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 \( \mathcal{D} = \{(x_i, Y_i) \mid (1 \leq i \leq m)\} \) where \( x_i \in \mathcal{X} \) is a \( d \)-dimensional instance and \( Y_i \subseteq \mathcal{Y} \) is the candidate label set associated with \( x_i \), it is assumed that the ground-truth label set \( \hat{Y}_i \) for \( x_i \) resides in its candidate label set \( Y_i \), i.e. \( \hat{Y}_i \subseteq Y_i \). Accordingly, the task of partial multi-label learning is to induce a multi-label predictor \( f: \mathcal{X} \rightarrow 2^\mathcal{Y} \) 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.
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 labeling 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 \( M \) to determine how the class space is partitioned. The two commonly used designs of coding matrix \( M \), i.e., binary coding [8] (i.e., \( +1 \) and \( -1 \)), ternary coding [1] (i.e., \( \pm 1 \)). For the coding matrix, its \( j \)-th row \( M(j,:) \) represents a unique \( L \)-bits codeword for the \( j \)-th class label \( c_j \) (\( 1 \leq j \leq q \)), and its \( l \)-th column \( M(:, l) \) partitions the class space \( \mathcal{Y} \) into two or three parts: positive part \( \mathcal{Y}_j^+ \), negative part \( \mathcal{Y}_j^- \) and neutral part \( \mathcal{Y}_j^0 \) which only exists for ternary coding:

\[
\mathcal{Y}_j^+ = \{c_j | M(j, l) = +1, 1 \leq j \leq q\}
\]

\[
\mathcal{Y}_j^- = \{c_j | M(j, l) = -1, 1 \leq j \leq q\}
\]

\[
\mathcal{Y}_j^0 = \{c_j | M(j, l) = 0, 1 \leq j \leq q\}
\]

According to the \( l \)-th column of \( M \), a binary classification data set can be constructed where the instances associated with the label in \( \mathcal{Y}_j^+ \) (or \( \mathcal{Y}_j^- \)) are used as positive (or negative) samples, respectively, and the remaining instances associated with the label in \( \mathcal{Y}_j^0 \) will be disregarded. 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 \( M \).

In the decoding phase, the main task is to obtain the final prediction of unseen instance \( x_* \) according to the outputs of the \( L \) learned binary classifiers:

\[
h(x_*) = [h_1(x_*), h_2(x_*), \ldots, h_L(x_*)]^	op
\]

Generally, the final prediction for \( x_* \) is determined as follows:

\[
f(x_*) = c_{\hat{j}}, \text{where } \hat{j} = \arg \min_{1 \leq j \leq q} \text{dist}(h(x_*), M(j,:))
\]

Here, \( \text{dist}(\cdot, \cdot) \) is some kind of distance function (e.g., hamming distance). In other words, the class whose codeword is closest to the predicted vector \( h(x_*) \) 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., DE-COC [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 \( \text{dist}(\cdot, \cdot) \). 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 \( D \) w.r.t. each column of the ternary coding matrix \( M \).

Specifically, let \( v \in \{+1, 0, -1\}^q \) 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 \( v \), the label space can be divided into positive groups \( \mathcal{Y}_v^+ \), negative group \( \mathcal{Y}_v^- \) and neutral group \( \mathcal{Y}_v^0 \):

\[
\mathcal{Y}_v^+ = \{v_j | v_j = +1, 1 \leq j \leq q\}
\]

\[
\mathcal{Y}_v^- = \{v_j | v_j = -1, 1 \leq j \leq q\}
\]

\[
\mathcal{Y}_v^0 = \{v_j | v_j = 0, 1 \leq j \leq q\}
\]

where \( v_j \) denotes the \( j \)-th item in \( v \). 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 \( B_v \) according to the divided label space by ternary vector \( v \) in Eq.(1). Specifically, \( x_* \) is used as a positive (or negative) example when the following condition \( C_v^+ \) (or \( C_v^- \)) is true:

\[
C_v^+ = (Y_i \cap \mathcal{Y}_v^+ \neq \emptyset) \land (Y_i \cap \mathcal{Y}_v^- = \emptyset)
\]

\[
C_v^- = (Y_i \cap \mathcal{Y}_v^- = \emptyset) \land (Y_i \cap \mathcal{Y}_v^+ \neq \emptyset)
\]

In other words, if \( Y_i \) entirely falls into the union of \( \mathcal{Y}_v^+ \) (or \( \mathcal{Y}_v^- \)) and \( \mathcal{Y}_v^0 \) (i.e., \( Y_i \cap \mathcal{Y}_v^+ = \emptyset \) or \( Y_i \cap \mathcal{Y}_v^- = \emptyset \)) and its intersection with \( \mathcal{Y}_v^- \) (or \( \mathcal{Y}_v^+ \)) is not empty (i.e., \( Y_i \cap \mathcal{Y}_v^- \neq \emptyset \) or \( Y_i \cap \mathcal{Y}_v^+ \neq \emptyset \)), then \( x_* \) is used as a positive (or negative)
example in $B_i$. Formally speaking, $B_i$ is constructed as follows:

$$B_i = \{(x_i, +1) \mid C_i^+ \text{ true, } 1 \leq i \leq m\} \cup \{(x_i, -1) \mid C_i^- \text{ true, } 1 \leq i \leq m\}$$

As shown in Eq.(2), not all PML training examples will contribute to construct the binary training set $B_i$. To guarantee informative training set with enough samples, PAMB discards the current randomly generated ternary vector $v$ when the corresponding $B_i$ does not meet the pre-set eligibility condition.\(^1\) If $B_i$ is admissible, PAMB records $v$ as one column of the coding matrix $M$, i.e., $M(:, l) = v$ where $M(:, l)$ denotes $M$’s $l$-th column ($1 \leq l \leq L$). After $M$ is obtained, PAMB induces a binary classifier $h_l(\cdot) = \text{sign}(f_l(\cdot))$ by invoking the employed binary classification algorithm $\mathcal{L}$ on $B_{M(:, l)}$: $h_l \leftarrow \mathcal{L}(B_{M(:, l)})$. Here, $B_{M(:, l)}$ is the binary training set which is constructed according to the divided label space by ternary vector $M(:, l)$ in Eq.(1), $h_l(x) \in \{+1, -1\}$ returns the predicted binary label for instance $x$ while $f_l(x) \in \mathbb{R}$ returns the predicted confidence for instance $x$, 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 $v$ will be accepted as one column of $M$. In other words, there is no necessary correspondence between $v$ and the columns of $M$. Therefore, we cannot use $v_l$ to represent the $l$-th column of $M$, i.e., $M(:, l)$. 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 matrix $M$ (Fig.2a) is 2, 2, 1, respectively. The matrix shown in Fig.2b corresponds to the label matrix of the PML training set $D = \{(x_i, Y_i) \mid 1 \leq i \leq 6\}$ with six examples, where each column corresponds to one label and each row 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 $(x_1, Y_1)$ as an example, because $Y_1 = \{y_1, y_2\}$, $x_1$ is used as a positive example in $B_{M(:, 1)}$ as $Y_1$ entirely falls into $\gamma_{M(:, 1)}^+$ (Fig.2d), and is also used as a negative example in $B_{M(:, 2)}$ as $Y_1$ intersects with $\gamma_{M(:, 2)}^-$ (Fig.2e). Besides, $x_6$ isn’t used in $B_{M(:, 2)}$ because $Y_6 = \{y_1, y_4\}$ entirely falls into $\gamma_{M(:, 2)}^{-}$, but doesn’t intersect with either one of $\gamma_{M(:, 2)}^+$ or $\gamma_{M(:, 2)}^-$ (Fig.2f), and $x_4$ isn’t used in $B_{M(:, 4)}$ because $Y_4 = \{y_1, y_4\}$ intersects with both $\gamma_{M(:, 4)}^+$ and $\gamma_{M(:, 4)}$ (Fig.2g). As a result, the original PML training set $D$ 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 $M$. 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 $v$, if $z = 0$, we actually conduct binary encoding rather than ternary encoding. According to the ECOC en-
one hand, to avoid deriving empty binary training sets, it is do not belong to \( Y \) introduce disturbing examples whose ground-truth label set example in \( B \) or negative examples in more and more training examples can be used as positive \( Y \) especially when the average number of candidate labels derived binary training set will obtain the best adequacy \( B \) example, if introducing more disturbing examples. Take Fig.2e as an example, regardless of whether \( y \) and achieve better generalization performance for \( P \) \( Y \) \( \sum \) \( \text{avg.#CLs} \), \( \text{avg.#CLs} \leq \frac{q}{2} \) or \( q-2 \), i.e., both \( Y_v^+ \) and \( Y_v^- \) 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 \( x \), based on the \( L \) binary classifiers \( h_l \) \( (1 \leq l \leq L) \) 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 \( M \) and the corresponding binary classifiers \( h_l \) \( (1 \leq l \leq L) \), PAMB generates a performance matrix \( H \in \mathbb{R}^{q \times L} \), where the \((j, l)\)-th item \( H(j, l) \) is defined as follows:

\[
H(j, l) = \frac{1}{|D_j|} \sum_{(x_i, Y_i) \in D_j} \| h_l(x_i) = M(j, l) \| \quad (3)
\]

where \( D_j \) consists of all PML training examples whose candidate

\[
\text{card}(\pi) \quad |\pi| \rightarrow \frac{1}{|D_j|} \sum_{(x_i, Y_i) \in D_j} \| h_l(x_i) = M(j, l) \| \quad (3)
\]

Here, \(| \cdot | \) returns the cardinality of a set and \( \| \pi \| \) returns 1 if predicate \( \pi \) holds and 0 otherwise. As shown in Eq.(3), \( D_j \) consists of all PML training examples whose candidate
Multiple relevant labels for each PML instance, to identify the relevant label set $Y_r$ for $x_s$, we introduce a threshold $\tau$ to divide the whole label set into relevant part and irrelevant part:

$$Y_r = \{y_j \mid \hat{p}_j > \tau, 1 \leq j \leq q\}$$  

In this paper, $\tau$ 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 $v$ 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), $v$ will be recorded as one column of coding matrix and discarded otherwise. Meanwhile, the corresponding binary training set $B_v$ will be accepted as $B_v(M(:,:), l, i.e., the binary training set for the $l$-th column of $M$ (Step 7), over which a binary classifier will be induced (Step 8). In the decoding phase, a weighted performance matrix $H$ is calculated by using empirical performance (Steps 12 to 13). The normalized possibility vector $\hat{p}$ is obtained based on $H$ and predictive possibility value of binary classifiers (Steps 14 to 15). Finally, the predicted class label set $Y_r$ are obtained based on the threshold $\tau$ and the normalized possibility vector $\hat{p}$ (Step 16). Both $Y_r$ and $\hat{p}$ 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.

### 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,
the experiments are conducted on 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 mirflckr, 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.

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

<table>
<thead>
<tr>
<th>Data Set</th>
<th>avg.#CLs</th>
<th>PAMB</th>
<th>PARTICLE-VLS</th>
<th>PARTICLE-MAP</th>
<th>FPML</th>
<th>PMI-LRS</th>
<th>ML-KNN</th>
<th>LIFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>music_emotion</td>
<td>3.29</td>
<td>.207±.003</td>
<td>.210±.003</td>
<td>.223±.003</td>
<td>.298±.006</td>
<td>.331±.001</td>
<td>.318±.008</td>
<td>.368±.007</td>
</tr>
<tr>
<td>music_style</td>
<td>6.04</td>
<td>.119±.004</td>
<td>.119±.004</td>
<td>.152±.004</td>
<td>.186±.002</td>
<td>.208±.005</td>
<td>.219±.005</td>
<td>.267±.007</td>
</tr>
<tr>
<td>mirflckr</td>
<td>3.35</td>
<td>.164±.028</td>
<td>.174±.031</td>
<td>.198±.04</td>
<td>.240±.066</td>
<td>.279±.018</td>
<td>.231±.059</td>
<td>.221±.058</td>
</tr>
<tr>
<td>emotions</td>
<td>3</td>
<td>.217±.027</td>
<td>.200±.021</td>
<td>.238±.033</td>
<td>.297±.026</td>
<td>.306±.003</td>
<td>.243±.024</td>
<td>.275±.038</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.214±.026</td>
<td>.199±.024</td>
<td>.240±.031</td>
<td>.305±.020</td>
<td>.653±.000</td>
<td>.302±.027</td>
<td>.620±.040</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.259±.027</td>
<td>.354±.031</td>
<td>.286±.020</td>
<td>.379±.020</td>
<td>.653±.000</td>
<td>441±.038</td>
<td>.686±.028</td>
</tr>
<tr>
<td>image</td>
<td>2</td>
<td>.175±.015</td>
<td>.180±.014</td>
<td>.193±.012</td>
<td>.347±.011</td>
<td>.214±.004</td>
<td>.218±.016</td>
<td>.222±.008</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.207±.010</td>
<td>.184±.012</td>
<td>.222±.025</td>
<td>.366±.018</td>
<td>.698±.003</td>
<td>.262±.014</td>
<td>.721±.009</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.240±.011</td>
<td>.391±.018</td>
<td>.262±.007</td>
<td>.752±.006</td>
<td>.203±.020</td>
<td>.203±.020</td>
<td>.368±.009</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.216±.009</td>
<td>.240±.013</td>
<td>.254±.015</td>
<td>.215±.010</td>
<td>.269±.003</td>
<td>.207±.011</td>
<td>.253±.011</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>.209±.007</td>
<td>.197±.006</td>
<td>.232±.008</td>
<td>.231±.009</td>
<td>.684±.000</td>
<td>.238±.006</td>
<td>.697±.012</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.226±.008</td>
<td>.686±.008</td>
<td>.317±.009</td>
<td>.270±.012</td>
<td>.702±.000</td>
<td>.331±.018</td>
<td>.697±.008</td>
</tr>
<tr>
<td>reference</td>
<td>7</td>
<td>.102±.004</td>
<td>.086±.003</td>
<td>.136±.007</td>
<td>.090±.003</td>
<td>.112±.001</td>
<td>.118±.005</td>
<td>.244±.027</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>.217±.014</td>
<td>.089±.003</td>
<td>.126±.005</td>
<td>.090±.003</td>
<td>.152±.003</td>
<td>.137±.008</td>
<td>.916±.001</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.228±.021</td>
<td>.094±.003</td>
<td>108±.007</td>
<td>.090±.003</td>
<td>.190±.006</td>
<td>.243±.028</td>
<td>.916±.001</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.310±.030</td>
<td>.907±.002</td>
<td>.317±.007</td>
<td>.090±.003</td>
<td>.232±.007</td>
<td>.400±.043</td>
<td>.917±.001</td>
</tr>
<tr>
<td>arts</td>
<td>7</td>
<td>.121±.004</td>
<td>.151±.003</td>
<td>.203±.004</td>
<td>.221±.008</td>
<td>.269±.001</td>
<td>.282±.013</td>
<td>.233±.007</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>.124±.003</td>
<td>.153±.003</td>
<td>.202±.025</td>
<td>.265±.014</td>
<td>.587±.003</td>
<td>.309±.019</td>
<td>.890±.003</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.145±.003</td>
<td>.206±.003</td>
<td>.198±.031</td>
<td>.263±.009</td>
<td>.757±.005</td>
<td>.356±.035</td>
<td>.895±.002</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.161±.005</td>
<td>.335±.007</td>
<td>.226±.017</td>
<td>.430±.027</td>
<td>.822±.003</td>
<td>.450±.023</td>
<td>.895±.002</td>
</tr>
<tr>
<td>recreation</td>
<td>7</td>
<td>.097±.005</td>
<td>.113±.002</td>
<td>.144±.015</td>
<td>.200±.008</td>
<td>.345±.002</td>
<td>.285±.026</td>
<td>.348±.007</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>.099±.008</td>
<td>.100±.002</td>
<td>.194±.046</td>
<td>.235±.018</td>
<td>.588±.006</td>
<td>.336±.030</td>
<td>.932±.002</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.101±.003</td>
<td>.090±.002</td>
<td>.142±.024</td>
<td>.234±.040</td>
<td>.739±.008</td>
<td>.318±.028</td>
<td>.934±.001</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>.117±.005</td>
<td>.102±.002</td>
<td>.141±.021</td>
<td>.312±.044</td>
<td>.839±.009</td>
<td>.244±.031</td>
<td>.935±.001</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>.122±.004</td>
<td>.344±.006</td>
<td>.177±.006</td>
<td>.298±.016</td>
<td>.871±.004</td>
<td>.417±.049</td>
<td>.934±.001</td>
</tr>
</tbody>
</table>

### Table 3
Experimental results in terms of **rankin...
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., ‘emotions3’ 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., 2^15). 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 supervision 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 microF1, are employed to evaluate the performance of PAMB and other state-of-the-art methods.
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 microF1, 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 posteriori reasoning (MAP).
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
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 kNN 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 [avg.#CLs] for sufficient data sets and [avg.#CLs + 2] 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 $r_{ij}$ be the rank of the $j$th approach over the $i$th data set, then the average rank of each approach can be given by $R_j = \frac{1}{N} \sum_{i=1}^{N} r_{ij}$ ($1 \leq j \leq q$). 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:

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

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

Table 8 summarizes the Friedman statistics $F_F$ in terms of each evaluation metrics and the critical value. 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 - $\alpha$, $k - 1$, $(k - 1)\times(N - 1)$) where $\alpha$ denotes the significance level.
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):

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

Here, $q_\alpha$ is a constant with specified significance level $\alpha$ and number of comparing approaches $k$, and $q_\alpha = 2.638$ when $\alpha = 0.05$ and $k = 7$ (cf. Table 5b in [7]), thus 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 any one 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 × 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.

### 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 loss | ranking loss | one-error | coverage | average precision | microF1 | hamming loss | ranking loss | one-error | coverage | average precision | microF1 | hamming loss | ranking loss | one-error | coverage | average precision | microF1 | hamming loss | ranking loss | one-error | coverage | average precision | microF1 | hamming loss | ranking loss | one-error | coverage | average precision | microF1 | hamming loss | ranking loss | one-error | coverage | average precision | microF1 | hamming loss | ranking loss | one-error | coverage | average precision | microF1 |
|---------|---------------------|--------------|--------------|----------|----------|-------------------|--------|--------------|--------------|----------|----------|-------------------|--------|--------------|--------------|----------|----------|-------------------|--------|--------------|--------------|----------|----------|-------------------|--------|--------------|--------------|----------|----------|-------------------|--------|--------------|--------------|----------|----------|-------------------|--------|
| music_emotion | Exponential        | 211±.004     | 251±.009     | 236±.008 | 439±.030 | 414±.010          | 621±.013 | 357±.015     | 348±.014     | 365±.013 | 733±.010 | 733±.011         | 545±.016 | 211±.004     | 251±.009     | 236±.008 | 439±.030 | 414±.010          | 621±.013 | 357±.015     | 348±.014     | 365±.013 | 733±.010 | 733±.011         | 545±.016 |
| music_style  | AttEuclidean       | 211±.005     | 251±.009     | 236±.008 | 439±.030 | 414±.010          | 621±.013 | 357±.015     | 348±.014     | 365±.013 | 733±.010 | 733±.011         | 545±.016 | 211±.005     | 251±.009     | 236±.008 | 439±.030 | 414±.010          | 621±.013 | 357±.015     | 348±.014     | 365±.013 | 733±.010 | 733±.011         | 545±.016 |

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 in 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’ in each column of coding ma-
trix 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, emotions3 and image2, PAMB usually achieves better performance with a relatively small z. For insufficient data set yeast7, reference7, arts7 and recreation11, 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 \([\text{avg.#CLS}] \) for sufficient data sets and \([\text{avg.#CLS} + 2]\) 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, emotions3 and image2, 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 yeast7, reference7, arts7 and recreation11, 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


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.