Towards Enabling Binary Decomposition for Partial Multi-Label Learning

Bing-Qing Liu, Bin-Bin Jia, Min-Ling Zhang, *Senior Member, IEEE*

Abstract—Partial multi-label learning (PML) is an emerging weakly supervised learning framework, where each training example is associated with multiple candidate labels which are only partially valid. To learn the multi-label predictive model from PML training examples, most existing approaches work by identifying valid labels within candidate label set via label confidence estimation. In this paper, a novel strategy towards partial multi-label learning is proposed by enabling binary decomposition for handling PML training examples. Specifically, the widely used error-correcting output codes (ECOC) techniques are adapted to transform the PML learning problem into a number of binary learning problems, which refrains from using the error-prone procedure of estimating labeling confidence of individual candidate label. In the encoding phase, a ternary encoding scheme is utilized to balance the *definiteness* and *adequacy* of the derived binary training set. In the decoding phase, a loss weighted scheme is applied to consider the empirical performance and predictive margin of derived binary classifiers. Extensive comparative studies against state-of-the-art PML learning approaches clearly show the performance advantage of the proposed binary decomposition strategy for partial multi-label learning.

Index Terms—Machine learning, partial multi-label learning, binary decomposition, error-correcting output codes

1 Introduction

Multi-label learning [13], [50], [53] deals with the problem where each example is associated with multiple class labels, which has been widely utilized in learning from real-world objects with rich semantics [14], [17], [18], [24], [27], [31], [33], [38], [42]. Most works on multi-label learning assume that the class labels associated with the training example are valid ones, which may not hold in real-world scenarios due to the difficulty of acquiring accurate labeling information. One commonly encountered practical situation is that the set of labels acquired for the real-world object are only *candidate* ones which are partially valid. For instance, in online image tagging (Fig.1a), a set of candidate labels can be acquired from the crowdsourcing annotators while only some of the labels would be valid ones due to potential carelessness of the annotators; in text categorization (Fig.1b), a set of candidate labels can be acquired from the human labeler where some of the labels are false positive ones due to the labeler's misunderstanding of the document semantics.

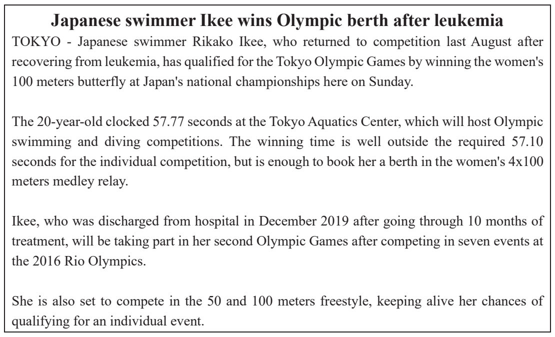
To deal with the inaccurate labeling information in learning from multi-label examples, the task of *partial multi-label learning* (PML) has attracted significant attentions in recent years as an emerging weakly supervised learning framework [11], [22], [23], [30], [32], [35], [36]. Formally, let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional feature space and $\mathcal{Y} = \{y_1, y_2, ..., y_q\}$ denote the output space with q possible class labels. Given the PML training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid (1 \leq i \leq m)\}$ where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional instance and $Y_i \subseteq \mathcal{Y}$ is the candidate label set associated with $\boldsymbol{x}_i$, it is assumed that the ground-truth label set $\widetilde{Y}_i$ for $\boldsymbol{x}_i$ resides in its candidate label set Y_i, i.e. $\widetilde{Y}_i \subseteq Y_i$. Accordingly, the task of partial multi-label learning is to induce a multi-label predictor $f : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ from $\mathcal{D}$ which can assign a set of proper labels for the unseen instance.

As the ground-truth labeling information of PML training example is not directly accessible to the learning approach, most existing approaches work by estimating the labeling confidence of each candidate label being the ground-truth label, which can be instantiated via an iterative [15], [30], [35], [40], [43] or stage-wise [11], [32], [39] procedure. However, the estimated labeling confidence might be error-prone due to the inherent uncertainty brought by the initialization and optimization steps of the iterative or stage-wise procedure, especially when the proportion of false positive labels of PML training examples is high.

In this paper, a novel strategy to learn from PML examples is proposed which refrains from using the error-prone procedure of estimating labeling confidence of individual candidate label. Specifically, a transformation-based PML learning approach named Pamb, i.e. *PArtial Multi-label learning via Binary decomposition*, is proposed. Pamb works by transforming the original partial multi-label learning problem into a number of binary learning problems where the popular error-correcting output codes (ECOC) techniques are adapted to fit training examples with candidate label sets. Briefly, in the encoding phase, a number of binary training sets are derived from original PML training examples based on a ternary encoding scheme balancing two factors of *definiteness* and *adequacy*. Here, *definiteness* refers to the confidence that examples in one derived binary training set do belong to the derived binary training set where indefiniteness comes from the existence of irrelevant labels

- *Bing-Qing Liu is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China. Email: liubq@seu.edu.cn*
- *Bin-Bin Jia is with the College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China. Email: jiabinbin@lut.edu.cn*
- *Min-Ling Zhang (corresponding author) is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. E-mail: zhangml@seu.edu.cn*

(a) online image tagging



Japanese swimmer Ikee wins Olympic berth after leukemia

TOKYO - Japanese swimmer Rikako Ikee, who returned to competition last August after recovering from leukemia, has qualified for the Tokyo Olympic Games by winning the women's 100 meters butterfly at Japan's national championships here on Sunday.

The 20-year-old clocked 57.77 seconds at the Tokyo Aquatics Center, which will host Olympic swimming and diving competitions. The winning time is well outside the required 57.10 seconds for the individual competition, but is enough to book her a berth in the women's 4x100 meters medley relay.

Ikee, who was discharged from hospital in December 2019 after going through 10 months of treatment, will be taking part in her second Olympic Games after competing in seven events at the 2016 Rio Olympics.

She is also set to compete in the 50 and 100 meters freestyle, keeping alive her chances of qualifying for an individual event.

Candidate Labels: Tokyo Olympic diving swimming Brazil

(b) text categorization

Fig. 1. Two exemplary scenarios of partial multi-label learning, where the ground-truth labels among the set of candidate labels are shown in red.

in candidate label sets, and *adequacy* refers to the degree that the data distribution of one derived binary training set approximates its ground-truth implicit data distribution where inadequacy comes from the examples discarding rule in ECOC as well as their large candidate label sets. In the decoding phase, the predictions on unseen instance are yielded based on a loss weighted scheme synthesizing the empirical performance and predictive margin of derived binary classifiers. Experimental studies show that PAMB is capable of achieving highly competitive performance against state-of-the-art approaches by employing the binary decomposition strategy for partial multi-label learning.

The rest of this paper is organized as follows. Section 2 discusses related works on partial multi-label learning. Section 3 presents technical details of the proposed PAMB approach. Section 4 reports experimental results of comparative studies. Finally, Section 5 concludes this paper and indicates several issues for future work.

2 Related work

In PML, each training example is associated with multiple candidate labels, among which multiple labels can be relevant. There are two popular learning frameworks which are highly related to PML, including multi-label learning (MLL) [13], [50], [53] and partial label learning (PLL) [6], [21], [48]. In MLL, each training example can be associated with multiple labels and all of them are relevant. In PLL, each training example is associated with multiple candidate labels, among which only one is relevant. Compared with MLL, each PML training examples can also be associated with multiple relevant labels but they are concealed in candidate label set. Compared with PLL, each PML training example is also associated with multiple candidate labels, but there can be more than one relevant label among candidate label set. In other words, the characteristics of both MLL and PLL exist in the PML problem, which leads to that it is more challenging to deal with the PML problem than MLL and PLL.

The PML problem can be intuitively solved by regarding all candidate labels as relevant ones and then inducing a multi-label predictor with any off-the-shelf multi-label algorithms. However, the resulting PML model in this way will be misled by the irrelevant labels in candidate label set. To alleviate the negative impact of inaccurate labeling information for model induction, most existing PML approaches work by estimating the labeling confidence of each candidate label being the ground-truth label in an *iterative* or a *stage-wise* manner. Iterative approaches alternately optimize the predictive model and estimate the labeling confidence with some constraints or assumptions for ground-truth labeling information, such as low rank approximation [30], [36], [43], [44], feature-induced regularization [4], [35], [43], [45], label correlation exploitation [28], [35], [41], manifold assumption [19], [29], and prior knowledge [37], etc. However, the objective function is generally convex only in each optimization step, and the obtained labeling confidence cannot be globally optimal and might be affected by initialization. Stage-wise approaches firstly estimate the labeling confidence and then learn the predictive model based on the estimated results. PARTICLE [11] directly elicits the credible labels from candidate label set and then induces a tailored multi-label predictor, while both DRAMA [32] and PML-LD [39] estimate the labeling confidence by exploiting the information in feature space and then induce a multi-output regression model supervised by the estimated confidence scores. However, the performance of the learned model in second stage will be degenerated if the supervisory information obtained in the first stage is inaccurate. Therefore, the label confidence estimation procedure might be error-prone in either *iterative* or *stage-wise* methods.

Binary decomposition is initially designed to transform the multi-class classification problem into multiple binary classification problems, where one-vs-rest (OvR), one-vs-one (OvO) and error-correcting output codes (ECOC) [8] are the three most widely-used decomposition strategies. Binary decomposition has also been adapted to solve the related learning frameworks MLL and PLL. For MLL, many methods have been designed to decompose the MLL problem into multiple binary classification problems by respectively adapting OvR [3], [46], [49], OvO [12] and ECOC [2], [16]. For PLL, some methods which are developed by adapting OvO [34] or ECOC [20], [48] have also achieved state-of-the-art performance. In this paper, we aim at enabling binary decomposition for handling PML training examples, where ECOC is adapted to transform the PML problem into multiple binary classification problems. Accordingly, a novel approach named PAMB is proposed and its technical details will be presented in the next section.

3 THE PAMB APPROACH

In this section, we firstly give a brief introduction to the error correcting output codes (ECOC) method which is the basis of the proposed PAMB approach. After that, we present the technical details of our PAMB approach with two subsections which introduce its encoding phase and decoding phase, respectively.

3.1 ECOC Overview

The ECOC method is one of the popular strategies which can decompose a multi-class classification problem into multiple binary classification problems. Specifically, given a multi-class classification problem with q classes, i.e., $\mathcal{Y} = \{c_1, c_2, \ldots, c_q\}$, there are two main steps in the ECOC method, i.e., encoding and decoding.

In the encoding phase, the main task is to generate a coding matrix $\mathbf{M}$ to determine how the class space is partitioned. The two commonly used designs of coding matrix are binary coding [8] (i.e., $\mathbf{M} \in \{+1, -1\}^{q \times L}$) and ternary coding [1] (i.e., $\mathbf{M} \in \{+1, 0, -1\}^{q \times L}$). For the coding matrix, its j-th row $\mathbf{M}(j,:)$ represents a unique L-bits codeword for the j-th class label c_j ($1 \le j \le q$), and its l-th column $\mathbf{M}(:,l)$ partitions the class space $\mathcal{Y}$ into two or three parts: positive part $\mathcal{Y}_l^+$, negative part $\mathcal{Y}_l^-$ and neutral part $\mathcal{Y}_l^0$ which only exists for ternary coding:

$$\begin{aligned} \mathcal{Y}_l^+ &= \{c_j \mid \mathbf{M}(j,l) = +1, 1 \le j \le q\} \\ \mathcal{Y}_l^- &= \{c_j \mid \mathbf{M}(j,l) = -1, 1 \le j \le q\} \\ \mathcal{Y}_l^0 &= \{c_j \mid \mathbf{M}(j,l) = \ \ 0, 1 \le j \le q\} \end{aligned}$$

According to the l-th column of $\mathbf{M}$, a binary classification data set can be constructed where the instances associated with the label in $\mathcal{Y}_l^+$ (or $\mathcal{Y}_l^-$) are used as positive (or negative) samples, respectively, and the remaining instances associated with the label in $\mathcal{Y}_l^0$ will be disgarded. Based on the constructed data set, a binary classifier h_l can be learned and finally a total of L classifiers can be obtained according to $\mathbf{M}$.

In the decoding phase, the main task is to obtain the final prediction of unseen instance $\boldsymbol{x}_*$ according to the outputs of the L learned binary classifiers:

$$\boldsymbol{h}(\boldsymbol{x}_*) = [h_1(\boldsymbol{x}_*), h_2(\boldsymbol{x}_*), ..., h_L(\boldsymbol{x}_*)]^\top$$

Generally, the final prediction for $\boldsymbol{x}_*$ is determined as follows:

$$f(\boldsymbol{x}_*) = c_{\hat{j}}, \text{where } \hat{j} = \arg\min_{1 \le j \le q} dist(\boldsymbol{h}(\boldsymbol{x}_*), \mathbf{M}(j,:))$$

Here, $dist(\cdot,\cdot)$ is some kind of distance function (e.g., hamming distance). In other words, the class whose codeword is closest to the predicted vector $\boldsymbol{h}(\boldsymbol{x}_*)$ will be considered as the final prediction.

According to the technical details of ECOC, it is easy to know that both one-vs-rest (OvR) and one-vs-one (OvO) can be regarded as a special case of ECOC. Besides, the vanilla ECOC strategy simply generates the coding matrix in a random manner, while the specific characteristics of practical problems are not considered which might lead to suboptimal models. Thus, there are some works focusing on problem-dependent encoding strategies which try to find the coding matrix most suitable for given problems, e.g., DECOC [26], SECOC [10], M^{2}ECOC [52] and SM^{2}ECOC [51].

As to the decoding phase of ECOC, the key is how to utilize the outputs of the L learned binary classifiers to determine the final prediction, i.e., the design of distance function $dist(\cdot,\cdot)$. Commonly used decoding strategies include hamming decoding [8], attenuated Euclidean decoding [25], loss-based decoding [1] and weighted loss-based decoding [9], etc., where loss-weighted decoding has claimed better performance because it can utilize each binary classifier's empirical performance and predictive confidence.

The PML problem is a more complicated learning setting than multi-class classification due to the existence of irrelevant labels in candidate label set and the possible multiple ground-truth relevant labels. In this paper, the proposed PAMB approach solves the PML problem by adapting ECOC with ternary encoding and loss-weighted decoding. For ternary encoding, we aim at dealing with large candidate label set with the '0's in coding matrix. For loss-weighted decoding, we aim at taking full advantage of the ability of learned binary classifiers. Technical details of PAMB's encoding phase and decoding phase are scrutinized in the following Subsection 3.2 and Subsection 3.3, respectively.

3.2 Encoding Phase

In the encoding phase, PAMB induces a total of L binary classifier over the binary training sets derived from the original PML training set $\mathcal{D}$ w.r.t. each column of the ternary coding matrix $\mathbf{M}$.

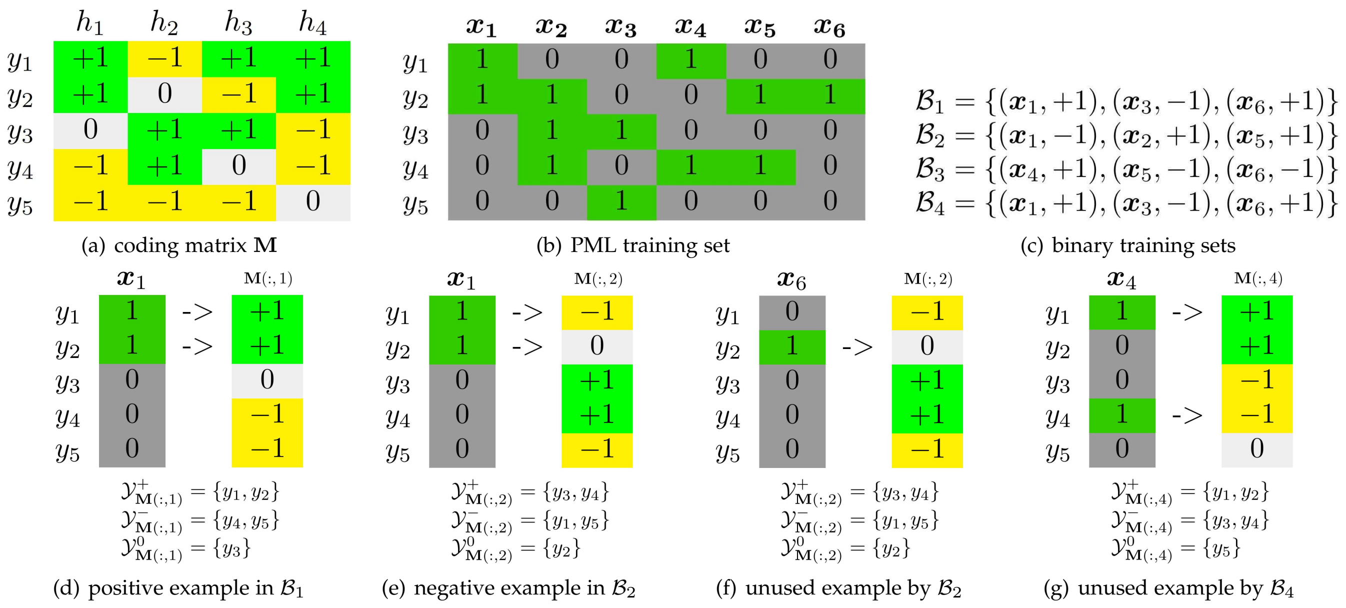
Specifically, let $\boldsymbol{v} \in \{+1, 0, -1\}^q$ be a randomly generated q-bits column ternary vector, where the number of '0's is z and the number of both '$+1$'s and '-1's is $\frac{q-z}{2}$. Here, z is a hyper-parameter of PAMB and its value is usually set to something which makes $q - z$ even. With $\boldsymbol{v}$, the label space can be divided into positive groups $\mathcal{Y}_{\boldsymbol{v}}^+$, negative group $\mathcal{Y}_{\boldsymbol{v}}^-$ and neutral group $\mathcal{Y}_{\boldsymbol{v}}^0$:

$$\begin{aligned} \mathcal{Y}_{\boldsymbol{v}}^+ &= \{y_j \mid v_j = +1, 1 \le j \le q\} \\ \mathcal{Y}_{\boldsymbol{v}}^- &= \{y_j \mid v_j = -1, 1 \le j \le q\} \\ \mathcal{Y}_{\boldsymbol{v}}^0 &= \{y_j \mid v_j = \ \ 0, 1 \le j \le q\} \end{aligned} \tag{1}$$

where v_j denotes the j-th item in $\boldsymbol{v}$. Instead of estimating labeling confidence of individual candidate label, PAMB regards the candidate label set of each PML training example as an entirety to construct a binary training set $\mathcal{B}_{\boldsymbol{v}}$ according to the divided label space by ternary vector $\boldsymbol{v}$ in Eq.(1). Specifically, $\boldsymbol{x}_i$ is used as a positive (or negative) example when the following condition C_i^+ (or C_i^-) is true:

$$\begin{aligned} C_i^+ &= (Y_i \cap \mathcal{Y}_{\boldsymbol{v}}^+ \neq \varnothing) \wedge (Y_i \cap \mathcal{Y}_{\boldsymbol{v}}^- = \varnothing) \\ C_i^- &= (Y_i \cap \mathcal{Y}_{\boldsymbol{v}}^+ = \varnothing) \wedge (Y_i \cap \mathcal{Y}_{\boldsymbol{v}}^- \neq \varnothing) \end{aligned}$$

In other words, if Y_i entirely falls into the union of $\mathcal{Y}_{\boldsymbol{v}}^+$ (or $\mathcal{Y}_{\boldsymbol{v}}^-$) and $\mathcal{Y}_{\boldsymbol{v}}^0$ (i.e., $Y_i \cap \mathcal{Y}_{\boldsymbol{v}}^- = \varnothing$ or $Y_i \cap \mathcal{Y}_{\boldsymbol{v}}^+ = \varnothing$) and its intersection with $\mathcal{Y}_{\boldsymbol{v}}^+$ (or $\mathcal{Y}_{\boldsymbol{v}}^-$) is not empty (i.e., $Y_i \cap \mathcal{Y}_{\boldsymbol{v}}^+ \neq \varnothing$ or $Y_i \cap \mathcal{Y}_{\boldsymbol{v}}^- \neq \varnothing$), then $\boldsymbol{x}_i$ is used as a positive (or negative)



$$\begin{array}{c|cccc} & h_1 & h_2 & h_3 & h_4 \\ \hline y_1 & +1 & -1 & +1 & +1 \\ y_2 & +1 & 0 & -1 & +1 \\ y_3 & 0 & +1 & +1 & -1 \\ y_4 & -1 & +1 & 0 & -1 \\ y_5 & -1 & -1 & -1 & 0 \end{array}$$

(a) coding matrix $\mathbf{M}$

$$\begin{array}{c|cccccc} & \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \boldsymbol{x}_4 & \boldsymbol{x}_5 & \boldsymbol{x}_6 \\ \hline y_1 & 1 & 0 & 0 & 1 & 0 & 0 \\ y_2 & 1 & 1 & 0 & 0 & 1 & 1 \\ y_3 & 0 & 1 & 1 & 0 & 0 & 0 \\ y_4 & 0 & 1 & 0 & 1 & 1 & 0 \\ y_5 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$$

(b) PML training set

$$\mathcal{B}_1 = \{(\boldsymbol{x}_1,+1),(\boldsymbol{x}_3,-1),(\boldsymbol{x}_6,+1)\}$$
$$\mathcal{B}_2 = \{(\boldsymbol{x}_1,-1),(\boldsymbol{x}_2,+1),(\boldsymbol{x}_5,+1)\}$$
$$\mathcal{B}_3 = \{(\boldsymbol{x}_4,+1),(\boldsymbol{x}_5,-1),(\boldsymbol{x}_6,-1)\}$$
$$\mathcal{B}_4 = \{(\boldsymbol{x}_1,+1),(\boldsymbol{x}_3,-1),(\boldsymbol{x}_6,+1)\}$$

(c) binary training sets

$\mathcal{Y}^+_{\mathbf{M}(:,1)} = \{y_1, y_2\}$, $\mathcal{Y}^-_{\mathbf{M}(:,1)} = \{y_4, y_5\}$, $\mathcal{Y}^0_{\mathbf{M}(:,1)} = \{y_3\}$

(d) positive example in $\mathcal{B}_1$

$\mathcal{Y}^+_{\mathbf{M}(:,2)} = \{y_3, y_4\}$, $\mathcal{Y}^-_{\mathbf{M}(:,2)} = \{y_1, y_5\}$, $\mathcal{Y}^0_{\mathbf{M}(:,2)} = \{y_2\}$

(e) negative example in $\mathcal{B}_2$

$\mathcal{Y}^+_{\mathbf{M}(:,2)} = \{y_3, y_4\}$, $\mathcal{Y}^-_{\mathbf{M}(:,2)} = \{y_1, y_5\}$, $\mathcal{Y}^0_{\mathbf{M}(:,2)} = \{y_2\}$

(f) unused example by $\mathcal{B}_2$

$\mathcal{Y}^+_{\mathbf{M}(:,4)} = \{y_1, y_2\}$, $\mathcal{Y}^-_{\mathbf{M}(:,4)} = \{y_3, y_4\}$, $\mathcal{Y}^0_{\mathbf{M}(:,4)} = \{y_5\}$

(g) unused example by $\mathcal{B}_4$

Fig. 2. An illustrative example of PAMB's encoding phase. Here, we denote $\mathcal{B}_{\mathbf{M}(:,l)}$ by $\mathcal{B}_l$ for brevity ($1 \leq l \leq 4$). (a) A 5×4 coding matrix, where the symbol on the left of each row represents that the codeword of label y_j corresponds to the vector in the row ($1 \leq j \leq 5$) and the symbol on the top of each column represents that the classifier h_l is learned according to the ternary code vector in the column ($1 \leq l \leq 4$); (b) A PML training set with six examples where each row corresponds to one label and each column corresponds to the label vector of one instance; (c) Four binary training sets constructed according to the four columns of coding matrix; (d) $\boldsymbol{x}_1$ is used as positive example in $\mathcal{B}_1$ according to the 1st column of $\mathbf{M}$; (e) $\boldsymbol{x}_1$ is used as negative example in $\mathcal{B}_2$ according to the 2nd column of $\mathbf{M}$; (f) $\boldsymbol{x}_6$ is unused in $\mathcal{B}_2$ according to the 2nd column of $\mathbf{M}$; (g) $\boldsymbol{x}_4$ is unused in $\mathcal{B}_4$ according to the 4th column of $\mathbf{M}$.

example in $\mathcal{B}_{\boldsymbol{v}}$. Formally speaking, $\mathcal{B}_{\boldsymbol{v}}$ is constructed as follows:

$$\begin{aligned}\mathcal{B}_{\boldsymbol{v}} = &\{(\boldsymbol{x}_i,+1) \mid C_i^+ = true, 1 \leq i \leq m\} \bigcup \\ &\{(\boldsymbol{x}_i,-1) \mid C_i^- = true, 1 \leq i \leq m\}\end{aligned} \tag{2}$$

As shown in Eq.(2), not all PML training examples will contribute to construct the binary training set $\mathcal{B}_{\boldsymbol{v}}$. To guarantee informative training set with enough samples, PAMB discards the current randomly generated ternary vector $\boldsymbol{v}$ when the corresponding $\mathcal{B}_{\boldsymbol{v}}$ does not meet the pre-set eligibility condition.[1] If $\mathcal{B}_{\boldsymbol{v}}$ is admissible, PAMB records $\boldsymbol{v}$ as one column of coding matrix $\mathbf{M}$, i.e., $\mathbf{M}(:,l) = \boldsymbol{v}$ where $\mathbf{M}(:,l)$ denotes $\mathbf{M}$'s l-th column ($1 \leq l \leq L$). After $\mathbf{M}$ is obtained, PAMB induces a binary classifier $h_l(\cdot) = \text{sign}(f_l(\cdot))$ by invoking the employed binary classification algorithm $\mathfrak{L}$ on $\mathcal{B}_{\mathbf{M}(:,l)}$: $h_l \leftarrow \mathfrak{L}(\mathcal{B}_{\mathbf{M}(:,l)})$. Here, $\mathcal{B}_{\mathbf{M}(:,l)}$ is the binary training set which is constructed according to the divided label space by ternary vector $\mathbf{M}(:,l)$ in Eq.(1), $h_l(\boldsymbol{x}) \in \{+1,-1\}$ returns the predicted binary label for instance $\boldsymbol{x}$ while $f_l(\boldsymbol{x}) \in \mathbb{R}$ returns the predicted confidence for instance $\boldsymbol{x}$, where $\text{sign}(\cdot)$ is the signed function.

Fig.2 shows an illustrative example for the encoding phase of PAMB.[2] It is worth noting that not every randomly generated $\boldsymbol{v}$ will be accepted as one column of $\mathbf{M}$. In other words, there is no necessary correspondence between $\boldsymbol{v}$ and the columns of $\mathbf{M}$. Therefore, we cannot use $\boldsymbol{v}_l$ to represent the l-th column of $\mathbf{M}$, i.e., $\mathbf{M}(:,l)$. In this example, the value of z is set to 1, so the number of '+1's,'−1's and '0's in each column of coding matirx $\mathbf{M}$ (Fig.2a) is 2, 2, 1, respectively. The matrix shown in Fig.2b corresponds to the label matrix of the PML training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq 6\}$ with six examples, where each column corresponds to one training example and each row corresponds to one label. If the item in the i-th column and j-th row equals 1, then $y_j \in Y_i$; otherwise, $y_j \notin Y_i$. Take $(\boldsymbol{x}_1, Y_1)$ as an example, because $Y_1 = \{y_1, y_2\}$, $\boldsymbol{x}_1$ is used as a positive example in $\mathcal{B}_{\mathbf{M}(:,1)}$ as Y_1 entirely falls into $\mathcal{Y}^+_{\mathbf{M}(:,1)}$ (Fig.2d), and is also used as a negative example in $\mathcal{B}_{\mathbf{M}(:,2)}$ as Y_1 intersects with $\mathcal{Y}^-_{\mathbf{M}(:,2)}$ and entirely falls into $\mathcal{Y}^-_{\mathbf{M}(:,2)} \cup \mathcal{Y}^0_{\mathbf{M}(:,2)}$ (Fig.2e). Besides, $\boldsymbol{x}_6$ isn't used in $\mathcal{B}_{\mathbf{M}(:,2)}$ because $Y_6 = \{y_2\}$ entirely falls into $\mathcal{Y}^0_{\mathbf{M}(:,2)}$ but doesn't intersect with either one of $\mathcal{Y}^+_{\mathbf{M}(:,2)}$ or $\mathcal{Y}^-_{\mathbf{M}(:,2)}$ (Fig.2f), and $\boldsymbol{x}_4$ isn't used in $\mathcal{B}_{\mathbf{M}(:,4)}$ because $Y_4 = \{y_1, y_4\}$ intersects with both $\mathcal{Y}^+_{\mathbf{M}(:,4)}$ and $\mathcal{Y}^-_{\mathbf{M}(:,4)}$ (Fig.2g). As a result, the original PML training set $\mathcal{D}$ is transformed into four binary training sets as shown in Fig.2c.

It is worth noting that PAMB decomposes the PML problem into a number of binary classification problems according to the ternary coding matrix $\mathbf{M}$. The purpose of this ternary encoding scheme is to make a compromise between the *definiteness* and *adequacy* of the derived binary training sets. Specifically, for the randomly generated q-bits vector $\boldsymbol{v}$, if $z = 0$, we actually conduct binary encoding rather than ternary encoding. According to the ECOC en-

1. In this paper, the number of training examples in $\mathcal{B}_{\boldsymbol{v}}$ cannot be less than $\lceil 0.01m \rceil$, and the number of positive and negative training examples cannot be less than 5, respectively.

2. For this simple illustrative example, the aforementioned pre-set eligibility condition when generating binary training sets is not considered here.

Algorithm 1 The PAMB approach

```
Input:   $\mathcal{D}$: the PML training set $\{(\boldsymbol{x}_i, Y_i) \mid 1 \le i \le m\}(\boldsymbol{x}_i \in \mathcal{X}, Y_i \subseteq \mathcal{Y}, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{y_1, y_2, ..., y_q\})$;
         $z$: the number of '0's in each column of coding matrix $\mathbf{M}$;
         $L$: the ECOC codeword length;
         $\mathfrak{L}$: the employed binary classification algorithm;
         $\boldsymbol{x}_*$: the unseen instance;
Output:  $Y_*$: the predicted class label set for $\boldsymbol{x}_*$;
         $\hat{\boldsymbol{p}}_*$: the predicted possibility vector of all labels for $\boldsymbol{x}_*$;
1:  Initialize $\mathbf{M}$ as a zero matrix with size $q \times L$ and $l = 1$;
2:  while $l \le L$ do
3:      Randomly generate $\boldsymbol{v} \in \{+1, 0, -1\}^q$ with $z$ '0's and $\frac{q-z}{2}$ '+1's and '−1's;
4:      Divide the label space into $\mathcal{Y}_{\boldsymbol{v}}^+$, $\mathcal{Y}_{\boldsymbol{v}}^-$ and $\mathcal{Y}_{\boldsymbol{v}}^0$ according to Eq.(1);
5:      Construct a binary training set $\mathcal{B}_{\boldsymbol{v}}$ from the original training set $\mathcal{D}$ according to Eq.(2);
6:      if the pre-set eligibility condition for $\mathcal{B}_{\boldsymbol{v}}$ is true then
7:          Set $\boldsymbol{v}$ as the $l$-th column of coding matrix, i.e., $\mathbf{M}(:, l) = \boldsymbol{v}$, $\mathcal{B}_{\mathbf{M}(:,l)} = \mathcal{B}_{\boldsymbol{v}}$;
8:          Induce the binary classifier $h_l(\cdot)$ by invoking $\mathfrak{L}$ on $\mathcal{B}_{\mathbf{M}(:,l)}$: $h_l \leftarrow \mathfrak{L}(\mathcal{B}_{\mathbf{M}(:,l)})$;
9:          $l = l + 1$;
10:     end if
11: end while
12: Calculate the performance matrix $\mathbf{H}$ according to Eq.(3);
13: Calculate the weighted performance matrix $\hat{\mathbf{H}}$ according to Eq.(4);
14: Calculate the confidence vector $\boldsymbol{p}_*$ for unseen instance $\boldsymbol{x}_*$ according to Eq.(5);
15: Calculate the normalized confidence vector $\hat{\boldsymbol{p}}_*$ for unseen instance $\boldsymbol{x}_*$ according to Eq.(6);
16: Obtain the predicted label set $Y_*$ for unseen instance $\boldsymbol{x}_*$ according to Eq.(7);
17: Return $\hat{\boldsymbol{p}}_*$ and $Y_*$.
```

coding rule, examples whose candidate label sets entirely fall into $\mathcal{Y}_{\boldsymbol{v}}^+$ or $\mathcal{Y}_{\boldsymbol{v}}^-$ will be used as positive or negative examples while those examples whose candidate label sets intersect with both $\mathcal{Y}_{\boldsymbol{v}}^+$ and $\mathcal{Y}_{\boldsymbol{v}}^-$ will be discarded when generating $\mathcal{B}_{\boldsymbol{v}}$, thus there will be no indefiniteness for the belongingness of all examples in the derived binary training set $\mathcal{B}_{\boldsymbol{v}}$. Take Fig.2d as an example, regardless of whether y_1 or y_2 is an irrelevant label, $\boldsymbol{x}_1$ will be definitely used as a positive example by $\mathcal{B}_1$. However, this will lead to inadequate training examples in the binary training set $\mathcal{B}_{\boldsymbol{v}}$, especially when the average number of candidate labels (denoted by avg.#CLs) is relatively large to the number of labels (see Table 1 for an intuition) because the candidate label sets of many examples will intersect with both $\mathcal{Y}_{\boldsymbol{v}}^+$ and $\mathcal{Y}_{\boldsymbol{v}}^-$ (see Fig.2g for an intuition). As the value of z increases, more and more training examples can be used as positive or negative examples in $\mathcal{B}_{\boldsymbol{v}}$ because more candidate label sets of training examples can entirely fall into $\mathcal{Y}_{\boldsymbol{v}}^+ \cup \mathcal{Y}_{\boldsymbol{v}}^0$ or $\mathcal{Y}_{\boldsymbol{v}}^- \cup \mathcal{Y}_{\boldsymbol{v}}^0$ with a large $\mathcal{Y}_{\boldsymbol{v}}^0$. However, the *definiteness* of each example in $\mathcal{B}_{\boldsymbol{v}}$ will get worse because this operation will introduce disturbing examples whose ground-truth label set do not belong to $\mathcal{Y}_{\boldsymbol{v}}^+$ or $\mathcal{Y}_{\boldsymbol{v}}^-$, and the larger value of z means introducing more disturbing examples. Take Fig.2e as an example, if y_1 is an irrelevant label, $\boldsymbol{x}_1$ which should be discarded, will be wrongly used as a negative example in $\mathcal{B}_2$. When z reaches to its maximum value (i.e., $q-2$), we will obtain the best *adequacy* but worst *definiteness* for the derived binary training set $\mathcal{B}_{\boldsymbol{v}}$.

Therefore, it is necessary to employ the ternary decoding to deal with the challenging irrelevant labels and carefully tune the value of z to balance the *definiteness* and *adequacy* and achieve better generalization performance for PAMB. On one hand, to avoid deriving empty binary training sets, it is better to make the size of $\mathcal{Y}_{\boldsymbol{v}}^+ \cup \mathcal{Y}_{\boldsymbol{v}}^0$ or $\mathcal{Y}_{\boldsymbol{v}}^- \cup \mathcal{Y}_{\boldsymbol{v}}^0$ not less than avg.#CLs, which results the following empirical lower limit of z:

$$z = \begin{cases} \lceil 2 \times \text{avg.\#CLs} - q \rceil, & \text{avg.\#CLs} > \frac{q}{2} \\ 0, & \text{avg.\#CLs} \le \frac{q}{2} \end{cases}$$

On the other hand, the upper limit of z corresponds to $q-2$, i.e., both $\mathcal{Y}_{\boldsymbol{v}}^+$ and $\mathcal{Y}_{\boldsymbol{v}}^-$ only contain one class label. Note that this is only a reference range and a value randomly selected in this range does not necessarily mean better performance. In Subsection 4.5.1, we will further investigate how the performance of PAMB changes when value of z varies within this range.

3.3 Decoding Phase

In the decoding phase, PAMB makes prediction for unseen instance $\boldsymbol{x}_*$ based on the L binary classifiers h_l $(1 \le l \le L)$ induced in the encoding phase.

Specifically, the *loss-weighted decoding* [9] strategy is adapted for PML data which can take advantage of both empirical performance and predictive confidence of the induced binary classifiers. Based on the obtained coding matrix $\mathbf{M}$ and the corresponding binary classifiers h_l $(1 \le l \le L)$, PAMB generates a performance matrix $\mathbf{H} \in \mathbb{R}^{q \times L}$ where the (j, l)-th item $\mathbf{H}(j, l)$ is defined as follows:

$$\mathbf{H}(j,l) = \frac{1}{|\mathcal{D}_j|} \sum_{(\boldsymbol{x}_i, Y_i) \in \mathcal{D}_j} [\![h_l(\boldsymbol{x}_i) = \mathbf{M}(j,l)]\!] \qquad (3)$$

$$\text{where} \quad \mathcal{D}_j = \{(\boldsymbol{x}_i, Y_i) \mid y_j \in Y_i, 1 \le i \le m\}$$

Here, $|\cdot|$ returns the cardinality of a set and $[\![\pi]\!]$ returns 1 if predicate π holds and 0 otherwise. As shown in Eq.(3), $\mathcal{D}_j$ consists of all PML training examples whose candidate

TABLE 1
Characteristics of the experimental datasets. Here, *avg.#CLs* represents the average number of candidate labels per instance. *avg.#GLs* represents the average number of ground-truth labels per instance.

Data Set	#Examples	#Features	#Labels	avg.#CLs	avg.#GLs
music_emotion	6833	98	11	5.29	2.42
music_style	6839	98	10	6.04	1.44
mirflickr	10433	100	7	3.35	1.77
emotions	593	72	6	3,4,5	1.86
image	2000	294	5	2,3,4	1.23
yeast	2417	103	14	7,9,11,13	4.23
reference	5411	860	14	7,9,11,13	1.15
arts	4907	462	15	7,9,11,13	1.64
recreation	5000	606	22	11,13,15,17,19	1.42

label set contains class label y_j, so the value of $\mathbf{H}(j,l)$ corresponds to the proportion of examples in $\mathcal{D}_j$ whose binary prediction returned by h_l is consistent with the value of $\mathbf{M}(j,l)$. This value can also be regarded as the recall of classifier h_l if focusing on class label y_j. Under multi-label learning setting, for each class label, we should pay more attention on the examples associated with this label rather than the rest examples. This is the initial motivation why we use recall rather than other metrics (e.g., error rate) to weight the exponential loss. Note that binary classifier h_l can only return binary prediction '$+1$' or '-1', so $\mathbf{H}(j,l)=0$ when $\mathbf{M}(j,l)=0$. That is to say, the coding value '0' in $\mathbf{M}$ doesn't contribute to the final prediction.

To facilitate understanding, following the settings in the aforementioned illustrative example in Fig.2, take the calculating procedure of $\mathbf{H}(4,2)$ as an example, it is easy to know $\mathcal{D}_4=\{(\boldsymbol{x}_2,Y_2),(\boldsymbol{x}_4,Y_4),(\boldsymbol{x}_5,Y_5)\}$ because Y_2, Y_4 and Y_5 contain class label y_4. Moreover, suppose the binary predictions of examples in $\mathcal{D}_4$ returned by h_2 are $h_2(\boldsymbol{x}_2)=+1$, $h_2(\boldsymbol{x}_4)=-1$ and $h_2(\boldsymbol{x}_5)=+1$, respectively, then the value of $\mathbf{H}(4,2)$ equals 0.67 (i.e., 2/3) with $\mathbf{M}(4,2)=+1$.

To consider the relative performance of different binary classifiers, PAMB further normalizes each row of $\mathbf{H}$ and then obtains the following weighted performance matrix $\hat{\mathbf{H}}$:

$$\hat{\mathbf{H}}(j,l)=\frac{\mathbf{H}(j,l)}{\sum_{l=1}^{L}\mathbf{H}(j,l)},\quad (1\le j\le q, 1\le l\le L) \tag{4}$$

For unseen instance $\boldsymbol{x}_*$, PAMB calculates the *weighted exponential loss* to measure its closeness to each codeword (i.e., each row of $\mathbf{M}$). Existing works have shown that exponential loss can result in better performance as it enlarges the score between the predicted positive and negative instances by applying 'exponential' to the predictive confidence value:

$$p_j=-\sum_{l=1}^{L}\hat{\mathbf{H}}(j,l)\exp(-f_l(\boldsymbol{x}_*)\cdot\mathbf{M}(j,l)),\quad (1\le j\le q) \tag{5}$$

Let $\boldsymbol{p}_*=[p_1,p_2,\ldots,p_q]^\top$, each item can be regarded as the possibility of its corresponding label being a relevant label. To guarantee all possibility values for different scenarios in the same scale, we further conduct min-max normalization for $\boldsymbol{p}_*$ and then obtains the normalized possibility vector $\hat{\boldsymbol{p}}_*=[\hat{p}_1,\hat{p}_2,\ldots,\hat{p}_q]^\top$:

$$\hat{p}_j=\frac{p_j-\min(\boldsymbol{p}_*)}{\max(\boldsymbol{p}_*)-\min(\boldsymbol{p}_*)},\quad (1\le j\le q) \tag{6}$$

where $\min(\boldsymbol{p}_*)$ and $\max(\boldsymbol{p}_*)$ return the minimum item and maximum item in $\boldsymbol{p}_*$, respectively. As there can be multiple relevant labels for each PML instance, to identify the relevant label set Y_* for $\boldsymbol{x}_*$, we introduce a threshold τ to divide the whole label set into relevant part and irrelevant part:

$$Y_*=\{y_j \mid \hat{p}_j>\tau, 1\le j\le q\} \tag{7}$$

In this paper, τ is simply fixed as 0.9 with which PAMB generally achieves better performance.

Algorithm 1 summarizes the whole procedure of our proposed PAMB approach, including the encoding phase (Steps 1 to 11) and decoding phase (Steps 12 to 17). In the encoding phase, one q-bits column coding $\boldsymbol{v}$ is randomly generated based on the parameter z (Step 3). After dividing the label space (Step 4), a binary training set will be derived (Step 5). Then, if the training set is admissible (Step 6), $\boldsymbol{v}$ will be recorded as one column of coding matrix and discarded otherwise. Meanwhile, the corresponding binary training set $\mathcal{B}_{\boldsymbol{v}}$ will be accepted as $\mathcal{B}_{\mathbf{M}(:,l)}$, i.e., the binary training set for the l-th column of $\mathbf{M}$ (Step 7), over which a binary classifier will be induced (Step 8). In the decoding phase, a weighted performance matrix $\hat{\mathbf{H}}$ is calculated by using empirical performance (Steps 12 to 13). The normalized possibility vector $\hat{\boldsymbol{p}}$ is obtained based on $\hat{\mathbf{H}}$ and predictive possibility value of binary classifiers (Steps 14 to 15). Finally, the predicted class label set Y_* are obtained based on the threshold τ and the normalized possibility vector $\hat{\boldsymbol{p}}$ (Step 16). Both Y_* and $\hat{\boldsymbol{p}}$ are returned for performance evaluation (Step 17).

As shown in Algorithm 1, the proposed PAMB approach enables binary decomposition by adapting the ECOC techniques to solve the PML problem rather than attempts to identify valid labels within candidate label set via label confidence estimation. In the next section, extensive experiments will be conducted over both real-world and synthetic PML data sets, and the reported experimental results clearly show that PAMB achieves very competitive performance against other state-of-the-art baselines. Ablation study and parameter sensitivity analysis are also included to verify the effectiveness of PAMB's algorithmic design.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Benchmark Data Sets

To evaluate the generalization performance of our proposed PAMB approach, a total of nine data sets are employed for comparative studies in this paper. Specifically,

TABLE 2
Experimental results in terms of *hamming loss*, where the best performance (the smaller the better) is shown in boldface.

Data Set	avg.#CLs	PAMB	PARTICLE-VLS	PARTICLE-MAP	FPML	PML-LRS	ML-KNN	LIFT
music_emotion	5.29	**.207±.003**	.210±.005	.225±.003	.298±.006	.331±.001	.318±.008	.368±.007
music_style	6.04	**.116±.004**	.119±.003	.119±.004	.152±.004	.836±.002	.161±.005	.839±.005
mirflickr	3.35	**.164±.028**	.174±.031	.198±.084	.240±.066	.279±.018	.231±.059	.221±.058
emotions	3	.217±.027	**.200±.021**	.238±.033	.297±.020	.306±.003	.243±.024	.275±.038
	4	.214±.026	**.199±.024**	.240±.031	.305±.020	.653±.000	.302±.027	.620±.040
	5	**.259±.027**	.354±.031	.286±.020	.379±.020	.653±.000	.441±.038	.686±.028
image	2	**.175±.015**	.180±.014	.193±.012	.347±.011	.214±.004	.218±.016	.232±.008
	3	.207±.010	**.184±.012**	.222±.025	.366±.018	.698±.003	.262±.014	.721±.009
	4	**.240±.011**	.445±.018	.262±.027	.391±.018	.752±.000	.403±.020	.752±.005
yeast	7	.216±.009	.240±.013	.254±.015	.215±.010	.269±.003	**.207±.011**	.250±.011
	9	.208±.005	**.206±.010**	.247±.020	.219±.011	.633±.005	.212±.009	.624±.008
	11	.209±.007	**.197±.006**	.232±.008	.231±.009	.684±.000	.238±.006	.697±.012
	13	**.226±.008**	.686±.008	.317±.009	.270±.012	.702±.000	.331±.018	.697±.008
reference	7	.102±.004	**.086±.003**	.136±.007	.090±.003	.112±.001	.118±.005	.244±.027
	9	.217±.014	**.089±.003**	.126±.011	.090±.003	.152±.003	.137±.008	.916±.001
	11	.228±.021	.094±.003	.108±.007	**.090±.003**	.190±.006	.243±.028	.916±.001
	13	.310±.030	.907±.002	.317±.077	**.090±.003**	.232±.007	.400±.043	.917±.001
arts	7	**.121±.004**	.151±.003	.203±.034	.221±.008	.269±.001	.282±.013	.233±.007
	9	**.124±.003**	.153±.003	.202±.025	.265±.014	.587±.003	.309±.019	.890±.003
	11	**.145±.003**	.206±.003	.198±.031	.263±.009	.757±.005	.356±.035	.895±.002
	13	**.161±.005**	.335±.007	.226±.017	.430±.027	.822±.003	.450±.023	.895±.002
recreation	11	**.097±.005**	.113±.002	.144±.015	.200±.008	.345±.002	.285±.026	.348±.007
	13	**.099±.008**	.100±.002	.154±.046	.235±.018	.588±.006	.336±.030	.932±.002
	15	.101±.003	**.090±.002**	.142±.024	.234±.040	.739±.008	.318±.028	.934±.001
	17	.117±.005	**.102±.002**	.141±.021	.312±.044	.839±.009	.424±.031	.935±.001
	19	**.122±.004**	.344±.006	.177±.030	.298±.016	.871±.004	.417±.049	.934±.001

TABLE 3
Experimental results in terms of *ranking loss*, where the best performance (the smaller the better) is shown in boldface.

Data Set	avg.#CLs	PAMB	PARTICLE-VLS	PARTICLE-MAP	FPML	PML-LRS	ML-KNN	LIFT
music_emotion	5.29	**.233±.010**	.252±.009	.265±.012	.271±.010	.277±.001	.303±.011	.326±.011
music_style	6.04	**.134±.006**	.154±.006	.147±.003	.158±.006	.179±.003	.197±.009	.214±.010
mirflickr	3.35	.126±.039	.218±.029	.132±.125	**.117±.054**	.135±.004	.189±.032	.136±.039
emotions	3	**.161±.028**	.202±.031	.188±.037	.235±.019	.338±.009	.191±.020	.198±.042
	4	**.174±.028**	.197±.024	.198±.043	.249±.030	.455±.007	.244±.042	.248±.034
	5	**.232±.041**	.271±.033	.269±.046	.320±.035	.439±.005	.322±.052	.413±.052
image	2	**.167±.022**	.220±.020	.202±.022	.229±.015	.271±.009	.206±.027	.299±.025
	3	**.207±.015**	.240±.026	.223±.023	.236±.031	.294±.007	.240±.025	.362±.030
	4	**.255±.025**	.324±.032	.279±.027	.300±.014	.402±.015	.331±.021	.412±.022
yeast	7	**.174±.012**	.188±.014	.178±.010	.180±.012	.197±.001	.182±.011	.183±.011
	9	**.178±.015**	.186±.017	.187±.015	.189±.016	.214±.002	.194±.014	.195±.017
	11	**.177±.015**	.186±.010	.195±.008	.192±.009	.210±.002	.204±.010	.214±.006
	13	.213±.013	**.212±.014**	.298±.017	.218±.010	.215±.004	.231±.016	.271±.017
reference	7	.251±.011	.309±.009	.278±.013	**.247±.011**	.292±.010	.273±.015	.252±.013
	9	.262±.012	.289±.010	.266±.011	**.249±.010**	.319±.010	.281±.009	.256±.010
	11	.267±.010	.311±.014	.308±.019	**.266±.013**	.316±.008	.292±.011	.272±.014
	13	.285±.011	.305±.010	.350±.024	**.262±.012**	.360±.013	.291±.013	.266±.014
arts	7	**.197±.007**	.301±.010	.306±.024	.211±.009	.271±.003	.266±.008	.229±.008
	9	**.221±.010**	.323±.009	.378±.048	.232±.009	.288±.003	.280±.009	.263±.010
	11	**.239±.010**	.383±.014	.361±.048	.250±.009	.290±.005	.287±.011	.293±.008
	13	**.262±.009**	.402±.024	.566±.048	.284±.010	.300±.003	.316±.017	.293±.008
recreation	11	.186±.008	.314±.010	.400±.029	**.183±.008**	.216±.004	.239±.008	.210±.012
	13	**.193±.009**	.362±.012	.437±.027	.198±.010	.225±.002	.271±.011	.230±.011
	15	**.204±.014**	.312±.014	.358±.021	.208±.010	.235±.004	.262±.012	.256±.009
	17	.232±.009	.388±.013	.411±.036	**.220±.011**	.248±.007	.280±.010	.299±.018
	19	.242±.012	.334±.014	.474±.028	**.230±.008**	.241±.008	.290±.010	.351±.011

the experiments are conducted over three real-world PML data sets and twenty-three synthetic PML data sets which are generated from six multi-label data sets. The detailed characteristics of all data sets are summarized in Table 1, including *number of examples* (#Examples), *number of features* (#Features), *number of labels* (#Labels), *average number of candidate labels* (avg.#CLs), and *average number of ground-truth labels* (avg.#GLs).

For the three real-world PML datasets, including *music_emotion*, *music_style* and *mirflickr*, their candidate labels are tagged by common labelers and their ground truth labels are further examined by professional labelers. For the six multi-label data sets, including *emotions*, *image*, *yeast*, *reference*, *arts*, *recreation*, we generate several synthetic PML data sets from each of them by randomly choosing some irrelevant labels to form the candidate label sets with its ground truth labels. As shown in Table 1, each number in the column 'avg.#CLs' corresponds to one different setting with choosing different number of irrelevant labels, where twenty-three synthetic PML data sets are generated in total.

TABLE 4
Experimental results in terms of *one-error*, where the best performance (the smaller the better) is shown in boldface.

Data Set	avg.#CLs	PAMB	PARTICLE-VLS	PARTICLE-MAP	FPML	PML-LRS	ML-KNN	LIFT
music_emotion	5.29	**.403±.030**	.436±.023	.508±.024	.542±.027	.570±.002	.543±.026	.569±.028
music_style	6.04	**.333±.015**	.362±.020	.372±.017	.391±.015	.400±.000	.379±.014	.404±.019
mirflickr	3.35	.333±.085	**.148±.137**	.208±.256	.242±.146	.198±.013	.459±.167	.375±.072
emotions	3	**.253±.074**	.256±.071	.302±.089	.339±.050	.502±.009	.295±.058	.322±.071
	4	**.246±.072**	.256±.053	.300±.078	.344±.059	.508±.000	.314±.063	.397±.056
	5	**.346±.070**	.363±.067	.424±.080	.425±.061	.508±.000	.478±.097	.569±.071
image	2	**.299±.038**	.338±.029	.350±.035	.437±.023	.491±.012	.355±.036	.513±.029
	3	.387±.029	**.373±.026**	.391±.029	.440±.051	.530±.012	.407±.037	.600±.040
	4	.431±.039	**.379±.036**	.465±.041	.559±.041	.618±.035	.547±.033	.666±.037
yeast	7	.229±.026	.229±.036	.256±.037	.248±.041	**.198±.002**	.244±.030	.254±.035
	9	**.220±.023**	.243±.026	.250±.027	.249±.025	.290±.004	.252±.026	.255±.027
	11	**.221±.022**	.249±.024	.250±.027	.248±.021	.224±.011	.256±.029	.255±.026
	13	**.251±.034**	.255±.031	.478±.048	.253±.032	.264±.000	.255±.031	.276±.025
reference	7	.557±.018	**.545±.018**	.612±.017	.553±.021	.590±.007	.557±.020	.554±.021
	9	.578±.020	.557±.023	.574±.027	**.553±.021**	.590±.008	.557±.021	.554±.022
	11	.595±.022	**.552±.021**	.586±.028	.553±.021	.593±.007	.570±.022	.555±.022
	13	.620±.025	.565±.023	.685±.018	**.553±.021**	.594±.007	.569±.019	.558±.022
arts	7	**.511±.018**	.871±.012	.800±.106	.550±.014	.679±.038	.688±.024	.601±.013
	9	**.543±.022**	.874±.014	.898±.050	.622±.028	.732±.000	.717±.020	.671±.017
	11	**.587±.023**	.879±.014	.815±.069	.722±.023	.732±.000	.733±.023	.699±.023
	13	**.642±.012**	.737±.018	.916±.025	.739±.024	.760±.024	.762±.023	.699±.023
recreation	11	**.617±.016**	.767±.019	.929±.033	.622±.018	.798±.003	.769±.025	.724±.012
	13	**.644±.015**	.920±.009	.898±.042	.698±.021	.800±.000	.781±.013	.739±.018
	15	**.643±.017**	.793±.024	.878±.043	.717±.026	.800±.000	.794±.015	.763±.015
	17	**.670±.020**	.863±.011	.893±.027	.752±.020	.800±.000	.818±.017	.791±.019
	19	**.682±.017**	.767±.023	.930±.047	.770±.019	.800±.000	.857±.022	.839±.019

TABLE 5
Experimental results in terms of *coverage*, where the best performance (the smaller the better) is shown in boldface.

Data Set	avg.#CLs	PAMB	PARTICLE-VLS	PARTICLE-MAP	FPML	PML-LRS	ML-KNN	LIFT
music_emotion	5.29	**.405±.010**	.410±.010	.425±.012	.430±.010	.432±.001	.467±.010	.482±.010
music_style	6.04	**.194±.008**	.203±.009	.205±.007	.216±.009	.236±.004	.260±.011	.279±.012
mirflickr	3.35	.229±.082	.278±.062	.242±.064	**.225±.054**	.317±.003	.276±.068	.235±.058
emotions	3	**.301±.033**	.325±.027	.323±.036	.372±.038	.492±.008	.330±.033	.328±.043
	4	**.319±.047**	.322±.040	.341±.047	.387±.047	.600±.013	.386±.057	.381±.044
	5	**.363±.036**	.375±.035	.401±.053	.435±.042	.535±.007	.434±.035	.498±.038
image	2	**.187±.017**	.224±.017	.217±.019	.238±.014	.269±.006	.218±.021	.291±.019
	3	**.219±.016**	.232±.021	.233±.022	.242±.026	.289±.006	.245±.022	.340±.024
	4	**.257±.021**	.278±.023	.274±.022	.292±.010	.371±.012	.316±.017	.378±.013
yeast	7	.476±.017	**.457±.020**	.461±.016	.464±.019	.490±.002	.476±.015	.476±.016
	9	.481±.017	**.467±.017**	.492±.016	.481±.014	.486±.004	.498±.012	.491±.016
	11	.479±.019	**.464±.018**	.513±.015	.491±.017	.513±.007	.512±.022	.520±.015
	13	.535±.011	**.510±.010**	.590±.017	.540±.013	.515±.009	.554±.017	.607±.020
reference	7	.262±.011	.303±.009	.286±.013	**.259±.011**	.300±.009	.285±.015	**.259±.012**
	9	.274±.012	.279±.009	.277±.011	**.262±.010**	.331±.010	.291±.009	.268±.010
	11	**.279±.010**	.294±.013	.318±.018	**.279±.012**	.325±.008	.302±.009	.284±.015
	13	.295±.010	.290±.010	.355±.022	**.274±.010**	.365±.013	.302±.013	.278±.013
arts	7	**.260±.007**	.344±.007	.372±.019	.269±.009	.318±.004	.322±.009	.290±.006
	9	**.286±.008**	.368±.008	.434±.043	.290±.010	.334±.004	.335±.009	.323±.010
	11	**.303±.008**	.426±.012	.416±.046	.305±.007	.340±.005	.345±.011	.357±.010
	13	**.325±.009**	.442±.020	.600±.045	.332±.009	.342±.004	.369±.018	.357±.010
recreation	11	.232±.007	.346±.009	.439±.029	**.224±.011**	.251±.005	.281±.011	.253±.017
	13	**.240±.010**	.396±.011	.491±.023	.243±.011	.264±.003	.316±.010	.276±.011
	15	.251±.015	.349±.016	.400±.023	**.249±.010**	.279±.005	.307±.012	.299±.010
	17	.281±.010	.422±.013	.449±.034	**.260±.012**	.289±.008	.325±.009	.348±.018
	19	.283±.008	.366±.013	.509±.025	**.274±.009**	.283±.009	.336±.012	.399±.009

Besides, we also denote one specific synthetic PML data set by the concatenation of its data set name and the value of 'avg.#CLs', e.g., '*emotions*3' denotes the synthetic PML data set generated from *emotions* with the setting 'avg.#CLs $= 3$'.

In the following of this paper, we call the three real-world PML data sets and the first two multi-label data sets (i.e., *emotions* and *image*) as *sufficient* data sets because their number of training examples is larger than or close to the size of all possible label combinations (i.e., 2^q). In contrast, we call the rest four multi-label data sets (i.e., *yeast*, *reference*, *arts* and *recreation*) as *insufficient* data sets. Generally, sufficient data sets contain more *sufficient* supervisory information than insufficient data sets for model induction.

4.1.2 Evaluation Metrics

Since the final task of PML problem is to induce a multi-label predictor, six widely-used metrics for multi-label learning [13], [50], [53], including *hamming loss*, *ranking loss*, *one-error*, *coverage*, *average precision* and *micro*F1, are employed to evaluate the performance of PAMB and other state-of-

TABLE 6
Experimental results in terms of *average precision*, where the best performance (the larger the better) is shown in boldface.

Data Set	avg.#CLs	PAMB	PARTICLE-VLS	PARTICLE-MAP	FPML	PML-LRS	ML-KNN	LIFT
music_emotion	5.29	**.641±.013**	.622±.012	.589±.015	.574±.014	.564±.001	.555±.013	.533±.015
music_style	6.04	**.747±.009**	.724±.011	.720±.010	.704±.010	.683±.002	.686±.011	.662±.014
mirflickr	3.35	.781±.050	.681±.046	.807±.167	.812±.072	**.818±.006**	.686±.039	.764±.051
emotions	3	**.804±.041**	.787±.040	.773±.048	.739±.018	.652±.004	.772±.028	.767±.043
	4	**.801±.037**	.790±.028	.765±.041	.730±.030	.594±.004	.740±.038	.713±.034
	5	**.734±.044**	.712±.036	.693±.043	.674±.033	.613±.001	.652±.048	.584±.047
image	2	**.803±.024**	.766±.020	.767±.023	.720±.015	.679±.008	.765±.025	.664±.020
	3	**.752±.019**	.744±.021	.743±.021	.716±.033	.649±.007	.730±.024	.604±.026
	4	**.712±.026**	.682±.030	.693±.027	.641±.020	.582±.016	.636±.021	.557±.020
yeast	7	**.762±.014**	.751±.017	.745±.018	.742±.021	.744±.001	.745±.017	.741±.016
	9	**.756±.016**	.745±.017	.738±.018	.734±.017	.698±.002	.736±.015	.728±.019
	11	**.757±.017**	.745±.019	.723±.018	.732±.018	.721±.002	.721±.019	.706±.016
	13	.714±.022	**.723±.021**	.586±.018	.708±.017	.691±.003	.689±.020	.659±.020
reference	7	**.553±.013**	.542±.014	.509±.013	**.553±.014**	.517±.007	.539±.013	**.553±.014**
	9	.535±.015	.539±.016	.530±.017	**.552±.014**	.489±.008	.530±.015	.548±.016
	11	.524±.013	.541±.015	.511±.020	**.549±.013**	.511±.005	.522±.018	.543±.016
	13	.503±.016	.524±.015	.436±.011	.541±.017	.479±.009	.517±.015	**.543±.016**
arts	7	**.597±.012**	.368±.009	.394±.062	.572±.011	.479±.017	.478±.014	.537±.009
	9	**.569±.014**	.361±.009	.304±.055	.526±.015	.442±.006	.455±.014	.486±.009
	11	**.536±.012**	.302±.010	.356±.058	.474±.015	.443±.001	.448±.014	.456±.016
	13	**.497±.010**	.379±.015	.223±.026	.450±.017	.432±.008	.416±.017	.456±.016
recreation	11	**.516±.013**	.348±.014	.224±.031	.512±.012	.391±.008	.394±.018	.446±.011
	13	**.493±.012**	.253±.009	.224±.039	.468±.013	.382±.001	.374±.012	.424±.014
	15	**.491±.013**	.357±.016	.263±.029	.441±.012	.374±.003	.371±.010	.397±.009
	17	**.460±.015**	.266±.011	.235±.022	.417±.014	.366±.006	.347±.013	.359±.013
	19	**.444±.011**	.355±.016	.198±.038	.397±.013	.369±.008	.318±.010	.305±.013

TABLE 7
Experimental results in terms of *micro*F1, where the best performance (the larger the better) is shown in boldface.

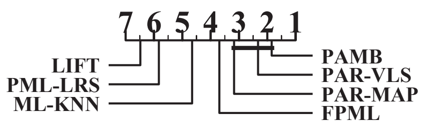
Data Set	avg.#CLs	PAMB	PARTICLE-VLS	PARTICLE-MAP	FPML	PML-LRS	ML-KNN	LIFT
music_emotion	5.29	.404±.011	.424±.013	.343±.015	**.485±.013**	.481±.001	.436±.012	.439±.012
music_style	6.04	**.589±.010**	.569±.013	.537±.014	.536±.013	.254±.001	.508±.014	.255±.005
mirflickr	3.35	**.721±.079**	.710±.096	.599±.138	.655±.133	.674±.010	.659±.125	.670±.128
emotions	3	.584±.047	.625±.051	.591±.075	.590±.029	.516±.006	.651±.042	**.646±.055**
	4	.611±.051	**.656±.041**	.567±.054	.579±.027	.516±.000	.589±.034	.498±.037
	5	.537±.054	**.597±.044**	.487±.053	.543±.036	.516±.000	.524±.044	.475±.031
image	2	**.633±.032**	.531±.033	.594±.021	.525±.013	.390±.005	.604±.028	.332±.037
	3	**.574±.020**	.560±.030	.553±.020	.524±.023	.404±.001	.549±.026	.404±.007
	4	**.527±.020**	.493±.017	.500±.033	.478±.014	.397±.000	.453±.017	.397±.006
yeast	7	.530±.021	.401±.060	.359±.096	.632±.016	.621±.003	.646±.016	**.647±.014**
	9	.583±.012	.573±.018	.399±.096	.625±.019	.491±.002	**.636±.015**	.489±.009
	11	.578±.015	**.644±.011**	.481±.013	.622±.018	.480±.000	.614±.014	.465±.015
	13	.523±.018	.468±.010	.338±.019	**.592±.018**	.459±.000	.558±.022	.465±.010
reference	7	.407±.017	.395±.021	.337±.015	**.415±.019**	.071±.001	.379±.011	.291±.014
	9	.298±.012	.399±.023	.350±.020	**.415±.019**	.085±.001	.346±.016	.153±.002
	11	.285±.017	.409±.018	.369±.026	**.415±.019**	.098±.003	.266±.015	.153±.002
	13	.240±.016	.153±.002	.207±.018	**.415±.019**	.109±.003	.216±.010	.153±.002
arts	7	**.401±.015**	.100±.011	.211±.083	.358±.012	.200±.003	.290±.009	.333±.012
	9	**.376±.013**	.104±.011	.116±.047	.324±.012	.184±.001	.276±.010	.189±.004
	11	**.356±.012**	.107±.010	.194±.041	.310±.014	.185±.001	.261±.013	.188±.004
	13	**.321±.010**	.209±.013	.099±.027	.255±.011	.186±.001	.240±.008	.188±.004
recreation	11	**.335±.015**	.189±.016	.096±.026	.283±.011	.137±.002	.213±.010	.220±.005
	13	**.316±.019**	.068±.011	.100±.022	.260±.011	.128±.001	.194±.007	.122±.002
	15	**.311±.013**	.197±.018	.127±.030	.251±.015	.127±.001	.198±.010	.122±.002
	17	**.287±.011**	.149±.016	.114±.018	.225±.008	.124±.001	.176±.008	.122±.002
	19	**.267±.013**	.176±.007	.088±.036	.223±.008	.124±.000	.174±.008	.122±.002

the-art baselines in this paper. For *hamming loss*, *ranking loss*, *one-error*, *coverage*, the smaller the value, the better the performance, while for *average precision* and *micro*F1, the larger the value, the better the performance.

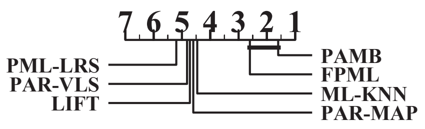
4.1.3 Comparing Approaches

In this paper, the performance of PAMB is compared with four state-of-the-art PML baselines, including PARTICLE (PARTICLE-VLS and PARTICLE-MAP) [11], FPML [43], PML-LRS [30], and two well-established multi-label baselines, including ML-KNN [49] and LIFT [47]:

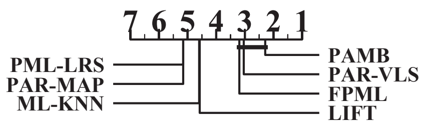
- PARTICLE, i.e., PARTIal multi-label learning via Credible Label Elicitation, works in a stage-wise mode, where credible labels are elicited from candidate label set in the first stage and then a tailored multi-label predictor is induced based on the obtained credible labels by *virtual label splitting* (VLS) or *maximum a posterior reasoning* (MAP).



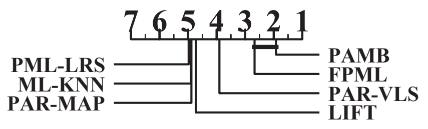
(a) hamming loss



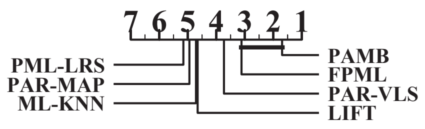
(b) ranking loss



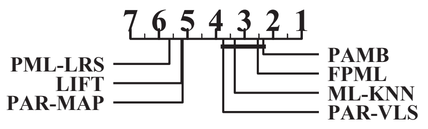
(c) one-error



(d) coverage

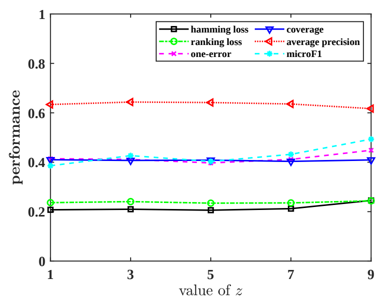


(e) average precision

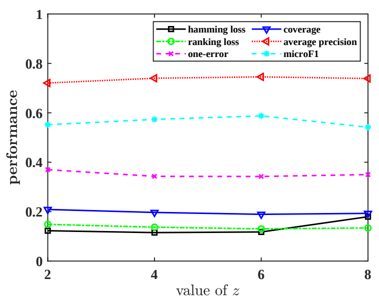


(f) microF1

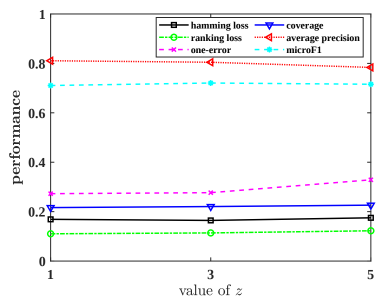
Fig. 3. Comparison results of PAMB (control approach) against other four PML and two MLL comparing approaches with the *Bonferroni-Dunn test*. Approaches not connected with PAMB by the thick horizontal line are considered to have significantly different performance with the control approach (CD = 2.1595 at 0.05 significance level).



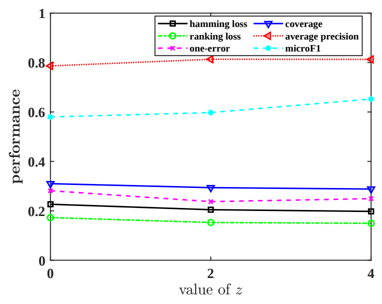
(a) music_emotion



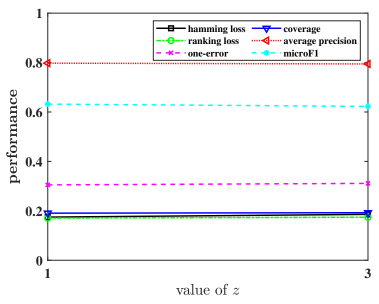
(b) music_style



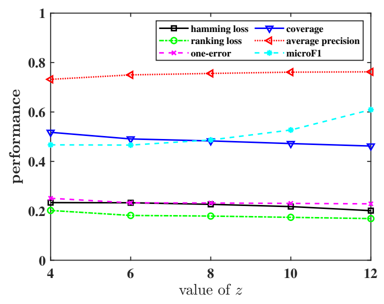
(c) mirflickr



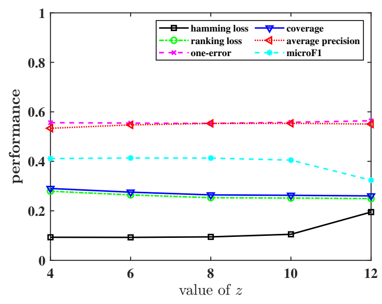
(d) emotions3



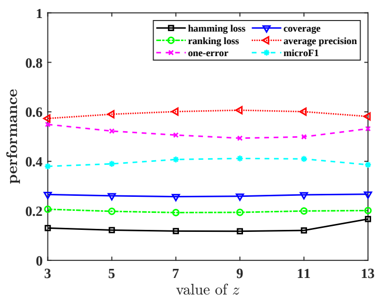
(e) image2



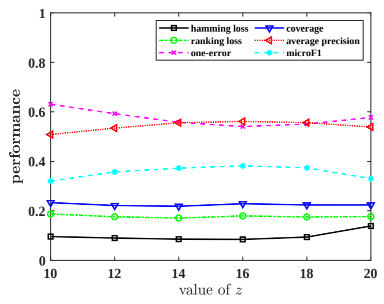
(f) yeast7



(g) reference7



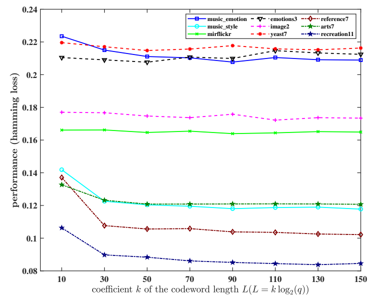
(h) arts7



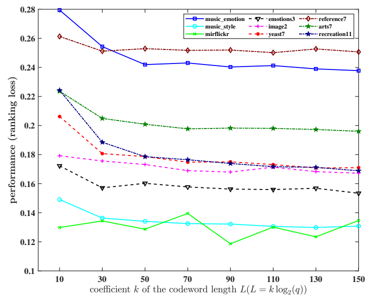
(i) recreation11

Fig. 4. The performance of PAMB changes as the value of z varies in some range over the three real-world PML data sets and six synthetic PML data sets.

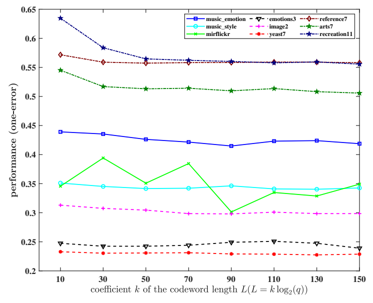
- FPML, i.e., Feature-induced Partial Multi-Label Learning, works in an iterative mode, where two optimization procedures are conducted iteratively to identify the ground-truth labels and train a multi-label predictor by leveraging a low-rank matrix approximation and dependencies between features and labels.
- PML-LRS, i.e., Partial Multi-Label Learning by Low-Rank and Sparse decomposition, also works in an iterative mode, where two optimization procedures



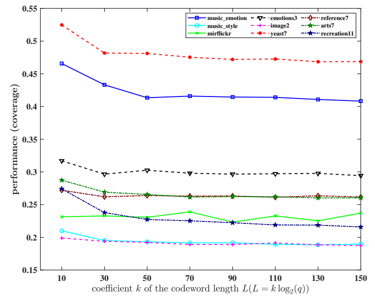
(a) hamming loss



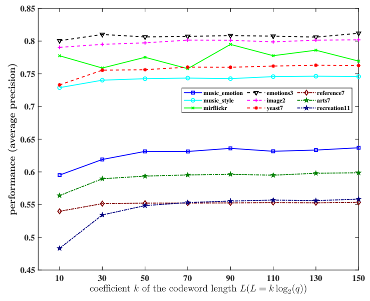
(b) ranking loss



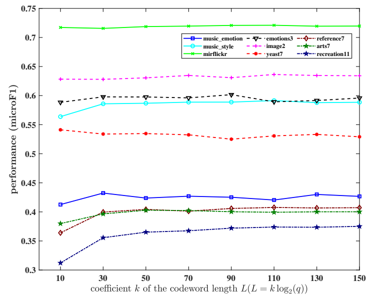
(c) one-error



(d) coverage



(e) average precision



(f) microF1

Fig. 5. The performance of PAMB changes as the coefficient k of codeword length L varies from the 10 to 150 over *music_emotion*, *music_style*, *mirflickr*, *emotions*3, *image*2, *yeast*7, *reference*7, *arts*7 and *recreation*11.

are conducted iteratively to recover the ground-truth label matrix from the observed one and induce the predictive model with low-rank and sparse decomposition strategy.

- ML-KNN, i.e., Multi-Label *k*-Nearest Neighbors, is a popular multi-label baseline and works by adapting *k*NN techniques for multi-label data, where the labels of unseen instance are predicted by maximum a posterior reasoning based on the counting statistics of its *k* nearest neighbors.
- LIFT, i.e., Label specIfic FeaTures for multi-label learning, is also a popular multi-label baseline and works by assuming existence of label-specific features for each label, where clustering analysis is performed on both positive and negative instances to generate label-specific features.

For all comparing approaches, their parameters are tuned according to their respective literatures. For the two multi-label baseline ML-KNN and LIFT, we regard all labels in candidate label set as ground-truth labels to induce the predictive models. For our proposed PAMB approach, L is set to $100\log_2(q)$, z is set to $\lfloor \text{avg.\#CLs} \rfloor$ for sufficient data sets and $\lceil \text{avg.\#CLs} + 2 \rceil$ for insufficient data sets. Moreover, the binary classification model in PAMB, PARTICLE, and LIFT, is induced with support vector machine (SVM) which is implemented by the popular Libsvm package [5].

We conduct *ten-fold cross validation* for all comparing approaches over each data set, and record both the mean value and standard derivation of each evaluation metric for performance comparison.

4.2 Experimental results

Tables 2 to 7 report the detailed experimental results of all comparing approaches in terms of six evaluation metrics respectively, where the best result over each data set in terms of each evaluation metric is emphasized in boldface.

In addition, *Friedman test* [7] is utilized to evaluate whether statistical difference exists in the generalization performance of all approaches. Let k be the number of comparing approaches, N be the number of employed data sets, and r_i^j be the rank of the jth approach over the ith data set, then the average rank of each approach can be given by $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j$ $(1 \leq j \leq q)$. Under the null hypothesis that all comparing approaches have the same generalization performance, the Friedman statistic F_F is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}, \quad \text{where}$$

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

Table 8 summarizes the Friedman statistics F_F in terms of each evaluation metrics and the critical value.[3] It is obvious that the null hypothesis is rejected in terms of all evaluation metrics at 0.05 significance level.

3. The critical value can be obtained by running the Matlab command icdf('F', $1-\alpha$, $k-1$, $(k-1)*(N-1)$) where α denotes the significance level.

TABLE 8
Summary of the Friedman statistics F_F in terms of six evaluation metrics and the critical value at 0.05 significance level for PAMB (#approaches $k = 7$, #data sets $N = 26$).

Evaluation Metric	F_F	Critical Value
Hamming loss	45.6958	
Ranking loss	17.6518	
One-error	11.3958	2.1595
Coverage	11.7056	
Average precision	12.5444	
MicroF1	16.6953	

As the results of *Friedman test* have told us that the performance of PAMB and other six state-of-the-art baselines is significantly different, to further evaluate whether PAMB achieves statistically superior or inferior performance against other baselines, we employ the *Bonferroni-Dunn test* [7] as a post-hoc test and regard PAMB as the control approach. Specifically, the performance of the control approach and other approaches is significantly different if the difference between their respective average ranks is no less than one critical difference (CD):

$$\mathrm{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

Here, q_α is a constant with specified significance level α and number of comparing approaches k, and $q_\alpha = 2.638$ when $\alpha = 0.05$ and $k = 7$ (cf. Table 5b in [7]), thus $\mathrm{CD} = 2.1595$.

Fig.3 shows the CD diagrams in terms of six evaluation metrics where PAMB is regarded as the control approach. In each diagram, the average rank of each approach R_j is marked on the axis with descending order from left to right. For the six comparing approaches, if the average rank of anyone differs from that of the control approach within one CD, then there will be a thick horizontal line connecting it and the control approach PAMB, which means that its performance is comparable with PAMB. Otherwise, the approach is considered to achieve significantly different performance against the control approach. Based on the reported experimental results, the following observations can be made:

- As shown in Tables 2 to 7, the proposed PAMB approach achieves the best performance in 98 cases among 156 configurations (26 data sets $\times$ 6 metrics). As shown in Fig.3, PAMB has the lowest average rank in terms of all the six evaluation metrics. Moreover, PAMB achieves statistically superior performance against each comparing approach in terms of at least one evaluation metric.
- The performance of PAMB is statistically superior against FPML in terms of *hamming loss* and comparable against FPML in terms of the rest five evaluation metrics. FPML utilizes the discriminative information in feature space to help estimate label confidence [43] and such idea can be introduced into PAMB to further improve its generalization performance in the future.

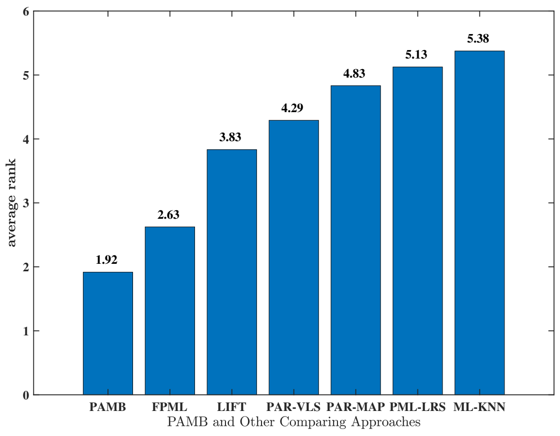


Fig. 6. The average ranks in terms of six evaluation metrics for PAMB and the six comparing approaches on the resampled *reference* dataset.

- The PML-LRS approach achieves poor performance in our experiments, where it has the highest average rank in terms of all the evaluation metrics except for *hamming loss*. Possible reason is that its sparse assumption for noisy labels [30] doesn't hold in the employed data sets. PAMB works without extra assumptions for data structure which is a desired algorithmic property in model induction.
- It is shown that PAMB usually achieves statistically superior performance against the two multi-label baselines (i.e., ML-KNN [49] and LIFT [47]), both of which directly regard all the labels in candidate label set as ground-truth ones. These experimental results not only show the superiority of our PAMB approach, but also suggest that it is better to specially design PML approaches with carefully utilizing the labeling information in candidate labels.

4.3 Ablation Study

In this paper, PAMB employs *weighted exponential loss* decoding (abbreviated as WeightedExp) scheme . To validate its effectiveness, we compared it with several popular decoding schemes, including hamming decoding [8] (abbreviated as Hamming), attenuated Euclidean decoding [25] (abbreviated as AttEuclidean), linear loss-based decoding (abbreviated as Linear) [1], exponential loss-based decoding (abbreviated as Exponential) [1], weighted linear loss-based decoding (abbreviated as WeightedLin) [9].

Table 9 reports the detailed experimental results where the best performance in terms of each evaluation metric over each data set is shown in bold face. It can be observed that the *weighted exponential loss* decoding scheme achieves best performance in most cases (33 out of 54 cases) which clearly validates the superiority of the decoding scheme in PAMB.

4.4 Limitation Analysis

As shown in Subsection 3.2, PAMB generates the coding matrix in a random manner. In other words, PAMB doesn't

TABLE 9
Experimental results among different decoding scheme on three real-world and six synthetic PML datasets, where the best performance is shown in boldface. Note that the *WeightedExp* is the decoding scheme in our PAMB approach.

Dataset	Evaluation Metrics	Hamming	AttEuclidean	Linear	Exponential	WeightedLin	WeightedExp
music_emotion	hamming loss	.211±.004	.211±.003	.211±.003	.211±.004	**.206±.003**	.207±.003
	ranking loss	.243±.007	.251±.009	.236±.008	.243±.010	.236±.009	**.234±.008**
	one-error	.439±.030	.441±.032	.439±.030	.437±.027	.405±.028	**.401±.028**
	coverage	.414±.010	.424±.015	.409±.010	.419±.014	.409±.011	**.406±.011**
	average precision	.621±.013	.616±.011	.624±.013	.621±.010	.637±.012	**.640±.012**
	microF1	.357±.015	.348±.014	.365±.013	.373±.012	.398±.016	**.405±.014**
music_style	hamming loss	.115±.004	.115±.004	**.114±.005**	**.114±.005**	.116±.004	.117±.004
	ranking loss	.141±.006	.144±.006	.137±.006	.139±.007	.135±.005	**.134±.005**
	one-error	.355±.019	.353±.020	.349±.021	.348±.020	.344±.016	**.336±.016**
	coverage	.200±.008	.202±.009	.197±.007	.200±.009	.195±.008	**.193±.009**
	average precision	.733±.010	.732±.011	.738±.012	.736±.012	.742±.009	**.746±.008**
	microF1	.545±.016	.539±.015	.557±.017	.560±.017	.573±.012	**.586±.011**
mirflickr	hamming loss	.218±.057	.225±.062	.214±.058	.214±.063	**.162±.029**	.165±.027
	ranking loss	.170±.050	.163±.056	.164±.055	.167±.056	**.120±.039**	.122±.038
	one-error	.497±.066	.448±.132	.458±.102	.457±.096	**.306±.099**	.316±.095
	coverage	.260±.087	.255±.088	.257±.091	.264±.095	**.224±.080**	.225±.080
	average precision	.706±.055	.719±.066	.722±.065	.716±.062	**.796±.052**	.790±.051
	microF1	.549±.131	.494±.158	.571±.114	.575±.141	.712±.071	**.720±.079**
emotions3	hamming loss	.228±.030	.231±.026	.226±.030	.225±.030	.223±.029	**.218±.032**
	ranking loss	.170±.026	.178±.031	.172±.029	.177±.027	.163±.028	**.161±.027**
	one-error	.264±.061	.268±.058	.264±.060	.278±.067	.254±.068	**.253±.069**
	coverage	.308±.040	.315±.042	.313±.041	.317±.037	.303±.037	**.301±.036**
	average precision	.799±.031	.793±.034	.794±.035	.788±.036	.802±.041	**.805±.040**
	microF1	.542±.055	.531±.047	.550±.057	.560±.055	.563±.047	**.584±.057**
image2	hamming loss	.176±.019	.175±.016	.173±.016	**.171±.016**	**.171±.016**	.172±.016
	ranking loss	.178±.020	.179±.022	.169±.019	.170±.021	**.168±.019**	**.168±.019**
	one-error	.299±.034	.298±.036	.305±.038	.302±.040	**.300±.037**	**.300±.037**
	coverage	.195±.018	.195±.017	**.189±.016**	.191±.016	**.189±.015**	**.189±.016**
	average precision	.798±.021	.798±.021	.799±.022	.799±.022	**.802±.021**	.801±.021
	microF1	.626±.036	.627±.033	.634±.032	.639±.033	**.637±.032**	**.637±.033**
yeast7	hamming loss	.245±.010	.255±.009	.241±.008	.232±.008	.232±.009	**.215±.007**
	ranking loss	.173±.014	.178±.015	**.171±.015**	.177±.015	.174±.012	.173±.013
	one-error	.249±.039	.249±.040	.244±.036	.246±.036	**.224±.028**	.228±.029
	coverage	**.460±.018**	.470±.017	.462±.019	.477±.018	.476±.016	.471±.018
	average precision	.754±.021	.751±.022	.756±.020	.751±.020	.761±.015	**.762±.016**
	microF1	.412±.041	.360±.021	.427±.028	.472±.015	.466±.018	**.534±.012**
reference7	hamming loss	.092±.003	**.091±.003**	.092±.003	.095±.003	.099±.003	.103±.007
	ranking loss	.249±.012	.254±.013	**.244±.011**	.250±.014	.250±.011	.249±.011
	one-error	.553±.020	.556±.019	**.552±.018**	.554±.018	.560±.020	.557±.020
	coverage	.258±.012	.261±.013	**.255±.012**	.261±.014	.261±.011	.260±.011
	average precision	.556±.012	.553±.012	**.558±.011**	.553±.013	.552±.012	.553±.013
	microF1	.414±.017	.413±.016	**.417±.018**	.411±.016	.406±.015	.404±.020
arts7	hamming loss	.120±.005	**.118±.004**	.121±.004	.128±.006	.119±.004	.121±.004
	ranking loss	.203±.008	.208±.010	.199±.007	.220±.018	.198±.008	**.197±.008**
	one-error	.538±.024	.539±.023	.531±.025	.559±.028	.512±.014	**.514±.016**
	coverage	.262±.008	.268±.008	**.260±.007**	.286±.017	.261±.008	.261±.007
	average precision	.582±.015	.578±.014	.586±.015	.562±.025	**.595±.009**	**.595±.010**
	microF1	.381±.019	.372±.016	.385±.015	.368±.022	.400±.012	**.401±.011**
recreation11	hamming loss	.095±.005	**.092±.005**	.096±.004	.100±.015	.095±.004	.096±.004
	ranking loss	.197±.010	.204±.014	.191±.010	.244±.031	.188±.011	**.187±.011**
	one-error	.679±.031	.679±.030	.669±.027	.733±.066	.622±.025	**.621±.028**
	coverage	.238±.011	.244±.015	.234±.011	.290±.030	.234±.013	**.233±.013**
	average precision	.472±.022	.469±.024	.481±.019	.417±.042	.512±.020	**.513±.021**
	microF1	.285±.026	.278±.022	.292±.020	.248±.044	.333±.019	**.334±.018**

specifically consider the characteristics of PML problems, e.g., positive/negative correlations between a pair of labels, labels with high/low frequency. Thus, PAMB might achieve suboptimal performance over some PML data sets with certain characteristics.

For example, as shown in Tables 2-7, PAMB achieves relatively inferior performance against comparing approaches over *reference* dataset. Possible reason is that the number of examples associated with 14th label is 20 times more than the number of examples associated with 1st label in *reference*. In other words, there exists extreme class-imbalance problem in *reference*. To verify this conjecture, we resample $1/3$ examples associated with the 14th label of *reference* dataset to make it more balanced. Fig.6 shows the average ranks in terms of six evaluation metrics for PAMB and the six baselines over *reference*(7,9,11,13). It can be observed that PAMB achieves the best performance over the resampled *reference* data set.

4.5 Parameter Sensitivity Analysis

As shown in Algorithm 1, PAMB has two parameters to be set, i.e., z (the number of '0's in each column of coding matrix $\mathbf{M}$) and L (the ECOC code length). In this subsection, we further investigate how the performance of PAMB changes with different values of these two parameters, respectively.

4.5.1 Analysis w.r.t. Parameter z

As discussed before, a proper value of z should be set for PAMB to make a better compromise between the *definiteness* and *adequacy* of the derived binary training sets. Fig.4 illustrates how the performance of PAMB changes when the value of z increases from one small value to $q-2$. It is shown that, the performance of PAMB indeed fluctuates as the value of z varies. Specifically, for sufficient data sets *music_emotion*, *music_style*, *mirflickr*, *emotions*3 and *image*2, PAMB usually achieves better performance with a relatively small z. For insufficient data set *yeast*7, *reference*7, *arts*7 and *recreation*11, PAMB usually achieves better performance with a relatively larger z. Possible reason is that, sufficient data sets have relatively enough training examples w.r.t their label space and a small z can make the derived binary training sets adequate. In contrast, insufficient data sets need a relatively large z to obtain adequate binary training sets. In this paper, we set z to $\lfloor$avg.#CLs$\rfloor$ for sufficient data sets and $\lceil$avg.#CLs $+ 2\rceil$ for insufficient data sets, respectively.

4.5.2 Analysis w.r.t. Parameter L

The codeword length L controls how many binary training sets are generated from the original PML data set. According to previous studies on ECOC [2], [8], [16], [20], [48], L serves as a crucial parameter for approaches which are designed based ECOC techniques. Fig.5 illustrates how the performance of PAMB changes when the value of L increases from $10\log_2(q)$ to $150\log_2(q)$. It is shown that, the performance of PAMB will be improved as the value of L increases at first, and then will become stable when the value of L is large enough. Specifically, for sufficient data sets *mirflickr*, *music_emotion*, *music_style*, *emotions*3 and *image*2, the performance of PAMB become stable when L is larger than $50\log_2(q)$ even though there are some fluctuations on *mirflickr*. For insufficient data set *yeast*7, *reference*7, *arts*7 and *recreation*11, the performance of PAMB become stable when L is larger than $100\log_2(q)$. In this paper, we simply set L to $100\log_2(q)$ for PAMB over all experimental data sets.

5 Conclusion

In this paper, the problem of learning from PML data is investigated, where each training example is associated with a set of candidate labels which are partially valid. The main contributions of this paper are two-fold: (1) Different from existing PML approaches which mainly focus on estimating the labeling confidence of each candidate label being the ground-truth label, we present an alternative strategy which solves the PML problem by enabling binary decomposition for handling PML training examples. (2) From the binary decomposition perspective, we propose a transformation-based PML learning approach named PAMB which works by transforming the original PML problem into a number of binary learning problems by adapting the popular ECOC techniques. Extensive experiments clearly validate the superiority of our proposed PAMB approach.

In the future, it is interesting to explore more effective binary decomposition methods for PML by utilizing prior feature topology and label correlations. Moreover, it is also worth to design some binary classifiers tailored for the derived binary training sets which can deal with the definiteness and adequacy in a better way.

References

[1] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research*, vol. 1, no. Dec, pp. 113–141, 2000.

[2] G. Armano, C. Chira, and N. Hatami, "Error-correcting output codes for multi-label text categorization," in *Proceedings of the 3rd Italian Information Retrieval Workshop*, Bari, Italy, 2012, pp. 26–37.

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

[4] N. Cao, T. Zhang, and H. Jin, "Partial multi-label optimal margin distribution machine," in *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, Virtual Event, 2021, pp. 2198–2204.

[5] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, 2011, Article 27.

[6] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *Journal of Machine Learning Research*, vol. 12, no. 42, pp. 1501–1536, 2011.

[7] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. Jan, pp. 1–30, 2006.

[8] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 263–286, 1995.

[9] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 120–134, 2010.

[10] S. Escalera, D. M. Tax, O. Pujol, P. Radeva, and R. P. Duin, "Subclass problem-dependent design for error-correcting output codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1041–1054, 2008.

[11] J.-P. Fang and M.-L. Zhang, "Partial multi-label learning via credible label elicitation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3587–3599, 2021.

[12] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[13] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, 2015, Article 52.

[14] S. Gopal and Y. Yang, "Multilabel classification with meta-level features," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Geneva, Switzerland, 2010, pp. 315–322.

[15] S. He, K. Deng, L. Li, S. Shu, and L. Liu, "Discriminatively relabel for partial multi-label learning," in *Proceedings of the IEEE International Conference on Data Mining*, Beijing, China, 2019, pp. 280–288.

[16] T. Kajdanowicz and P. Kazienko, "Multi-label classification using error correcting output codes," *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 4, pp. 829–840, 2012.

[17] T. Li, S. Gao, and Y. Xu, "Deep multi-similarity hashing for multi-label image retrieval," in *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*, Singapore, 2017, pp. 2159–2162.

[18] X.-C. Li, D.-C. Zhan, J.-Q. Yang, and Y. Shi, "Deep multiple instance selection," *Science China Information Sciences*, vol. 64, no. 3, 2021, Article 130102.

[19] Z. Li, G. Lyu, and S. Feng, "Partial multi-label learning via multi-subspace representation," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, Yokohama, Japan, 2020, pp. 2612–2618.

[20] G. Lin, K. Liu, B. Wang, and X. Zhang, "Partial label learning based on label distributions and error-correcting output codes," *Soft Computing*, vol. 25, no. 2, pp. 1049–1064, 2021.

[21] L. Liu and T. G. Dietterich, “A conditional multinomial mixture model for superset label learning,” in *Advances In Neural Information Processing Systems 25*, Lake Tahoe, NV, 2012, pp. 548–556.

[22] G. Lyu, S. Feng, and Y. Li, “Partial multi-label learning via probabilistic graph matching mechanism,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Virtual Event, CA, 2020, pp. 105–113.

[23] G. Lyu, S. Feng, and Y. Li, “Noisy label tolerance: A new perspective of partial multi-label learning,” *Information Science*, vol. 543, pp. 454–466, 2021.

[24] J. Nam, Y.-B. Kim, E. Loza-Mencía, S. Park, R. Sarikaya, and J. Fürnkranz”, “Learning context-dependent label permutations for multi-label classification,” in *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, CA, 2019, pp. 4733–4742.

[25] O. Pujol, S. Escalera, and P. Radeva, “An incremental node embedding technique for error correcting output codes,” *Pattern Recognition*, vol. 41, no. 2, pp. 713–725, 2008.

[26] O. Pujol, P. Radeva, and J. Vitria, “Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 1007–1012, 2006.

[27] F. Sun, J. Tang, H. Li, G.-J. Qi, and T. S. Huang, “Multi-label image categorization with sparse factor representation,” *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1028–1037, 2014.

[28] L. Sun, S. Feng, J. Liu, G. Lyu, and C. Lang, “Global-local label correlation for partial multi-label learning,” *IEEE Transactions on Multimedia*, 2021, in press.

[29] L. Sun, S. Feng, G. Lyu, H. Zhang, and G. Dai, “Partial multi-label learning with noisy side information,” *Knowledge and Information Systems*, vol. 63, no. 2, pp. 541–564, 2021.

[30] L. Sun, S. Feng, T. Wang, C. Lang, and Y. Jin, “Partial multi-label learning by low-rank and sparse decomposition,” in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, 2019, pp. 5016–5023.

[31] Y.-P. Sun and M.-L. Zhang, “Compositional metric learning for multi-label classification,” *Frontiers of Computer Science*, vol. 15, no. 5, 2021, Article 155320.

[32] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, “Discriminative and correlative partial multi-label learning,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, Macau, China, 2019, pp. 3691–3697.

[33] B. Wu, E. Zhong, A. Horner, and Q. Yang, “Music emotion recognition by multi-label multi-layer multi-instance multi-view learning,” in *Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, FL, 2014, pp. 117–126.

[34] X. Wu and M.-L. Zhang, “Towards enabling binary decomposition for partial label learning,” in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 2868–2874.

[35] M.-K. Xie and S.-J. Huang, “Partial multi-label learning,” in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018, pp. 4302–4309.

[36] M.-K. Xie and S.-J. Huang, “Partial multi-label learning with noisy label identification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, in press.

[37] M.-K. Xie, F. Sun, and S.-J. Huang, “Partial multi-label learning with meta disambiguation,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Virtual Event, 2021, pp. 1904–1912.

[38] M. Xu and L.-Z. Guo, “Learning from group supervision: The impact of supervision deficiency on multi-label learning,” *Science China Information Sciences*, vol. 64, no. 3, 2021, Article 130101.

[39] N. Xu, Y.-P. Liu, and X. Geng, “Partial multi-label learning with label distribution,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, 2020, pp. 6510–6517.

[40] Y. Yan and Y. Guo, “Adversarial partial multi-label learning with label disambiguation,” in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Virtual Event, 2021, pp. 10 568–10 576.

[41] Y. Yan, S. Li, and L. Feng, “Partial multi-label learning with mutual teaching,” *Knowledge-Based Systems*, vol. 212, 2021, Article 106624.

[42] B. Yang, J.-T. Sun, T. Wang, and Z. Chen, “Effective multi-label active learning for text classification,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 917–926.

[43] G. Yu, X. Chen, C. Domeniconi, J. Wang, Z. Li, Z. Zhang, and X. Wu, “Feature-induced partial multi-label learning,” in *Proceedings of the 18th IEEE International Conference on Data Mining*, Singapore, 2018, pp. 1398–1403.

[44] T. Yu, G. Yu, J. Wang, C. Domeniconi, and X. Zhang, “Partial multi-label learning using label compression,” in *Proceedings of the 20th IEEE International Conference on Data Mining*, Sorrento, Italy, 2020, pp. 761–770.

[45] T. Yu, G. Yu, J. Wang, and M. Guo, “Partial multi-label learning with label and feature collaboration,” in *Proceedings of the 25th International Conference on Database Systems for Advanced Applications*, Jeju, Korea, 2020, pp. 621–637.

[46] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, “Binary relevance for multi-label learning: An overview,” *Frontiers of Computer Science*, vol. 12, no. 2, pp. 191–202, 2018.

[47] M.-L. Zhang and L. Wu, “LIFT: Multi-label learning with label-specific features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2014.

[48] M.-L. Zhang, F. Yu, and C.-Z. Tang, “Disambiguation-free partial label learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2155–2167, 2017.

[49] M.-L. Zhang and Z.-H. Zhou, “ML-KNN: A lazy learning approach to multi-label learning,” *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[50] M.-L. Zhang and Z.-H. Zhou, “A review on multi-label learning algorithms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[51] F. Zheng and H. Xue, “Subclass maximum margin tree error correcting output codes,” in *Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence, August 28-31, 2018, Proceedings, Part I*, Nanjing, China, 2018, pp. 454–462.

[52] F. Zheng, H. Xue, X. Chen, and Y. Wang, “Maximum margin tree error correcting output codes,” in *Proceedings of the 14th Pacific Rim International Conference on Artificial Intelligence*, Phuket, Thailand, 2016, pp. 681–691.

[53] 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: Springer, 2017.

Bing-Qing Liu received the BSc degree in computer science from China University of Mining and Technology, China, in 2020. Currently, he is working toward the master’s degree at Southeast University, China. His main research interests include machine learning and data mining, especially in learning from multi-label data.

Bin-Bin Jia received the bachelor’s degree from North China Electric Power University in 2010, and the master’s degree from Beihang University in 2013. He joined the College of Electrical and Information Engineering, Lanzhou University of Technology in 2013 and is an assistant professor currently. From Sept. 2017 to Mar. 2022, he studied in the School of Computer Science and Engineering at Southeast University where he obtained the Ph.D. degree. His main research interests include machine learning and data mining, especially in multi-dimensional classification.

Min-Ling Zhang received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China, in 2001, 2004 and 2007, respectively. Currently, he is a Professor at the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining. In recent years, Dr. Zhang has served as the General Co-Chairs of ACML'18, Program Co-Chairs of PAKDD'19, CCF-ICAI'19, ACML'17, CCFAI'17, PRICAI'16, Senior PC member or Area Chair of AAAI 2017-2020, IJCAI 2017-2022, KDD 2021-2022, ICDM 2015-2022, etc. He is also on the editorial board of IEEE Transactions on Pattern Analysis and Machine Intelligence, ACM Transactions on Intelligent Systems and Technology, Science China Information Sciences, Frontiers of Computer Science, Machine Intelligence Research, etc. Dr. Zhang is the Steering Committee Member of ACML and PAKDD, Vice Chair of the CAAI Machine Learning Society, standing committee member of the CCF Artificial Intelligence & Pattern Recognition Society. He is a Distinguished Member of CCF, CAAI, and Senior Member of AAAI, ACM, IEEE.