705±.006∗	.641±.004
Slashdot	.774±.005●	.632±.009	.753±.010	.714±.009	.632±.009	.742±.009	.765±.008∗	.638±.003
Corel5k	.718±.004	.559±.006	.687±.008	.596±.005	.559±.006	.740±.002●	.723±.005∗	.552±.002
Rcv1-s1	.889±.003∗	.643±.012	.882±.003	.626±.007	.643±.012	.891±.003●	.881±.003	.728±.003
Rcv1-s2	.882±.002●	.640±.008	.880±.003	.622±.007	.640±.008	.882±.002●	.874±.002	.721±.003
Rcv1-s3	.880±.002●	.633±.012	.872±.003	.628±.006	.633±.012	.877±.002∗	.872±.003	.718±.004
Bibtex	.877±.003	.673±.009	.894±.003●	.706±.008	.673±.009	.881±.004∗	.873±.003	.696±.007
Eurlex-sm	.957±.002●	.778±.006	.952±.001∗	.796±.006	.778±.006	.944±.001	.951±.002	.872±.005
Tmc2007	.931±.001●	.784±.005	.916±.001	.793±.003	.784±.005	.906±.001	.928±.001∗	.859±.002
Mediamill	.844±.001●	.650±.003	.818±.001	.670±.003	.650±.003	.805±.001	.840±.001∗	.737±.001
Average rank	1.64	7.17	2.83	6.05	7.17	3.19	2.47	5.47

TABLE V
PERFORMANCE OF EACH COMPARING ALGORITHM (MEAN±STD. DEVIATION) IN TERMS OF *Ranking Loss* (RL; THE SMALLER THE VALUE OF RL, THE BETTER THE PERFORMANCE). FURTHERMORE, ON EACH DATASET, THE PERFORMANCE OF THE TOP-RANKED ALGORITHM AND THE RUNNER-UP ALGORITHM ARE MARKED WITH ● AND ∗, RESPECTIVELY. FOR EACH COMPARING ALGORITHM, ITS AVERAGE RANK ACROSS ALL DATASETS IS ALSO SUMMARIZED AT THE BOTTOM LINE

Data set	Algorithm							
	COCOA	THRSEL	IRUS	SMOTE-EN	BR	CLR	ECC	RAKEL
CAL500	.265±.003	.383±.008	.482±.008	.473±.009	.383±.008	.241±.002∗	.237±.004●	.340±.003
Emotions	.159±.014∗	.306±.019	.202±.011	.299±.013	.306±.019	.193±.011	.151±.011●	.200±.018
Birds	.098±.008∗	.167±.017	.123±.011	.163±.012	.167±.017	.110±.006	.095±.008●	.140±.011
Medical	.021±.003●	.057±.014	.030±.005	.072±.017	.057±.014	.023±.003	.022±.004∗	.087±.011
LLOG	.226±.008∗	.268±.012	.258±.008	.306±.013	.268±.012	.228±.006	.223±.005●	.357±.013
Enron	.116±.002∗	.231±.007	.252±.021	.249±.010	.231±.007	.121±.002	.114±.001●	.200±.007
Image	.149±.009∗	.312±.016	.182±.008	.289±.015	.312±.016	.199±.011	.147±.010●	.198±.008
Scene	.073±.004●	.248±.026	.089±.003	.222±.019	.248±.026	.111±.004	.073±.003●	.112±.005
Yeast	.186±.006∗	.348±.012	.439±.005	.399±.008	.348±.012	.204±.003	.182±.004●	.230±.004
Slashdot	.189±.004∗	.219±.008	.245±.020	.221±.005	.219±.008	.183±.005●	.189±.005∗	.332±.004
Corel5k	.201±.002	.257±.005	.362±.015	.343±.005	.257±.005	.186±.003●	.189±.003∗	.569±.006
Rcv1-s1	.078±.002	.287±.010	.104±.003	.301±.008	.287±.010	.077±.001∗	.074±.002●	.187±.003
Rcv1-s2	.081±.002	.269±.010	.108±.004	.277±.009	.269±.010	.079±.001●	.079±.002●	.194±.004
Rcv1-s3	.082±.002	.269±.012	.112±.002	.281±.006	.269±.012	.080±.001∗	.078±.002●	.195±.004
Bibtex	.059±.002	.128±.006	.049±.002●	.138±.006	.128±.006	.049±.002●	.053±.002	.150±.005
Eurlex-sm	.029±.001●	.150±.005	.036±.001	.141±.003	.150±.005	.031±.001	.030±.001∗	.087±.003
Tmc2007	.046±.001∗	.142±.002	.139±.001	.152±.003	.142±.002	.050±.001	.045±.001●	.100±.002
Mediamill	.074±.001●	.221±.003	.277±.002	.291±.002	.221±.003	.081±.001	.074±.001●	.141±.001
Average rank	2.25	6.17	5.03	6.89	6.17	2.61	1.44	5.39

where

$$\chi_F^2 = \frac{12T}{k(k+1)}\left[\sum_{j=1}^{n} R_j^2 - \frac{k(k+1)^2}{4}\right].$$

The Friedman statistics F_F and the corresponding critical value on each evaluation metric are summarized in Table VII. It is clear from Table VII that, at the 0.05 significance level, the null hypothesis of equal performance among the

TABLE VI
Performance of Each Comparing Algorithm (Mean±Std. Deviation) in Terms of *Average Precision* (AP; the Larger the Value of AP, the Better the Performance). Furthermore, on Each Dataset, the Performance of the Top-Ranked Algorithm and the Runner-Up Algorithm Are Marked With • and ∗, Respectively. For Each Comparing Algorithm, Its Average Rank Across All Datasets Is Also Summarized at the Bottom Line

Data set	Algorithm							
	Cocoa	ThrSel	Irus	Smote-En	Br	Clr	Ecc	Rakel
CAL500	.477±.005	.330±.006	.276±.007	.253±.005	.330±.006	.506±.004∗	.511±.005•	.399±.006
Emotions	.800±.011∗	.682±.015	.755±.012	.679±.013	.682±.015	.768±.011	.809±.010•	.765±.018
Birds	.673±.016∗	.521±.024	.595±.029	.538±.026	.521±.024	.619±.020	.680±.028•	.587±.019
Medical	.920±.007•	.874±.020	.883±.014	.845±.020	.874±.020	.912±.008	.920±.010•	.828±.019
LLOG	.346±.009∗	.306±.012	.307±.009	.278±.010	.306±.012	.342±.009	.353±.010•	.218±.013
Enron	.711±.007∗	.596±.008	.532±.021	.532±.006	.596±.008	.702±.006	.718±.007•	.652±.010
Image	.818±.009∗	.671±.013	.781±.007	.680±.015	.671±.013	.763±.010	.820±.010•	.774±.006
Scene	.868±.007∗	.707±.016	.844±.005	.710±.013	.707±.016	.809±.007	.870±.003•	.822±.006
Yeast	.762±.007∗	.595±.008	.543±.004	.535±.010	.595±.008	.739±.004	.766±.005•	.715±.004
Slashdot	.603±.006•	.565±.012	.504±.046	.571±.010	.565±.012	.591±.009	.598±.007∗	.484±.007
Corel5k	.396±.004∗	.343±.006	.189±.045	.244±.003	.343±.006	.386±.006	.405±.003•	.213±.006
Rcv1-s1	.601±.005∗	.428±.007	.555±.007	.403±.005	.428±.007	.597±.004	.626±.004•	.502±.005
Rcv1-s2	.611±.004∗	.458±.007	.568±.008	.437±.005	.458±.007	.611±.003∗	.630±.004•	.515±.006
Rcv1-s3	.608±.005∗	.463±.008	.571±.004	.438±.008	.463±.008	.607±.003	.635±.003•	.506±.006
Bibtex	.725±.005	.610±.012	.731±.005∗	.602±.008	.610±.012	.727±.007	.735±.006•	.594±.008
Eurlex-sm	.864±.003•	.682±.006	.829±.002	.669±.004	.682±.006	.838±.002	.864±.004•	.788±.005
Tmc2007	.859±.001∗	.757±.002	.694±.002	.728±.002	.757±.002	.846±.001	.864±.001•	.819±.002
Mediamill	.803±.001•	.597±.004	.494±.003	.413±.001	.597±.004	.778±.001	.801±.001∗	.736±.001
Average rank	2.03	5.94	5.25	7.19	5.94	3.14	1.17	5.28

TABLE VII
Summary of the Friedman Statistics F_F in Terms of F_{macro} (*Macro-Averaging F-Measure*), AUC$_{\text{macro}}$ (*Macro-Averaging AUC*), RL (*Ranking Loss*), and AP (*Average Precision*). The Critical Value w.r.t. k (# Comparing Algorithms) and T (# Datasets) at 0.05 Significance Level Is Also Given

Metric	F_F	k	T	critical value
F_{macro}	9.1053	9	18	1.9384
AUC_{macro}	76.5477	8	18	2.0868
RL	41.5142	8	18	2.0868
AP	48.9641	8	18	2.0868

comparing algorithms should be rejected for all evaluation metrics.

Thereafter, by treating Cocoa as the control algorithm, we employ *Holm's procedure* [14] as the *post-hoc* test to show whether Cocoa achieves significantly different performance against each of the other algorithms. Without loss of generality, we take the first comparing algorithm $\mathcal{A}_1$ as Cocoa. Among the other $k-1$ comparing algorithms $\mathcal{A}_j$ $(2 \leq j \leq k)$, we take $\mathcal{A}_j$ as the one which has the $(j-1)$th largest average rank over all datasets. Then, the test statistic for comparing $\mathcal{A}_1$ (i.e., Cocoa) and $\mathcal{A}_j$ corresponds to

$$z_j = \left(R_1 - R_j\right) \Bigg/ \sqrt{\frac{k(k+1)}{6T}} \qquad (2 \leq j \leq k). \tag{8}$$

Accordingly, let p_j denote the p-value of z_j under normal distribution. Given the significance level α, Holm's procedure works in a stepwise manner by checking whether the statistic p_j is below $\alpha/(k-j+1)$ in ascending order of j. Specifically, Holm's procedure terminates at j^* where j^* corresponds to the first j such that $p_j < \alpha/(k-j+1)$ does not hold.[7] Then, Cocoa is deemed to have significantly different performance against $\mathcal{A}_j$ with $j \in \{2, \ldots, j^*-1\}$.

Tables VIII and IX report the statistics of the *post-hoc* test based on Holm's procedure at 0.05 significance level, where Cocoa is treated as the control algorithm. Specifically, the following observations can be made based on the reported experimental results.

1) In terms of *macro-averaging F-measure* (F_{macro}, Table III), among the nine comparing algorithms, Cocoa achieves best and runner-up performance in 55.6% and 33.3% cases, respectively. As shown in Table VIII, it is impressive that Cocoa significantly outperforms all comparing algorithms. Note that although Cocoa is not tailored to optimize the macro-averaging F-measure as Rml does, its performance is rather competitive to Rml on this imbalance-specific metric.
2) In terms of *macro-averaging AUC* (AUC$_{\text{macro}}$, Table IV), among the eight comparing algorithms, Cocoa achieves best and runner-up performance in 55.6% and 27.8% cases, respectively. As shown in Table VIII, it is also noteworthy that Cocoa significantly outperforms Br, ThrSel, Smote-En, and Rakel. The statistically comparable performance of Clr and Ecc against Cocoa on AUC$_{\text{macro}}$ show their good ability in ranking positive (minority class) examples higher than

[7] If $p_j < \alpha/(k-j+1)$ holds for all j, j^* takes the value of $k+1$.

TABLE VIII
Comparison of Cocoa (Control Algorithm) Against Other Comparing Algorithms (With *Holm's Procedure* as the Post-Hoc Test at Significance Level $\alpha = 0.05$) in Terms of Imbalance-Specific Evaluation Metrics

macro-averaging F-measure (# comparing algorithms $k = 9$)				
j	algorithm	z_j	p_j	$\alpha/(k-j+1)$
2	Clr	-5.598	2.171e-8	0.006
3	Br	-5.083	3.718e-7	0.007
4	Ecc	-4.995	5.877e-7	0.008
5	Irus	-4.568	4.924e-6	0.010
6	Rakel	-3.801	1.440e-4	0.013
7	Smote-En	-3.352	8.021e-4	0.017
8	ThrSel	-2.925	3.446e-3	0.025
9	Rml	-2.016	4.384e-2	0.050

macro-averaging AUC (# comparing algorithms $k = 8$)				
j	algorithm	z_j	p_j	$\alpha/(k-j+1)$
2	Br	-6.773	1.263e-11	0.007
3	ThrSel	-6.773	1.263e-11	0.008
4	Smote-En	-5.401	6.622e-8	0.010
5	Rakel	-4.691	2.722e-6	0.013
6	Clr	-1.898	5.765e-2	0.017
7	Irus	-1.457	1.450e-1	0.025
8	Ecc	-1.017	3.094e-1	0.050

TABLE IX
Comparison of Cocoa (Control Algorithm) Against Other Comparing Algorithms (With *Holm's Procedure* as the *Post-Hoc* Test at Significance Level $\alpha = 0.05$) in Terms of Canonical Multi-label Evaluation Metrics

ranking loss (# comparing algorithms $k = 8$)				
j	algorithm	z_j	p_j	$\alpha/(k-j+1)$
2	Smote-En	-5.683	1.325e-8	0.007
3	Br	-4.801	1.579e-6	0.008
4	ThrSel	-4.801	1.579e-6	0.010
5	Rakel	-3.846	1.202e-4	0.013
6	Irus	-3.405	6.622e-4	0.017
7	Clr	-0.441	6.593e-1	0.025
8	Ecc	0.992	1.000e0	0.050

average precision (# comparing algorithms $k = 8$)				
j	algorithm	z_j	p_j	$\alpha/(k-j+1)$
2	Smote-En	-5.160	2.621e-10	0.007
3	Br	-3.910	1.678e-6	0.008
4	ThrSel	-3.910	1.678e-6	0.010
5	Rakel	-3.250	6.879e-5	0.013
6	Irus	-3.220	8.024e-5	0.017
7	Clr	-1.110	1.740e-1	0.025
8	Ecc	0.860	1.000e0	0.050

negative (majority class) examples, while the inferior performance on F_{macro} is due to their less effective bipartitioning procedure w.r.t. each class label.

3) In terms of canonical multi-label evaluation metrics, among the eight comparing algorithms, Cocoa achieves runner-up or better performance in 72.2% cases on *ranking loss* (Table V) and 88.9% cases on *average precision* (Table VI). As shown in Table IX, Cocoa achieves comparable performance to Clr and Ecc and significantly outperforms the other comparing algorithms on both *ranking loss* and *average precision*. These observations show that Cocoa not only achieves promising results in terms of imbalance-specific metrics emphasizing generalization performance on minority class but also achieves competitive results in terms of canonical multi-label evaluation metrics assuming equal importance of minority and majority classes.

4) Cocoa significantly outperforms ThrSel, Smote-En, Br, and Rakel in terms of all evaluation metrics. It is also interesting to notice that the simple strategy of combining binary decomposition with threshold calibration (i.e., ThrSel) can lead to relatively good performance (third best) in terms of *macro-averaging F-measure*, while its performance degenerates to that of Br in terms of *macro-averaging AUC* where threshold calibration does not count.

5) In terms of imbalance-specific evaluation metrics (Tables III and IV), the performance advantage of Cocoa is more pronounced on datasets with large number of examples, such as `Eurlex-sm`, `tmc2007`, and `Mediamill`. Furthermore, Cocoa tends to achieve good performance on datasets with small number of class labels, such as `Emotions`, `Birds`, `Medical`, `Image`, `Scene`, `Yeast`, and `Slashdot`.

6) In terms of canonical multi-label evaluation metrics (Tables V and VI), the performance of Cocoa is inferior on datasets with large number of class labels such as `CAL500`. It is worth noting that on the pairwise evaluation metric *ranking loss*, Cocoa barely achieves best or runner-up performance on datasets with large average imbalance ratio, such as `Rcv1-s1`, `Rcv1-s2`, `Rcv1-s3`, and `Bibtext`. These results indicate that the cross-coupling strategy employed by Cocoa may bring benefits to class-imbalance classification by compromising its ranking performance between relevant and irrelevant labels.

C. Further Analysis

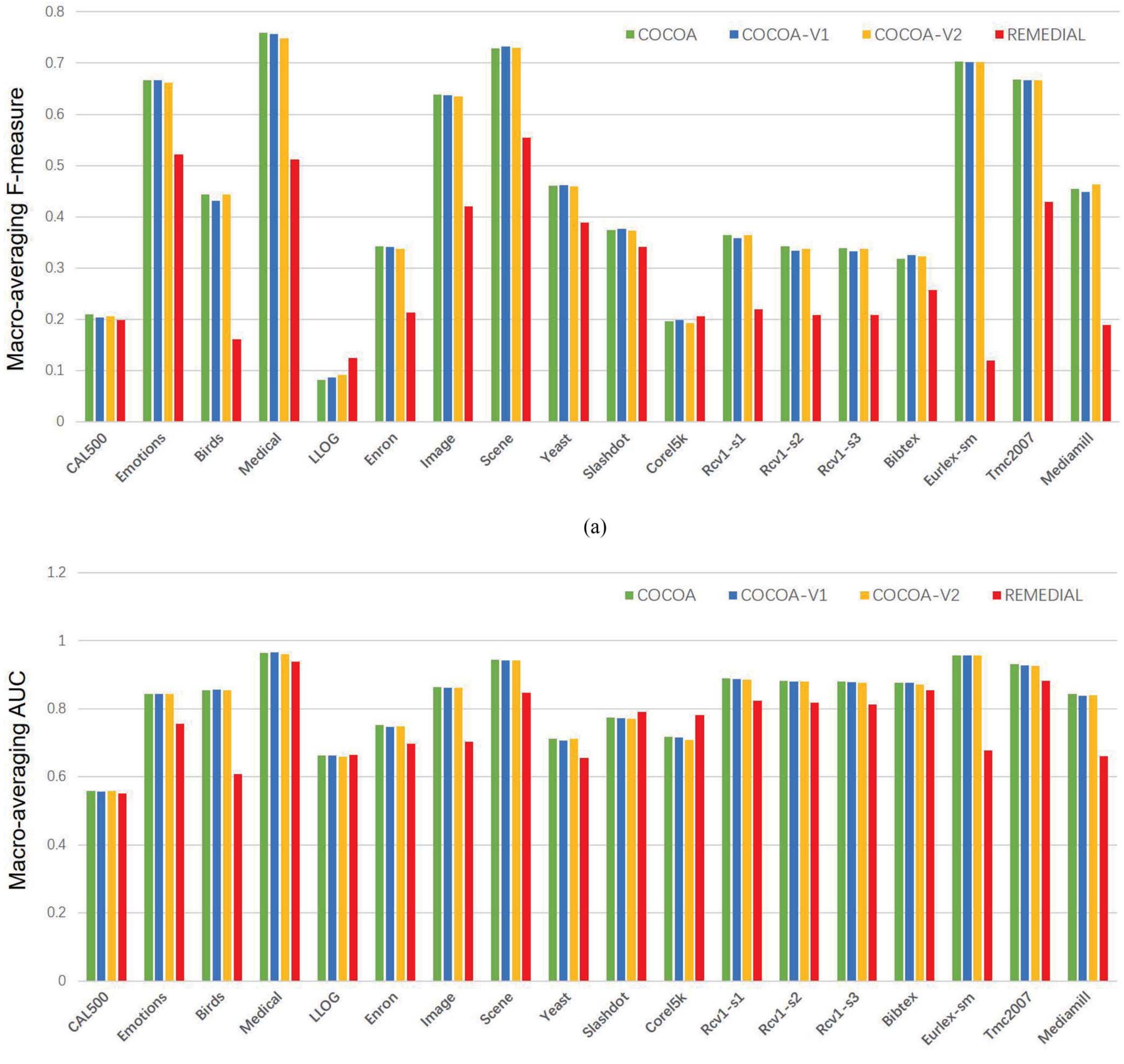
In this section, the following comparing algorithms are further considered to analyze specific properties of Cocoa.

1) As shown in (5), Cocoa assumes equal weight for each coupling label $y_k \in \mathcal{I}_K$ in aggregating the predictive confidence of each multiclass imbalance learner $g_{jk}(\cdot)$. As an alternative, we can rewrite (5) in the following way by taking account of the class recognition in predictive confidence aggregation:

$$f_j(\boldsymbol{x}) = \sum_{y_k \in \mathcal{I}_K} w_k \cdot g_{jk}(+2 \mid \boldsymbol{x}) \quad \text{where}$$
$$\hat{w}_k = \sum_{i=1}^{N} [\![y_k \in Y_i]\!], \quad w_k = \frac{\hat{w}_k}{\sum_{y_k \in \mathcal{I}_K} \hat{w}_k}. \tag{9}$$

By replacing (5) with the above definition and keeping the other algorithmic components of Cocoa unchanged, the resulting variant is called Cocoa-V1.

2) As shown in step 3 of Table I, for each class label y_j, the corresponding subset of coupling labels $\mathcal{I}_K$ is generated in a random manner. For each class label $y_k \in \mathcal{Y} \setminus \{y_j\}$, let $C_{jk} = \sum_{i=1}^{N} [\![\phi(Y_i, y_j) == \phi(Y_i, y_k)]\!]$ be the count of



(a)

(b)

Fig. 1. Performance of Cocoa and the comparing algorithms Cocoa-V1, Cocoa-V2, and Remedial in terms of imbalance-specific evaluation metrics. (a) *Macro-averaging F-measure* (F_{macro}; the larger the value of F_{macro}, the better the performance). (b) *Macro-averaging AUC* ($\text{AUC}_{\text{macro}}$; the larger the value of $\text{AUC}_{\text{macro}}$, the better the performance).

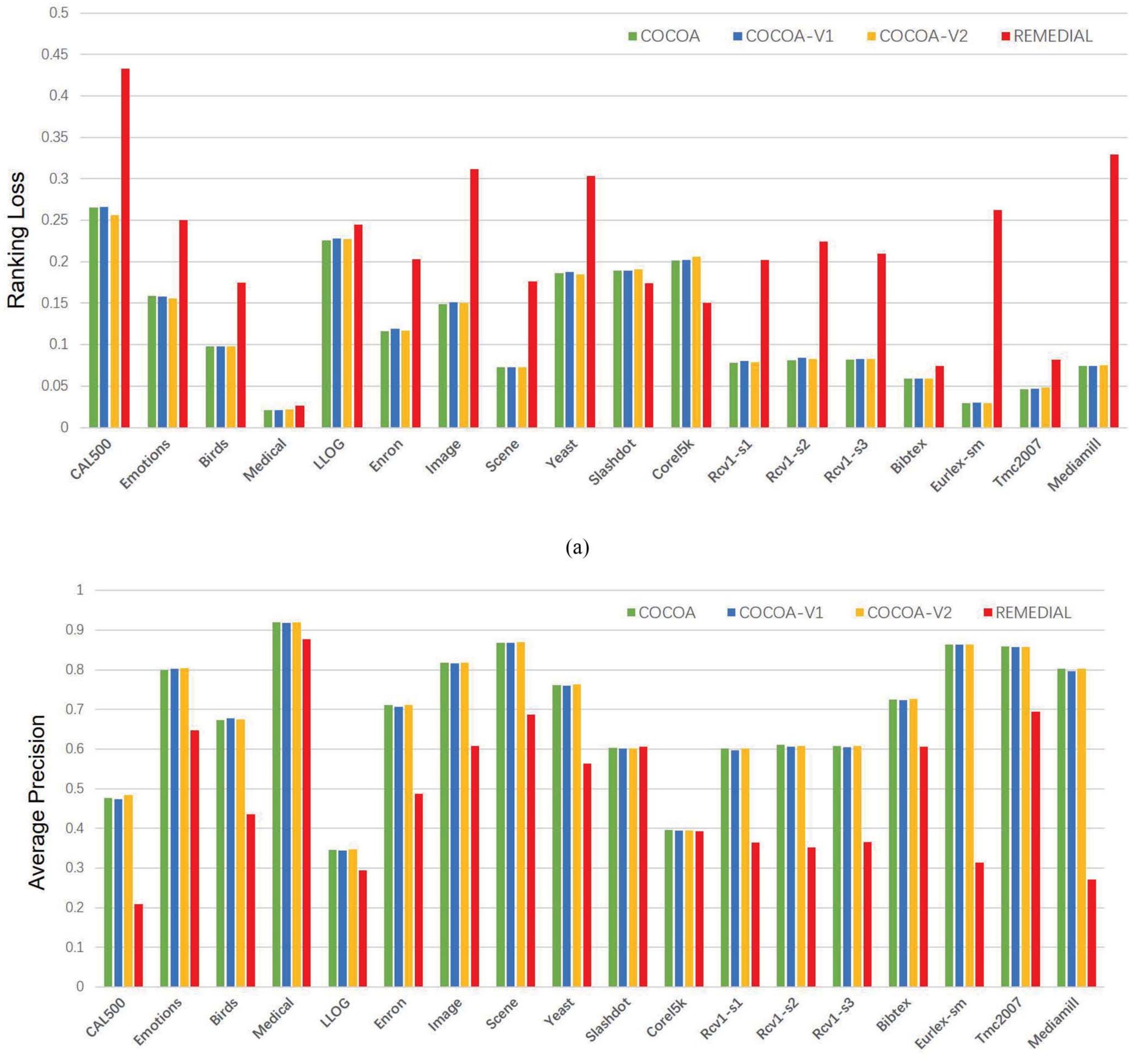
identical joint assignment which indicates the degree of correlation between y_j and y_k. As an alternative, we can form $\mathcal{I}_K$ by choosing K labels from $\mathcal{Y} \setminus \{y_j\}$ which have the highest degree of correlation with y_j. By keeping the other algorithmic components of Cocoa unchanged, the resulting variant is called Cocoa-V2.

3) In Section IV-B, the performance of Cocoa is compared against several well-established class-imbalance multi-label learning algorithms based on undersampling/oversampling or F-measure maximization. It is worth noting that Cocoa relies on the key strategy of cross-coupling which considers the joint assignment of a pair of class labels. In light of this, we further employ one recently proposed class-imbalance multi-label learning approach called Remedial [10] for comparative studies, which works in a similar manner by exploiting label concurrence between minority and majority labels.

Figs. 1 and 2 illustrate the performance of Cocoa and the comparing algorithms Cocoa-V1, Cocoa-V2, and Remedial in terms of imbalance-specific and canonical multi-label evaluation metrics, respectively. Furthermore, Table X summarizes the win/tie/loss counts of Cocoa against Cocoa-V1, Cocoa-V2, and Remedial over all the benchmark datasets based on the pairwise t-test at 0.05 significance level.

Based on the reported results, we can observe the following.

1) In most cases, Cocoa achieves significantly better or at least comparable performance to its variant Cocoa-V1.



(a)

(b)

Fig. 2. Performance of COCOA and the comparing algorithms COCOA-V1, COCOA-V2, and REMEDIAL in terms of canonical multi-label evaluation metrics. (a) *Ranking loss* (RL; the smaller the value of RL, the better the performance). (b) *Average precision* (AP; the larger the value of AP, the better the performance).

These results indicate that assuming equal weight for each coupling label in (5) serves as a good practice for COCOA.

2) In most cases, COCOA achieves significantly better or at least comparable performance to its variant COCOA-V2. These results indicate that the random coupling strategy employed by COCOA is effective in generating a predictive model with good generalization performance.
3) In most cases, COCOA achieves significantly better performance than REMEDIAL.

TABLE X
PAIRWISE t-TEST BETWEEN COCOA AND THE COMPARING ALGORITHMS COCOA-V1, COCOA-V2, AND REMEDIAL AT 0.05 SIGNIFICANCE LEVEL. THE WIN/TIE/LOSS COUNTS OVER 18 BENCHMARK DATASETS ARE RECORDED IN TERMS OF EACH EVALUATION METRIC

	COCOA **against**		
Metric	COCOA-V1	COCOA-V2	REMEDIAL
F_{macro}	5/10/3	3/13/2	16/0/2
AUC_{macro}	5/12/1	8/10/0	14/2/2
RL	4/14/0	4/12/0	15/1/2
AP	7/10/1	1/16/1	16/2/0
In Total	21/46/5	16/51/5	61/5/6

V. CONCLUSION

In this article, the intrinsic property of class-imbalance for learning from multi-label data is investigated. Specifically, a simple yet effective class-imbalance multi-label learning approach called COCOA is proposed which considers the exploitation of label correlations via cross-coupling and the exploration of class-imbalance via undersampling. Extensive experiments over a total of 18 benchmark datasets as well as up to eight comparing algorithms clearly validate the effectiveness of the proposed approach in solving class-imbalance multi-label learning problems.

In the future, it is interesting to investigate other strategies for simultaneous label correlations exploitation and class-imbalance exploration. Furthermore, in addition to the label-wise class-imbalance issue of skewness between positive examples and negative examples for each label, it is also important to investigate the instance-wise class-imbalance issue of skewness between relevant labels and irrelevant labels for each instance, especially for large-scale multi-label learning with huge output space [2], [27], [58].

Acknowledgment

The authors wish to thank the Associate Editor and anonymous reviewers for their insightful comments and suggestions and the Big Data Center of Southeast University for providing the facility support on the numerical calculations in this article.

References

[1] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[2] R. Babbar and B. Schölkopf, "Data scarcity, robustness and extreme multi-label classification," *Mach. Learn.*, vol. 108, pp. 1329–1351, Mar. 2019.

[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.

[4] A. Braytee, W. Liu, A. Anaissi, and P. J. Kennedy, "Correlated multi-label classification with incomplete label space and class imbalance," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 5, p. 56, 2019.

[5] F. Briggs *et al.*, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *J. Acoust. Soc. Amer.*, vol. 131, no. 6, pp. 4640–4650, 2012.

[6] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for weakly-supervised multi-label image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 121–135, Jan. 2015.

[7] N. Cesa-Bianchi, M. Re, and G. Valentini, "Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference," *Mach. Learn.*, vol. 88, no. 1, pp. 209–241, 2012.

[8] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Addressing imbalance in multi-label classification: Measures and random resampling algorithms," *Neurocomputing*, vol. 163, pp. 3–16, Sep. 2015.

[9] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "MLSMOTE: Approaching imbalanced multi-label learning through synthetic instance generation," *Knowl. Based Syst.*, vol. 89, pp. 385–397, Nov. 2015.

[10] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Dealing with difficult minority labels in imbalanced mutilabel data sets," *Neurocomputing*, vols. 326–327, pp. 39–53, Jan. 2019.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[12] Z. A. Daniels and D. N. Metaxas, "Addressing imbalance in multi-label classification using structured Hellinger forests," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 1826–1832.

[13] K. Dembczyński, A. Jachnik, W. Kotłowski, W. Waegeman, and E. Hüllermeier, "Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 1130–1138.

[14] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[15] X. Ding, B. Li, W. Xiong, W. Guo, W. Hu, and B. Wang, "Multi-instance multi-label learning combining hierarchical context and its application to image annotation," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1616–1627, Aug. 2016.

[16] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA, USA: MIT Press, 2002, pp. 681–687.

[17] R.-E. Fan and C.-J. Lin, "A study on threshold selection for multi-label classification," Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, Rep., 2007.

[18] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Berlin, Germany: Springer, 2018.

[19] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multi-label classification via calibrated label ranking," *Mach. Learn.*, vol. 73, no. 2, pp. 133–153, 2008.

[20] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manag.*, Bremen, Germany, 2005, pp. 195–200.

[21] E. Gibaja and S. Ventura, "A tutorial on multi-label learning," *ACM Comput. Surveys*, vol. 47, no. 3, p. 52, 2015.

[22] A. F. Giraldo-Forero, A. F. Cardona-Escobar, and A. E. Castro-Ospina, "Multi-label learning by hyperparameters calibration for treating class imbalance," in *Lecture Notes in Artificial Intelligence 10870*. Berlin, Germany: Springer, 2018, pp. 327–337.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.

[24] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[25] S.-J. Huang, G.-X. Li, W.-Y. Huang, and S.-Y. Li, "Incremental multi-label learning with active queries," *J. Comput. Sci. Technol.*, vol. 35, no. 2, pp. 234–246, 2020.

[26] W. Indyk, T. Kajdanowicz, and P. Kazienko, "Relational large scale multi-label classification method for video categorization," *Multimedia Tools Appl.*, vol. 65, no. 1, pp. 63–74, 2013.

[27] H. Jain, V. Balasubramanian, B. Chunduri, and M. Varma, "Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches," in *Proc. 12th ACM Int. Conf. Web Search Data Min.*, Melbourne, VIC, Australia, 2019, pp. 528–536.

[28] S. Ji, L. Tang, S. Yu, and J. Ye, "A shared-subspace learning framework for multi-label classification," *ACM Trans. Knowl. Discov. Data*, vol. 4, no. 2, p. 8, 2010.

[29] Y. Li, B. Wu, Y. Zhao, H. Yao, and Q. Ji, "Handling missing labels and class imbalance challenges simultaneously for facial action unit recognition," *Multimedia Tools Appl.*, vol. 78, pp. 20309–20332, Feb. 2019.

[30] P. Lim, C. K. Goh, and K. C. Tan, "Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2850–2861, Sep. 2017.

[31] W. Lin and D. Xu, "Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types," *Bioinformatics*, vol. 32, no. 34, pp. 3745–3752, 2016.

[32] B. Liu and G. Tsoumakas, "Synthetic oversampling of multi-label data based on local label distribution," in *Lecture Notes in Artificial Intelligence 11907*. Berlin, Germany: Springer, 2020, pp. 180–193.

[33] J. Liu, W. C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proc. 40th Int. ACM SIGIR Conf. Res Develop. Inf. Retrieval*, Tokyo, Japan, 2017, pp. 115–124.

[34] W. Liu and I. Tsang, "On the optimality of classifier chain for multi-label classification," in *Advances in Neural Information Processing Systems 28*. Cambridge, MA, USA: MIT Press, 2015, pp. 712–720.

[35] X.-Y. Liu, Q.-Q. Li, and Z.-H. Zhou, "Learning imbalanced multi-class data with optimal dichotomy weights," in *Proc. 13th IEEE Int. Conf. Data Min.*, Dallas, TX, USA, 2013, pp. 478–487.

[36] X.-Y. Liu, S.-T. Wang, and M.-L. Zhang, "Transfer synthetic over-sampling for class-imbalance learning with limited minority class data," *Front. Comput. Sci.*, vol. 13, no. 5, pp. 996–1009, 2019.

[37] Y. Liu, Y. Liu, C. Wang, X. Wang, P. Zhou, G. Yu, and K. C. C. Chan, "What strikes the strings of your heart?—Multi-label dimensionality reduction for music emotion analysis via brain imaging," *IEEE Trans. Auton. Mental Develop.*, vol. 7, no. 3, pp. 176–188, Sep. 2015.

[38] H.-Y. Lo, S.-D. Lin, and H.-M. Wang, "Generalized k-labelsets ensemble for multi-label and cost-sensitive classification," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1679–1691, Jul. 2014.

[39] Y. Ma, C. Cui, J. Yu, J. Guo, G. Yang, and Y. Yin, "Multi-task MIML learning for pre-course student performance prediction," *Front. Comput. Sci.*, vol. 14, no. 5, 2020, Art. no. 145313.

[40] W. W. Y. Ng, J. Hu, D. S. Yeung, S. Yin, and F. Roli, "Diversified sensitivity-based undersampling for imbalance classification problems," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2402–2412, Nov. 2015.

[41] X. Pan, Y.-X. Fan, J. Jia, and H.-B. Shen, "Identifying RNA-binding proteins using multi-label deep learning," *Sci. China Inf. Sci.*, vol. 62, no. 1, 2019, Art. no. 019103.

[42] R. M. Pereira, Y. M. G. Costa, and C. N. Silla, Jr., "MLTL: A multi-label approach for the Tomek link undersampling algorithm," *Neurocomputing*, vol. 383, pp. 95–105, Mar. 2020.

[43] J. Petterson and T. Caetano, "Reverse multi-label learning," in *Advances in Neural Information Processing Systems 23*. Cambridge, MA, USA: MIT Press, 2010, pp. 1912–1920.

[44] I. Pillai, G. Fumera, and F. Roli, "Threshold optimisation for multi-label classifiers," *Pattern Recognit.*, vol. 46, no. 7, pp. 2055–2065, 2013.

[45] I. Pillai, G. Fumera, and F. Roli, "Designing multi-label classifiers that maximize F measures: State of the art," *Pattern Recognit.*, vol. 61, pp. 394–404, Jan. 2017.

[46] J. R. Quevedo, O. Luaces, and A. Bahamonde, "Multi-label classifiers with a probabilistic thresholding strategy," *Pattern Recognit.*, vol. 45, no. 2, pp. 876–883, 2012.

[47] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, 2011.

[48] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Mach. Learn.*, vol. 88, nos. 1–2, pp. 157–208, 2012.

[49] P. Sadhukhan and S. Palit, "Reverse-nearest neighborhood based oversampling for imbalanced, multi-label datasets," *Pattern Recognit. Lett.*, vol. 125, pp. 813–820, Jul. 2019.

[50] E. Spyromitros-Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas, "Dealing with concept drift and class imbalance in multi-label stream classification," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, 2011, pp. 1583–1588.

[51] K. W. Sun and C. H. Lee, "Addressing class-imbalance in multi-label learning via two-stage multi-label hypernetwork," *Neurocomputing*, vol. 266, pp. 375–389, Nov. 2017.

[52] M. A. Tahir, J. Kittler, and F. Yan, "Inverse random under sampling for class imbalance problem and its application to multi-label classification," *Pattern Recognit.*, vol. 45, no. 10, pp. 3738–3750, 2012.

[53] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[54] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "MULAN: A java library for multi-label learning," *J. Mach. Learn. Res.*, vol. 12, pp. 2411–2414, Jul. 2011.

[55] P. Vateekul, M. Kubat, and K. Sarinnapakorn, "Hierarchical multi-label classification with SVMs: A case study in gene function prediction," *Intell. Data Anal.*, vol. 18, no. 4, pp. 717–738, 2014.

[56] J. Wang, Y. Zhao, X. Wu, and X.-S. Hua, "A transductive multi-label learning approach for video concept detection," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2274–2286, 2011.

[57] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012.

[58] T. Wei, W.-W. Tu, and Y.-F. Li, "Learning for tail label data: A label-specific feature approach," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macau, China, 2019, pp. 3842–3848.

[59] S. Wen *et al.*, "Multi-label image classification via feature/label co-projection," *IEEE Trans. Cybern.*, early access, Feb. 6, 2020, doi: 10.1109/TSMC.2020.2967071.

[60] B. Wu, S. Lyu, and B. Ghanem, "Constrained submodular minimization for missing labels and class imbalance in multi-label learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 2229–2236.

[61] X. Yang, Q. Kuang, W. Zhang, and G. Zhang, "AMDO: An over-sampling technique for multi-class imbalanced problems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1672–1685, Sep. 2018.

[62] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: An overview," *Front. Comput. Sci.*, vol. 12, no. 2, pp. 191–202, 2018.

[63] M.-L. Zhang and Z.-H. Zhou, "ML-kNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[64] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[65] W. Zhang, J. Jiang, Y. Shao, and B. Cui, "Snapshot boosting: A fast ensemble framework for deep neural networks," *Sci. China Inf. Sci.*, vol. 63, no. 1, 2020, Art. no. 112102.

[66] Y. Zhang, Y. Wang, X.-Y. Liu, S. Mi, and M.-L. Zhang, "Large-scale multi-label classification using unknown streaming images," *Pattern Recognit.*, vol. 99, Mar. 2020, Art. no. 107100.

[67] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: Chapman Hall, 2012.

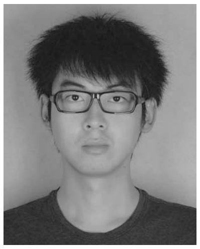
[68] Z.-H. Zhou and M.-L. Zhang, "Multi-label learning," in *Encyclopedia of Machine Learning and Data Mining, 2nd Edition*, C. Sammut and G. I. Webb, Eds. Berlin, Germany: Springer, 2017, pp. 875–881.

[69] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multi-label learning for image classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 450–461, Feb. 2016.

Min-Ling Zhang (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Nanjing University, Nanjing, China, in 2001, 2004, and 2007, respectively.

He is currently a Professor with the School of Computer Science and Engineering, Southeast University, Nanjing. His main research interests include machine learning and data mining.

Prof. Zhang has served as the General Co-Chair of ACML'18, the Program Co-Chair of PAKDD'19, CCF-ICAI'19, ACML'17, CCFAI'17, and PRICAI'16, and the Senior PC Member or Area Chair of AAAI from 2017 to 2020, IJCAI from 2017 to 2021, and ICDM from 2015 to 2020. He is also on the Editorial Board of *ACM Transactions on Intelligent Systems and Technology*, *Neural Networks*, *Science China Information Sciences*, and *Frontiers of Computer Science*. He is the Steering Committee Member of ACML and PAKDD, the Secretary-General of the CAAI Machine Learning Society, and the Standing Committee Member of the CCF Artificial Intelligence and Pattern Recognition Society. He is a Distinguished Member of CCF and CAAI and a Senior Member of ACM.



Yu-Kun Li received the B.Sc. and M.Sc. degrees in computer science from Southeast University, Nanjing, China, in 2012 and 2015, respectively.

He is currently a Research and Development Engineer with Baidu Inc., Beijing, China. His main research interests include machine learning and data mining, especially in learning from multi-label data.

Hao Yang received the B.Sc. degree in computer science from Southeast University, Nanjing, China, in 2019, where he is currently pursuing the master's degree.

His main research interests include machine learning and data mining, especially in learning from multi-label data.

Xu-Ying Liu received the B.Sc. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003, and the M.Sc. and Ph.D. degrees from Nanjing University, Nanjing, in 2006 and 2010, respectively.

She is an Assistant Professor with the School of Computer Science and Engineering, Southeast University, Nanjing. Her research interests mainly include machine learning and data mining, especially cost-sensitive learning and class-imbalance learning.