Towards Class-Imbalance Aware Multi-Label Learning

Min-Ling Zhang Yu-Kun Li Xu-Ying Liu
[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University),
Ministry of Education, China
{zhangml, liyk, liuxy}@seu.edu.cn

Abstract

In multi-label learning, each object is represented by a single instance while associated with *a set of* class labels. Due to the huge (exponential) number of possible label sets for prediction, existing approaches mainly focus on how to exploit label correlations to facilitate the learning process. Nevertheless, an intrinsic characteristic of learning from multi-label data, i.e. the widely-existing *class-imbalance* among labels, has not been well investigated. Generally, the number of positive training instances w.r.t. each class label is far less than its negative counterparts, which may lead to performance degradation for most multi-label learning techniques. In this paper, a new multi-label learning approach named *Cross-Coupling Aggregation* (COCOA) is proposed, which aims at leveraging the exploitation of label correlations as well as the exploration of class-imbalance. Briefly, to induce the predictive model on each class label, one binary-class imbalance learner corresponding to the current label and several multi-class imbalance learners coupling with other labels are aggregated for prediction. Extensive experiments clearly validate the effectiveness of the proposed approach, especially in terms of imbalance-specific evaluation metrics such as F-measure and area under the ROC curve.

1 Introduction

Under the multi-label learning setting, each example is represented by a single instance (feature vector) while associated with multiple class labels simultaneously [Tsoumakas *et al.*, 2010; Zhang and Zhou, 2014]. Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ denote the input space of d-dimensional feature vectors and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ denote the output space of q class labels. Given 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 feature vector and $Y_i \subseteq \mathcal{Y}$ is the set of labels associated with $\boldsymbol{x}_i$, the task is to learn a multi-label classifier $h : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ from $\mathcal{D}$ which maps from the space of feature vectors to the space of *label sets*.

The key challenge to learn from multi-label data lies in the huge number of possible label sets for prediction, which is *exponential* to the size of label space (i.e. 2^q). To take on this challenge, existing approaches mainly focus on exploiting *correlations* among class labels to facilitate the learning process [Tsoumakas *et al.*, 2010; Zhang and Zhou, 2014]. Based on the *order of correlations* being considered, existing approaches can be roughly grouped into three categories, i.e. first-order approaches which assume independence among class labels, second-order approaches which consider correlations between a pair of class labels, and high-order approaches which consider correlations among all the class labels or subsets of class labels.

On the other hand, an inherent property of learning from multi-label data, i.e. the *class-imbalance* among labels, has not been fully taken into consideration by most existing approaches. 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 *positive* and *negative* training examples w.r.t. y_j. Generally, the corresponding *imbalance ratio* $ImR_j = \max(|\mathcal{D}_j^+|, |\mathcal{D}_j^-|)/\min(|\mathcal{D}_j^+|, |\mathcal{D}_j^-|)$ would be high.[1] For instance, among the thirteen benchmark multi-label data sets used in this paper (Table 2), the `average` imbalance ratio across the label space (i.e. $\frac{1}{q}\sum_{j=1}^{q} ImR_j$) ranges from **2.1** to **17.9** (with nine of them greater than 5.0), and the `maximum` imbalance ratio across the label space (i.e. $\max_{1\leq j\leq q} ImR_j$) ranges from **3.0** to **50.0** (with eleven of them greater than 10.0).

Class-imbalance has long been regarded as one fundamental threat to compromise the performance of standard machine learning algorithms, which would also lead to performance degradation for most multi-label learning approaches [He and Garcia, 2009; Zhang and Zhou, 2014]. Therefore, a favorable practice towards designing multi-label learning algorithm should cautiously leverage the exploitation of label correlations as well as the exploration of class-imbalance. In this paper, a novel class-imbalance aware algorithm named COCOA, i.e. *CrOss-COupling Aggregation*, is proposed to learning from multi-label data. For each class label, COCOA builds one binary-class imbalance learner corresponding to the current label and also several multi-class imbalance

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

learners coupling with other labels. After that, the final prediction on each class label is obtained by aggregating the outputs yielded by the binary learner and the multi-class learners. Comparative studies across thirteen publicly-available multi-label data sets show that COCOA achieves highly competitive performance, especially in terms of appropriate evaluation metrics under class-imbalance scenario.

The rest of this paper is organized as follows. Section 2 presents technical details of the proposed COCOA approach. Section 3 discusses existing works related to COCOA. Section 4 reports the experimental results of comparative studies. Finally, Section 5 concludes.

2 The COCOA Approach

As shown in Section 1, the task of multi-label learning is to induce a multi-label classifier $h : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ from the training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq N\}$. This is equivalent to learn q real-valued functions $f_j : \mathcal{X} \rightarrow \mathbb{R}$ $(1 \leq j \leq q)$, each accompanied by a thresholding function $t_j : \mathcal{X} \rightarrow \mathbb{R}$. For any example $\boldsymbol{x} \in \mathcal{X}$, $f_j(\boldsymbol{x})$ returns the *confidence* of associating $\boldsymbol{x}$ with class label y_j, and the predicted label set is determined according to:

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

Let $\mathcal{D}_j$ denote the binary training set derived from $\mathcal{D}$ for the j-th class label y_j:

$$\mathcal{D}_j = \{(\boldsymbol{x}_i, \phi(Y_i, y_j)) \mid 1 \leq i \leq N\} \qquad (2)$$

$$\text{where } \phi(Y_i, y_j) = \begin{cases} +1, & \text{if } y_j \in Y_i \\ -1, & \text{otherwise} \end{cases}$$

Therefore, the derived binary training set consists of a subset of positive training examples ($\mathcal{D}_j^+$) and a subset of negative training examples ($\mathcal{D}_j^-$), i.e. $\mathcal{D}_j = \mathcal{D}_j^+ \bigcup \mathcal{D}_j^-$. To deal with the issue of having skewed distribution 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, i.e. $g_j \leftarrow \mathcal{B}(\mathcal{D}_j)$. Let $g_j(+1 \mid \boldsymbol{x})$ denote the predictive confidence that $\boldsymbol{x}$ should be regarded as a positive example for y_j, the real-valued function $f_j(\cdot)$ can then be instantiated as $f_j(\boldsymbol{x}) = g_j(+1 \mid \boldsymbol{x})$. In addition, the thresholding function $t_j(\cdot)$ can be simply set to some constant function such as $t_j(\boldsymbol{x}) = 0$.

In the above class-imbalance handling strategy, the predictive model $f_j(\cdot)$ for each class label y_j is built in an independent manner. To incorporate label correlations into the learning process, we choose to randomly couple another class label y_k $(k \neq j)$ with y_j as follows. Given the label pair (y_j, y_k), a multi-class training set $\mathcal{D}_{jk}$ can be derived from $\mathcal{D}$:

$$\mathcal{D}_{jk} = \{(\boldsymbol{x}_i, \psi(Y_i, y_j, y_k)) \mid 1 \leq i \leq N\} \qquad (3)$$

$$\text{where } \psi(Y_i, y_j, y_k) = \begin{cases} 0, & \text{if } y_j \notin Y_i \text{ and } y_k \notin Y_i \\ +1, & \text{if } y_j \notin Y_i \text{ and } y_k \in Y_i \\ +2, & \text{if } y_j \in Y_i \text{ and } y_k \notin Y_i \\ +3, & \text{if } y_j \in Y_i \text{ and } y_k \in Y_i \end{cases}$$

Table 1: The pseudo-code of COCOA.

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

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 label correlations can be exploited by making use of $\mathcal{D}_{jk}$ in the learning process, the issue of class-imbalance may be amplified by jointly considering y_j and y_k. Without loss of generality, suppose that positive examples $\mathcal{D}_j^+$ ($\mathcal{D}_k^+$) correspond to the *minority* class in the binary training set $\mathcal{D}_j$ ($\mathcal{D}_k$). Accordingly, the first class ($\psi(Y_i, y_k, y_k) = 0$) and the fourth class ($\psi(Y_i, y_k, y_k) = +3$) in $\mathcal{D}_{jk}$ would contain the largest and the smallest number of examples. Compared to the original 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 would roughly turn into $ImR_j \cdot ImR_k$ in four-class training set $\mathcal{D}_{jk}$. To deal with this potential problem, COCOA transforms the four-class data set $\mathcal{D}_{jk}$ into a tri-class data set $\mathcal{D}_{jk}^{\mathbf{tri}}$ by merging the third class and the fourth class (both with positive assignment for y_j):

$$\mathcal{D}_{jk}^{\mathbf{tri}} = \left\{\left(\boldsymbol{x}_i, \psi^{\mathbf{tri}}(Y_i, y_j, y_k)\right) \mid 1 \leq i \leq N\right\} \qquad (4)$$

$$\text{where } \psi^{\mathbf{tri}}(Y_i, y_j, y_k) = \begin{cases} 0, & \text{if } y_j \notin Y_i \text{ and } y_k \notin Y_i \\ +1, & \text{if } y_j \notin Y_i \text{ and } y_k \in Y_i \\ +2, & \text{if } y_j \in Y_i \end{cases}$$

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

Table 2: Characteristics of the benchmark multi-label data sets.

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
Medical	978	144	14	numeric	1.075	0.077	42	0.043	2.674	43.478	11.236
Enron	1702	50	24	nominal	3.113	0.130	547	0.321	1.000	43.478	5.348
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 (subset 1)	6000	472	42	numeric	2.458	0.059	574	0.096	3.344	50.000	15.152
Rcv1 (subset 2)	6000	472	39	numeric	2.170	0.056	489	0.082	3.215	47.619	15.873
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

By applying some *multi-class imbalance* learner $\mathcal{M}$ on $\mathcal{D}_{jk}^{\mathbf{tri}}$, one multi-class classifier g_{jk} can be induced, i.e. $g_{jk} \leftarrow \mathcal{M}(\mathcal{D}_{jk}^{\mathbf{tri}})$. 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 $\mathcal{I}_K \subset \mathcal{Y} \setminus \{y_j\}$ for pairwise coupling. The real-valued function $f_j(\cdot)$ is then instantiated by aggregating the predictive confidences of one binary-class imbalance learner and K multi-class imbalance learners:

$$f_j(\boldsymbol{x}) = g_j(+1 \mid \boldsymbol{x}) + \sum\nolimits_{y_k \in \mathcal{I}_K} g_{jk}(+2 \mid \boldsymbol{x}) \qquad (5)$$

For the thresholding function $t_j(\cdot)$, COCOA chooses to set it as a constant function $t_j(\boldsymbol{x}) = a_j$. By accompanying the constant a_j with f_j, any example $\boldsymbol{x}$ is predicted to be positive for y_j if $f_j(\boldsymbol{x}) > a_j$, and negative otherwise. Specifically, the "goodness" of a_j is evaluated based on certain metric which measures how well f_j can classify examples in $\mathcal{D}_j$ by using a_j as the bipartition threshold. In this paper, we employ 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 $\mathcal{D}_j$, the thresholding constant a_j is determined by maximizing the corresponding F-measure:

$$a_j = \arg\max\nolimits_{a \in \mathbb{R}} F(f_j, a, \mathcal{D}_j) \qquad (6)$$

Table 1 summarizes the complete procedure of the proposed COCOA approach. For each class label $y_j \in \mathcal{Y}$, one binary-class imbalance learner (Steps 2-3) and K coupling multi-class imbalance learners (Steps 4-8) are induced by manipulating the multi-label training set $\mathcal{D}$. After that, the predictive model for y_j is produced by aggregating the predictive confidences of the induced binary and multi-class classifiers (Steps 9-10). Finally, the predicted label set for the test example is obtained by querying the predictive models of all class labels (Step 12).

It is worth noting that although pairwise coupling (Step 6) only considers second-order correlations among labels, the overall label correlations exploited by COCOA are actually high-order as controlled by the parameter K. Here, COCOA fulfills high-order label correlations by imposing pairwise 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.

3 Related Work

In this section, we briefly discuss existing works related to COCOA. More comprehensive reviews on multi-label learning can be found in survey literatures such as [Tsoumakas *et al.*, 2010; Zhang and Zhou, 2014; Gibaja and Ventura, 2015].

As mentioned in Section 2, one intuitive solution towards class-imbalance multi-label learning is to firstly decompose the multi-label learning problem into q independent binary learning problems, one per class label (a.k.a. *binary relevance*). And then, for each decomposed binary learning problem, the skewness between the positive and negative training examples can be dealt with via popular binary imbalance learning techniques such as random or synthetic undersampling/oversampling [Spyromitros-Xioufis *et al.*, 2011; Tahir *et al.*, 2012]. However, useful information regarding label correlations will be ignored in this decomposition process.

Different from binary decomposition, one can also transform the multi-label learning problem into a multi-class problem by treating any distinct label combinations appearing in the training set as a new class (a.k.a. *label powerset*). After that, the skewness among the transformed classes can be dealt with via off-the-shelf multi-class imbalance learning techniques [Wang and Yao, 2012; Liu *et al.*, 2013]. Although label correlations can be exploited in this transformation process, the number of transformed classes (upper-bounded by $\min(N, 2^q)$) may be too large for any multi-class learner to work well.

Besides applying class-imbalance learning techniques to the transformed binary or multi-class problems, one can also make the multi-label learning algorithm be aware of the class-imbalance issue via parameter tuning. For COCOA, the thresholding constant is calibrated by maximizing imbalance-specific metric such as F-measure based on

Table 3: Performance of each comparing algorithm (mean±std. deviation) in terms of *macro-averaging F-measure* (MACRO-F). In addition, ●/○ indicates whether COCOA is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 1% significance level).

Algorithm	Data Set													win/tie/loss counts for COCOA
	CAL500	Emotions	Medical	Enron	Scene	Yeast	Slashdot	Corel5k	Rcv1 (subset 1)	Rcv1 (subset 2)	Eurlex-sm	Tmc2007	Mediamill	
COCOA	0.207±0.009	0.662±0.013	0.690±0.015	0.324±0.009	0.732±0.013	0.457±0.015	0.327±0.009	0.195±0.004	0.363±0.008	0.337±0.009	0.703±0.007	0.669±0.002	0.459±0.004	/
USAM	0.217±0.006○	0.591±0.016●	0.670±0.012●	0.266±0.011●	0.624±0.008●	0.432±0.010●	0.259±0.010●	0.141±0.004●	0.318±0.005●	0.306±0.005●	0.562±0.007●	0.607±0.002●	0.337±0.003●	**12/0/1**
USAM-EN	0.246±0.004○	0.590±0.018●	0.665±0.025●	0.274±0.010●	0.620±0.011●	0.437±0.012	0.296±0.007●	0.150±0.002●	0.317±0.005●	0.303±0.005●	0.563±0.004●	0.608±0.002●	0.337±0.003●	**11/1/1**
SMOTE	0.237±0.006○	0.584±0.020●	0.672±0.022	0.266±0.006●	0.619±0.007●	0.430±0.006●	0.326±0.005	0.125±0.003●	0.314±0.006●	0.305±0.004●	0.552±0.003●	0.566±0.003●	0.338±0.001●	**10/2/1**
SMOTE-EN	0.239±0.004○	0.582±0.017●	0.672±0.022	0.275±0.004●	0.624±0.007●	0.431±0.005●	0.315±0.007	0.126±0.002●	0.313±0.004●	0.304±0.004●	0.553±0.003●	0.567±0.003●	0.341±0.001●	**10/2/1**
RML	0.209±0.008	0.645±0.016	0.666±0.018	0.309±0.010●	0.684±0.013●	0.471±0.014	0.311±0.009●	0.215±0.009○	0.387±0.020○	0.363±0.029○	0.059±0.003●	0.568±0.039●	0.268±0.019●	**6/4/3**
ML-KNN	0.074±0.002●	0.592±0.026●	0.474±0.031●	0.174±0.011●	0.715±0.011	0.380±0.008●	0.198±0.014●	0.028±0.004●	0.122±0.008●	0.103±0.008●	0.525±0.012●	0.479±0.008●	0.245±0.004●	**12/1/0**
CLR	0.081±0.007●	0.595±0.017●	0.650±0.012●	0.229±0.006●	0.631±0.013●	0.413±0.010●	0.233±0.007●	0.049±0.004●	0.227±0.007●	0.226±0.006●	0.599±0.006●	0.623±0.003●	0.268±0.004●	**13/0/0**
ECC	0.102±0.004●	0.642±0.014●	0.647±0.021●	0.241±0.006●	0.716±0.009	0.394±0.008●	0.250±0.007●	0.064±0.004●	0.216±0.007●	0.199±0.004●	0.619±0.009●	0.642±0.003●	0.277±0.002●	**12/1/0**
RAKEL	0.193±0.003●	0.613±0.018●	0.576±0.014●	0.256±0.006●	0.686±0.008●	0.420±0.005●	0.248±0.006●	0.084±0.005●	0.272±0.007●	0.263±0.005●	0.632±0.008●	0.643±0.004●	0.378±0.002●	**13/0/0**

the training set, which could also be calibrated based on some held-out validation set [Fan and Lin, 2007] or be optimized with an extra learning procedure [Quevedo *et al.*, 2012; Pillai *et al.*, 2013]. Furthermore, instead of only tuning the thresholding parameter, another sophisticated choice is to design multi-label learning algorithms directly optimizing the macro-averaging F-measure [Dembczyński *et al.*, 2013; Petterson and Caetano, 2010].

In view of the randomness in pairwise coupling, COCOA makes use of ensemble learning to aggregate the predictions of K randomly-generated imbalance learners. There have been multi-label learning methods which also utilize ensemble learning to deal with their inherent random factors, such as ensembling chaining classifier with random order [Read *et al.*, 2011] or ensembling multi-class learner derived from random k-labelsets [Tsoumakas *et al.*, 2011a]. Furthermore, ensemble learning can be employed as a meta-strategy to improve generalization with homogeneous [Shi *et al.*, 2011] or heterogeneous [Tahir *et al.*, 2010] component multi-label learners.

4 Experiments

4.1 Experimental Setup

Data Sets

To serve as a solid basis for performance evaluation, a total of thirteen benchmark multi-label data sets have been collected for experimental studies. For each multi-label data set $\mathcal{S}$, 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 are further used to characterize properties of $\mathcal{S}$, whose definitions can be found in [Read *et al.*, 2011] while not detailed here due to page limit.

Let ImR_j denote the imbalance ratio on the j-th class label $(1 \leq j \leq q)$, the level of class-imbalance on $\mathcal{S}$ can be characterized by the `average` imbalance ratio ($\frac{1}{q}\sum_{j=1}^{q} ImR_j$), `minimum` imbalance ratio ($\min_{1 \leq j \leq q} ImR_j$) and `maximum` imbalance ratio ($\max_{1 \leq j \leq q} ImR_j$) across the label space. As a common practice in class-imbalance studies [He and Garcia, 2009], *extreme imbalance* is not considered in this paper. Specifically, any class label with rare appearance (less than 20 positive examples) or with overly-high imbalance ratio ($ImR_j \geq 50$) is excluded from the label space.

Table 2 summarizes characteristics of the experimental data sets, which are roughly ordered according to $|\mathcal{S}|$. As shown in Table 2, the thirteen data sets exhibit diversified properties from different aspects. These data sets cover a broad range of scenarios, including music (`CAL500`, `Emotions`), image (`Scene`, `Corel5k`), video (`Mediamill`), biology (`Yeast`), and text (`the others`). Here, dimensionality reduction is performed on text data sets by retaining features with high document frequency.

Comparing Algorithms

In this paper, COCOA is compared against two series of algorithms. As discussed in Section 3, the first series include several approaches which are capable of dealing with the class-imbalance issue in multi-label data:

Table 4: Performance of each comparing algorithm (mean±std. deviation) in terms of *macro-averaging AUC* (MACRO-AUC). In addition, ●/○ indicates whether COCOA is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 1% significance level).

Algorithm	Data Set						
	CAL500	Emotions	Medical	Enron	Scene	Yeast	Slashdot
COCOA	0.557±0.005	0.843±0.010	0.958±0.006	0.731±0.006	0.943±0.003	0.710±0.006	0.736±0.005
USAM	0.514±0.005●	0.708±0.019●	0.855±0.012●	0.606±0.010●	0.790±0.009●	0.578±0.006●	0.617±0.004●
USAM-EN	0.513±0.004●	0.708±0.015●	0.860±0.024●	0.600±0.004●	0.788±0.009●	0.583±0.006●	0.618±0.004●
SMOTE	0.513±0.005●	0.703±0.019●	0.874±0.019●	0.617±0.007●	0.776±0.008●	0.579±0.006●	0.688±0.008●
SMOTE-EN	0.513±0.004●	0.704±0.013●	0.874±0.019●	0.617±0.007●	0.777±0.011●	0.581±0.007●	0.686±0.008●
RML	–	–	–	–	–	–	–
ML-KNN	0.516±0.007●	0.806±0.015●	0.909±0.008●	0.663±0.006●	0.926±0.005●	0.679±0.004●	0.676±0.006●
CLR	0.561±0.004○	0.796±0.010●	0.948±0.008●	0.709±0.007●	0.894±0.005●	0.650±0.004●	0.698±0.009●
ECC	0.549±0.007●	0.841±0.009	0.925±0.009●	0.723±0.006●	0.938±0.003●	0.689±0.006●	0.706±0.009●
RAKEL	0.528±0.005●	0.797±0.015●	0.828±0.006●	0.640±0.003●	0.892±0.004●	0.640±0.004●	0.612±0.002●

Algorithm	Data Set						**win/tie/loss counts for** COCOA
	Corel5k	Rcv1 (subset 1)	Rcv1 (subset 2)	Eurlex-sm	Tmc2007	Mediamill	
COCOA	0.719±0.004	0.889±0.003	0.882±0.002	0.957±0.002	0.930±0.001	0.843±0.001	/
USAM	0.572±0.003●	0.674±0.010●	0.672±0.009●	0.788±0.009●	0.801±0.003●	0.655±0.004●	**13/0/0/**
USAM-EN	0.574±0.002●	0.676±0.010●	0.671±0.010●	0.789±0.006●	0.800±0.003●	0.654±0.006●	**13/0/0/**
SMOTE	0.597±0.004●	0.625±0.009●	0.620±0.008●	0.795±0.005●	0.793±0.003●	0.669±0.002●	**13/0/0/**
SMOTE-EN	0.596±0.004●	0.626±0.006●	0.620±0.009●	0.795±0.004●	0.793±0.003●	0.670±0.002●	**13/0/0/**
RML	–	–	–	–	–	–	–
ML-KNN	0.590±0.005●	0.718±0.009●	0.710±0.009●	0.887±0.004●	0.849±0.003●	0.767±0.001●	**13/0/0/**
CLR	0.740±0.002○	0.891±0.003	0.882±0.002	0.944±0.001	0.906±0.001●	0.805±0.001●	**8/3/2**
ECC	0.697±0.006●	0.864±0.002●	0.855±0.003●	0.945±0.002●	0.921±0.001●	0.826±0.001●	**12/1/0**
RAKEL	0.552±0.002●	0.728±0.003●	0.721±0.003●	0.872±0.005●	0.859±0.002●	0.737±0.001●	**13/0/0/**

[*] MACRO-AUC not applicable to RML, which does not yield real-valued outputs on each class label [Petterson and Caetano, 2010].

- Undersampling (USAM): 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.
- SMOTE: 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 [Chawla *et al.*, 2002] to form the new binary training set.

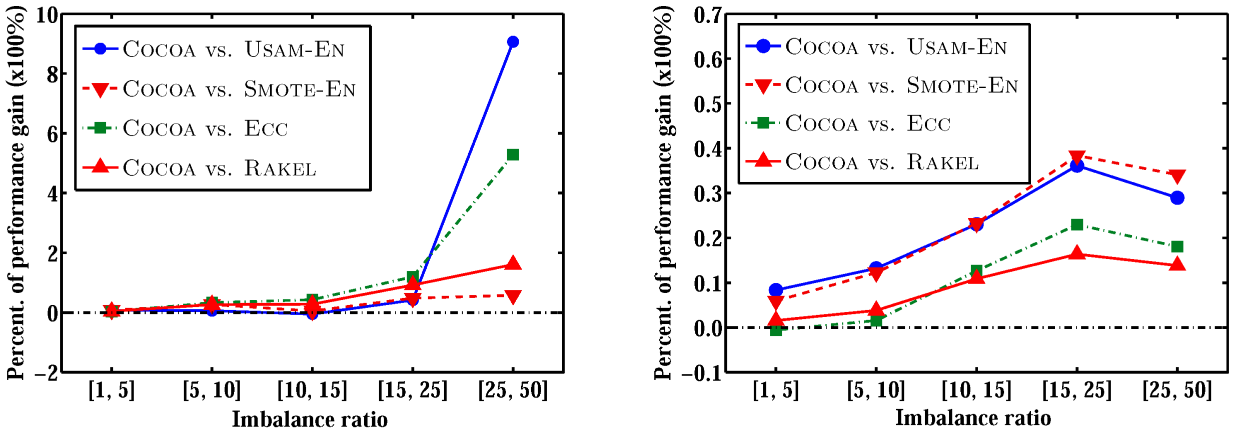
  Considering that COCOA utilizes ensemble learning in its learning process, an ensemble version of USAM and SMOTE are also employed for comparison (named as USAM-EN and SMOTE-EN).
- RML: Besides integrating binary decomposition with undersampling/oversampling, another way to handle class-imbalance is to design learning system which can directly optimize imbalance-specific metric. Here, the RML approach [Petterson and Caetano, 2010] is employed as another comparing algorithm, which maximizes macro-averaging F-measure on multi-label data via convex relaxation.

In addition to the above algorithms, the second series include several well-established multi-label learning algorithms [Zhang and Zhou, 2014], including first-order approach ML-KNN [Zhang and Zhou, 2007], second-order approach CLR [Fürnkranz *et al.*, 2008], and high-order approaches ECC [Read *et al.*, 2011] and RAKEL [Tsoumakas *et al.*, 2011a].

In this paper, all the comparing algorithms are instantiated as follows: 1) For USAM, SMOTE and their ensemble versions, decision tree is used as the base learner due to its popularity in class-imbalance studies [He and Garcia, 2009]. Specifically, these algorithms adopt implementations provided by the widely-used Weka platform with J48 decision tree (C4.5 implementation in Weka) as their base learner [Hall *et al.*, 2009]; 2) For RML, the original implementation provided in the literature is used; 3) For the second series of algorithms (ML-KNN, CLR, ECC and RAKEL), we adopt their canonical implementations provided by the MULAN multi-label learning library (upon Weka platform) with suggested parameter configurations [Tsoumakas *et al.*, 2011b]; 4) For COCOA, both the binary-class and multi-class imbalance learners ($\mathcal{B}$ and $\mathcal{M}$) are implemented in Weka using J48 decision tree with undersampling [Hall *et al.*, 2009]. Furthermore, the number of coupling class labels is set as $K = \min(q - 1, 10)$. For fair comparison, the ensemble size for USAM-EN and SMOTE-EN is also set to be 10.

4.2 Experimental Results

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 [He and Garcia, 2009]. In this paper, the multi-label classification performance is accordingly evaluated by *macro-averaging* the metric values across all class labels [Zhang and



(a) Enron

(b) Eurlex-sm

Figure 1: Performance gain between Cocoa and the comparing algorithm (PG_k) changes as the level of imbalance ratio (I_k) increases. On either data set, the performance of each algorithm is evaluated based on F-measure.

Zhou, 2014]. For either macro-averaging metric, the higher the metric value the better the performance.

Tables 3 and 4 give the detailed experimental results in terms of each evaluation metric respectively. Each data set is randomly split for training and testing, where 50% examples are chosen to form the training set and the remaining ones 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.

Furthermore, to show whether Cocoa performs significantly better/worse than the comparing algorithm, pairwise t-test at 1% significance level is conducted. Accordingly, a win/loss is counted and a marker •/○ is shown in the table whenever Cocoa achieves significantly superior/inferior performance on one data set. Otherwise, a tie is counted and no marker is given. The overall win/tie/loss counts across all data sets are summarized at the last column of each table.

In terms of Macro-F (Table 3), Cocoa significantly outperforms the comparing algorithms in 46.2% (Rml), 76.9% (Smote, Smote-En), 84.6% (Usam-En), 92.3% (Usam, Ml-knn, Ecc) and 100% (Clr, Rakel) cases, and hasn't been outperformed by algorithms in the second series. These results indicate that Cocoa is capable of achieving good balance between predictive exactness (precision) and completeness (recall) in handling class-imbalance multi-label learning.

In terms of Macro-Auc (Table 4), Cocoa significantly outperforms the comparing algorithms in 61.5% (Clr), 92.3% (Ecc) and 100% (Usam, Usam-En, Smote, Smote-En, Ml-knn, Rakel) cases, while has only been outperformed by Clr twice. These results indicate that the real-valued functions $f_j(\cdot)$ $(1 \leq j \leq q)$ learned by Cocoa is capable of yielding reasonable predictive confidence, and better classification performance can be further expected if it is combined with more sophisticated thresholding strategy other than the constant function (Table 1, Step 10).

To further investigate how Cocoa works under different levels of imbalance ratios, we roughly group the imbalance ratio ImR_j into five intervals I_k $(1 \leq k \leq 5)$ in ascending orders, i.e. $I_1 = [1,5]$, $I_2 = [5,10]$, $I_3 = [10,15]$, $I_4 = [15,25]$ and $I_5 = [25,50]$. Given one multi-label data set, let A_k denote the average performance of Cocoa over class labels whose imbalance ratios fall into I_k, and B_k denote the average performance of another comparing algorithm over class labels in the same interval. Accordingly, the percentage of performance gain, i.e. $PG_k = [(A_k - B_k)/B_k] \times 100\%$, is computed to reflect the relative performance between Cocoa and the comparing algorithm within the given interval.

Figure 1 illustrates how PG_k changes as the imbalance level I_k moves from I_1 to I_5. Due to page limit, the performance is evaluated by choosing the F-measure metric $\left(\frac{1}{|I_k|}\sum_{ImR_j \in I_k} F_j\right)$ and two data sets `Enron` and `Eurlex-sm` are considered. For brevity, the relative performance against four comparing algorithms (Usam-En, Smote-En, Ecc and Rakel) has been depicted. Similar trends can be observed for other evaluation metrics and comparing algorithms.

As shown in Figure 1, Cocoa maintains good relative performance against the comparing algorithms across different imbalance levels, where the curves hardly drop below the baseline ($PG_k = 0$). Furthermore, it is interesting that the performance advantage of Cocoa becomes more pronounced when the level of imbalance ratio is high (for $I_4 = [15,25]$ and $I_5 = [25,50]$). These results indicate that Cocoa can provide robust and preferable solutions in diverse class-imbalance scenarios.

5 Conclusion

In this paper, the class-imbalance issue in learning from multi-label data is studied. Accordingly, a novel class-imbalance multi-label learning algorithm named Cocoa is proposed, which works by leveraging the exploitation of label correlations and the exploration of class-imbalance. Specifically, one binary-class imbalance learner and several coupling multi-class imbalance learners are combined to yield the predictive model. Extensive experiments across thirteen benchmark data sets show that Cocoa performs favorably against the comparing algorithms, especially in terms of imbalance-specific metrics such as Macro-F and Macro-Auc.

Acknowledgments

The authors wish to thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Science Foundation of China (61175049, 61222309, 61473087), the Natural Science Foundation of Jiangsu Province (BK20141340), and the MOE Program for New Century Excellent Talents in University (NCET-13-0130).

References

[Chawla *et al.*, 2002] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[Dembczyński *et al.*, 2013] 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 *Proceedings of the 30th International Conference on Machine Learning*, pages 1130–1138, Atlanta, GA, 2013.

[Fan and Lin, 2007] R.-E. Fan and C.-J. Lin. A study on threshold selection for multi-label classification. Technical report, Department of Computer Science & Information Engineering, National Taiwan University, 2007.

[Fürnkranz *et al.*, 2008] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[Gibaja and Ventura, 2015] E. Gibaja and S. Ventura. A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3):Article 52, 2015.

[Hall *et al.*, 2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[He and Garcia, 2009] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[Liu *et al.*, 2013] X.-Y. Liu, Q.-Q. Li, and Z.-H. Zhou. Learning imbalanced multi-class data with optimal dichotomy weights. In *Proceedings of the 13th IEEE International Conference on Data Mining*, pages 478–487, Dallas, TX, 2013.

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

[Pillai *et al.*, 2013] I. Pillai, G. Fumera, and F. Roli. Threshold optimisation for multi-label classifiers. *Pattern Recognition*, 46(7):2055–2065, 2013.

[Quevedo *et al.*, 2012] J. R. Quevedo, O. Luaces, and A. Bahamonde. Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recognition*, 45(2):876–883, 2012.

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

[Shi *et al.*, 2011] C. Shi, X. Kong, P. S. Yu, and B. Wang. Multi-label ensemble learning. In *Lecture Notes in Artificial Intelligence 6913*, pages 223–239. Springer, Berlin, 2011.

[Spyromitros-Xioufis *et al.*, 2011] E. Spyromitros-Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas. Dealing with concept drift and class imbalance in multi-label stream classification. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1583–1588, Barcelona, Spain, 2011.

[Tahir *et al.*, 2010] M. A. Tahir, J. Kittler, K. Mikolajczyk, and F. Yan. Improving multilabel classification performance by using ensemble of multi-label classifiers. In *Lecture Notes in Computer Science 5997*, pages 11–21. Springer, Berlin, 2010.

[Tahir *et al.*, 2012] M. A. Tahir, J. Kittler, and F. Yan. Inverse random under sampling for class imbalance problem and its application to multi-label classification. *Pattern Recognition*, 45(10):3738–3750, 2012.

[Tsoumakas *et al.*, 2010] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–686. Springer, Berlin, 2010.

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

[Tsoumakas *et al.*, 2011b] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas. MULAN: A java library for multi-label learning. *Journal of Machine Learning Research*, 12(Jul):2411–2414, 2011.

[Wang and Yao, 2012] S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 42(4):1119–1130, 2012.

[Zhang and Zhou, 2007] M.-L. Zhang and Z.-H. Zhou. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[Zhang and Zhou, 2014] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.