Partial Multi-Label Learning With Noisy Label Identification

Ming-Kun Xie and Sheng-Jun Huang

Abstract—Partial multi-label learning (PML) deals with problems where each instance is assigned with a candidate label set, which contains multiple relevant labels and some noisy labels. Recent studies usually solve PML problems with the disambiguation strategy, which recovers ground-truth labels from the candidate label set by simply assuming that the noisy labels are generated randomly. In real applications, however, noisy labels are usually caused by some ambiguous contents of the example. Based on this observation, we propose a partial multi-label learning approach to simultaneously recover the ground-truth information and identify the noisy labels. The two objectives are formalized in a unified framework with trace norm and ℓ_1 norm regularizers. Under the supervision of the observed noise-corrupted label matrix, the multi-label classifier and noisy label identifier are jointly optimized by incorporating the label correlation exploitation and feature-induced noise model. Furthermore, by mapping each bag to a feature vector, we extend PML-NI method into multi-instance multi-label learning by identifying noisy labels based on ambiguous instances. A theoretical analysis of generalization bound and extensive experiments on multiple data sets from various real-world tasks demonstrate the effectiveness of the proposed approach.

Index Terms—Multi-lable learning, partial multi-label learning, candidate label set, noisy label identification, multi-instance multi-label learning

1 Introduction

Multi-label learning (MLL) solves problems where each object is assigned with multiple class labels simultaneously [1]. For instance, an image may be annotated with labels *sea*, *sunset* and *beach*. A large number of recent works have witnessed the great successes that MLL has achieved in many research areas, e.g., music emotion recognition [2], text categorization [3] and image annotation [4].

In traditional multi-label studies, a basic assumption is that each training instance has been precisely annotated with all of its relevant labels. However, in many real-world scenarios, it is difficult and costly to obtain precise annotations. Instead, it is more common that a set of candidate labels are roughly assigned by noisy annotators. In addition to the relevant labels, the candidate set may also contain some noisy labels, where the number of relevant or noisy labels is unknown. For example, in crowdsourcing image tagging (as shown in Fig. 1), among the candidate labels annotated by annotators, only some of them are accurate ones owing to potential unreliable annotators. The scenario has been formalized as a learning framework called partial multi-label learning (PML) by [5].

To solve PML problems, one straightforward method is to simply treat all the candidate labels as relevant ones. Then the PML problem can be solved by standard multi-label learning algorithms, e.g., Binary Relevance (BR) [6], ML-kNN [7], CPLST [8] and so on. However, such methods will be misled by the noisy labels in the candidate set, and fail to generalize well on future data.

In order to deal with the challenge, several PML techniques are proposed recently. Among them, the most commonly used strategy to learn from PML examples is *disambiguation*. It tries to recover ground-truth labeling information from candidate labels, by either introducing labeling confidences [5], [9] or employing low-rank and sparse decomposition scheme [10]. Despite the advances these methods have achieved, a potential limitation is that they neglect the cause of noisy labels in the candidate set, which may be an important information for recovering the ground-truth labels. These methods typically assume that noisy labels are generated randomly, which may be not consistent with many real-world scenarios. In practice, we observe that noisy labels are usually caused by some ambiguous contents of the example and there thus exist some relationships between the noisy labels and feature representations. For example, in crowdsourcing annotation scenario, annotators may be misled by some ambiguous contents associated with the example in specific tasks. Fig. 1 illustrates an example in crowdsourcing image tagging, annotators provided the image with noisy labels *flower*, *cat* and *people* due to the misleading objects marked by the red, green and blue boxes. Similar cases also happen in other tasks, such as ambiguous words in the text categorization and ambiguous melody fragments in the music emotion recognition.

Based on the observations mentioned above, in this paper, we propose a new approach for Partial Multi-label Learning with Noisy label Identification (PML-NI), which recovers the ground-truth labeling information and identifies the noisy labels simultaneously. Specifically, the multi-label classifier

- *The authors are with the MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China. E-mail: {mkxie, huangsj}@nuaa.edu.cn.*

Manuscript received 22 June 2020; revised 19 Dec. 2020; accepted 6 Feb. 2021.
Date of publication 15 Feb. 2021; date of current version 3 June 2022.
(Corresponding authors: Sheng-Jun Huang.)
Recommended for acceptance by L. Wang.
Digital Object Identifier no. 10.1109/TPAMI.2021.3059290

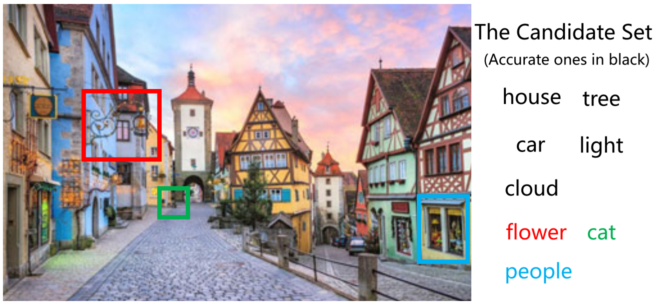


Fig. 1. An example of partial multi-label learning. The image is partially labeled by noisy annotators in crowdsourcing. Among the candidate labels, house, tree, car, light and cloud are ground-truth labels while flower, cat and people are noisy labels.

and noisy label identifier are learned jointly under the supervision of the observed noise-corrupted label matrix. On one hand, the multi-label classifier is constrained to be low rank by trace norm regularization to capture the correlation among labels; on the other hand, the noisy label identifier with sparsity regularization is trained to model the feature-induced noise labels. Our theoretical analysis shows that the generalization performance will be improved by introducing feature-induced noise model. Comprehensive experiments on multiple data sets from various real-world tasks further validate that the proposed approach consistently outperforms the compared methods.

In many real-world scenarios, one object can be represented by multiple instances simultaneously [11]. For example, multiple patches can be extracted from an image where each patch can be regarded as an instance, and thus the image can be represented by a set of instances. To deal with partial-labeled multi-instance data, we extend the partial multi-label learning into multi-instance setting, and propose a novel learning framework called multi-instance partial multi-label learning (MIPML). By mapping each bag into a feature vector, we extend PML-NI method to identify the noisy labels based on ambiguous instances.

The rest of this paper is organized as follows: Section 2 reviews some related works; Sections 3 and 4 introduce our proposed PML-NI and MIPML-NI approaches, respectively; experimental results are reported in Section 5, followed by the conclusion in Section 6.

2 Related Work

Partial multi-label learning is a powerful framework to deal with partially labeled data in multi-label setting. It is derived from two popular learning frameworks: multi-label learning and partial label learning.

There are plenty of literature on multi-label learning. Among them, Binary Relevance is the most simple approach which decomposes the task into a set of binary classification problems [6]. There are many studies trying to exploit the label correlations for enhancing the multi-label learning [12], [13]. Some of them focus on pairwise correlation [14], while some others consider high order correlation among all labels [15].

Partial label learning (PLL) is a framework for learning from partially labeled data for single label tasks [16], [17]. In PLL problem, the partial label set consists of exactly one ground-truth label and some other noisy labels. The most common strategy applied in PLL methods is *disambiguation*, which tries to recover the ground-truth label from the candidate set [18], [19], [20]. The disambiguation strategy are mostly implemented in two ways: one is to assume certain parametric model and the ground-truth label is regarded as the latent variable which can be iteratively refined by optimizing certain objectives, such as the maximum likelihood criterion [16], [17] or the maximum margin criterion [21]; the other one is to assume equal importance of each candidate label and then make prediction by averaging their modeling outputs. For parametric models, the averaged outputs for all candidate labels are distinguished from the outputs for candidate labels [22]. For non-parametric models, the predicted label for unseen instance is determined by averaging the candidate labeling information from its neighboring examples in the PL training set [23], [24]. Compared to partial label learning, PML is much more challenging owing to the number of ground-truth labels in the candidate set is unknown. Note that in multi-label learning, Label Powerset [25] transforms the multi-label learning problem to multiple multi-class problems. However, it is difficult to employ the similar technique to transform PML problem into multiple partial label learning problems since in PML, besides the relevant labels, the candidate set also contains multiple noisy labels. For a specific class label, it is difficult to determine whether an instance whose candidate set contains the label is a positive instance, since the label may be a noisy label for the instance.

To solve PML problems, the most intuitive method is to treat all candidate labels as relevant ones. In this case, PML problem can be solved by off-the-shelf multi-label learning algorithms. Nevertheless, such methods will be misled by the noisy labels in the candidate set, which may lead to degraded performances. In order to overcome this problem, some techniques are designed to solve PML problems recently. For example, two effective methods PML-*lc* and PML-*fp* are proposed in [5] by introducing a confidence value for each candidate label. In [26], authors propose to achieve disambiguation by utilizing low-rank matrix approximation and latent dependencies between labels and feature. The decomposition scheme is utilized to tackle PML data in [10]. PARTICLE [9] identifies the credible labels with high labeling confidences by employing an iterative label propagation procedure. In [27], authors deal with the PML problems by using adversarial training. DRAMA [28] trains a gradient boosting model fit the label confidence learned from manifold structure in the feature space. Disambiguation procedure is performed for candidate sets by using a relabel mechanism in [29]. PML has been extended to some other learning scenarios, such as multi-view learning [30] and semi-supervised learning [31]. Despite the advances these methods have achieved, a potential limitation is that they do not consider the cause of noisy labels in the candidate set, which may be an essential information for solving PML problems.

During the past few years, there were many methods developed to solve MIML problems. Among them, MIMLSVM and MIMLBoost [11] are two early proposed methods, in which the former degenerates the MIML problem into single-instance multi-label tasks while the later degenerates MIML into multi-instance single label learning. In [32], authors propose a generative model for MIML problems. Nearest neighbor is adapted into MIML setting in [33]. Neural network is employed to solve MIML problems in [34]. A hidden conditional random field model for MIML image annotation is proposed in [35]. By optimizing ranking loss, RankLossSIM method is proposed in [36] for MIML instance annotation. A method called KISAR is proposed to discover the relation between labels and instances in MIML learning [37]. In [38], authors propose MIMLfast to solve large-scale MIML problems. In [39], a discriminative probabilistic model is proposed for MIML instance annotation. Recently, as the development of deep learning, the emergence of deep MIML methods proposed in [40] offers a powerful framework for solving MIML problems.Some studies aims to extend the MIML framework to novel settings, such as multi-view learning [41], novel class detection [42] and instance clustering [43].

3 The PML Approach

For each partially labeled training example, we denote by $\mathbf{x}_i \in \mathbb{R}^d$ a feature vector and its corresponding label vector $\mathbf{y} \in \{0,1\}^q$ with q class labels. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] \in \{0,1\}^{q\times n}$ denote the feature matrix and noise-corrupted label matrix, respectively. In this setting, $\mathbf{y}_{ji} = 1$ means the jth label is a candidate label to the ith instance. For any $A, B \in \mathbb{R}^{d\times n}$, we denote by $\langle A, B\rangle = \mathrm{tr}(A^\top B)$ their Hilbert-Schmidt inner product.

3.1 The PML-NI Framework

In partial multi-label learning, each instance is associated with a candidate label set which contains both ground-truth labels and noisy labels and thus the observed matrix $\mathbf{Y}$ can be represented as following:

$$\mathbf{Y} = \mathbf{Y}_g + \mathbf{Y}_n,$$

where $\mathbf{Y}_g$ and $\mathbf{Y}_n$ denote the ground-truth label matrix and noisy label matrix, respectively. In traditional multi-label learning, the ground-truth label matrix $\mathbf{Y}_g$ is often approximated by a linear mapping W from the feature space to the target space

$$\begin{aligned} &\mathbf{Y}_g \approx W\mathbf{X} \\ &s.t. \quad rank(W) \le \epsilon, \end{aligned} \tag{1}$$

where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_q]^\top \in \mathbb{R}^{q\times d}$ is a weight matrix called the multi-label classifier. Here, for simplicity, we omit the bias term which can be easily extended. In multi-label learning, a common assumption is that there exist label correlations among different labels [15] and the feature mapping matrix W is thus linearly dependent. To capture such intrinsic property of the multi-label classifier, the constraint $rank(W) \le \epsilon$ is employed to introduce the low-rank assumption.

As mentioned in the above discussion, in many real-world scenarios, noisy labels are usually caused by some ambiguous contents of the example and there thus exist some relationships between noisy labels and feature contents. Here we model the noisy labels as the outputs of a linear mapping from the feature representations as follows:

$$\begin{aligned} &\mathbf{Y}_n \approx S\mathbf{X} \\ &s.t. \quad card(S) \le \sigma, \end{aligned} \tag{2}$$

where $S = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_q]^\top \in \mathbb{R}^{q\times d}$ is a weight matrix called the noisy label identifier. Here, the bias term is also omitted for simplicity. Note that noisy labels are usually caused by some specific content, i.e., only a few of ambiguous and the feature mapping matrix S is sparse which indicates only some key features are activated. To capture such structure information, we use constraint $card(S) \le \sigma$ to introduce feature-induced noise model, where $card(S)$ is the cardinality operator which measures the number of non-zero elements in S. Accordingly, the goal of our framework is to determine the optimal parameters W and S given the observed label matrix $\mathbf{Y}$ and feature matrix $\mathbf{X}$. However, neither the ground-truth labels $\mathbf{Y}_g$ nor noisy labels $\mathbf{Y}_n$ here are known and the equations in Eqs. (1) and (2) are thus intractable. To solve the problem, we propose a joint learning model H that can identify the noisy labels while training the multi-label classifier simultaneously

$$\begin{aligned} \min_{H,W,S} \; &\mathcal{L}(H, \mathbf{X}, \mathbf{Y}) + \lambda R(H) \\ s.t. \; & H = W + S \\ & rank(W) \le \epsilon \\ & card(S) \le \sigma. \end{aligned} \tag{3}$$

Here, $H \in \mathbb{R}^{q\times d}$ is the joint learning model that consists of the multi-label classifier W and noisy label identifier S. $\mathcal{L}$ is the loss function to minimize empirical loss between modeling outputs $H\mathbf{X}$ and the observed matrix $\mathbf{Y}$. R is a regularization term to control the model complexity, where λ is a balancing parameter. For simplicity, we choose the least square loss for model training and square Frobenius norm to control the model complexity, and then the optimization problem in Eq. (3) can be re-written by

$$\begin{aligned} \min_{H,W,S} \; &\frac{1}{2}\|\mathbf{Y} - H\mathbf{X}\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|H\|_{\mathrm{F}}^2 \\ s.t. \; & H = W + S \\ & rank(W) \le \epsilon \\ & card(S) \le \sigma. \end{aligned} \tag{4}$$

Unfortunately, the problem (4) is hardly solved due to the intractability of rankness and cardinality constraints. To deal with the issue, the Lagrange form of problem (4) is alternatively solved

$$\begin{aligned} \min_{H,W,S} \; &\frac{1}{2}\|\mathbf{Y} - H\mathbf{X}\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|H\|_{\mathrm{F}}^2 + \beta rank(W) + \gamma card(S) \\ s.t. \quad & H = W + S, \end{aligned} \tag{5}$$

where β and γ are balancing parameters. However, it is also difficult to solve the problem (5) due to the rankness and cardinality operators are highly non-convex and computationally NP-hard. Therefore, these two operators are relaxed by their convex surrogate, i.e., the trace norm for low-rank property [44] and ℓ_1-norm for sparsity [45]. Finally, the optimization problem can be formulated as follows:

$$\begin{aligned}\min_{\boldsymbol{H},\boldsymbol{W},\boldsymbol{S}} \frac{1}{2}\|\mathbf{Y}-\boldsymbol{H}\mathbf{X}\|_{\mathrm{F}}^{2}+\frac{\lambda}{2}\|\boldsymbol{H}\|_{\mathrm{F}}^{2}+\beta\|\boldsymbol{W}\|_{tr}+\gamma\|\boldsymbol{S}\|_{1} \\ s.t. \quad \boldsymbol{H}=\boldsymbol{W}+\boldsymbol{S}.\end{aligned} \tag{6}$$

3.2 Optimization

To solve the problem (6), we first apply the augmented Lagrange multiplier method to obtain the following Lagrange function:

$$\begin{aligned}\mathcal{L}(\boldsymbol{H},\boldsymbol{W},\boldsymbol{S},\boldsymbol{A},\mu) &= \|\mathbf{Y}-\boldsymbol{H}\mathbf{X}\|_{\mathrm{F}}^{2}+\frac{\lambda}{2}\|\boldsymbol{H}\|_{\mathrm{F}}^{2} \\ &+\beta\|\boldsymbol{W}\|_{\mathrm{tr}}+\gamma\|\boldsymbol{S}\|_{1}+\frac{\mu}{2}\|\boldsymbol{H}-\boldsymbol{W}-\boldsymbol{S}+\boldsymbol{A}/\mu\|_{\mathrm{F}}^{2},\end{aligned} \tag{7}$$

where $\boldsymbol{A}\in\mathbb{R}^{q\times d}$ is the Lagrange multiplier matrix and μ is the penalty parameter. The inexact ALM (IALM) [46] method can be employed to solve the optimization problem in Eq. (7) by optimizing each of variables iteratively. The main optimizing rules are summarized as follows.

With $\boldsymbol{W}$ and $\boldsymbol{S}$ fixed, the optimization problem in Eq. (7) with respective to $\boldsymbol{H}$ can be reformulated as follows:

$$\min_{\boldsymbol{H}}\|\mathbf{Y}-\boldsymbol{H}\mathbf{X}\|_{\mathrm{F}}^{2}+\frac{\lambda}{2}\|\boldsymbol{H}\|_{\mathrm{F}}^{2}+\frac{\mu}{2}\|\boldsymbol{H}-\boldsymbol{W}-\boldsymbol{S}+\boldsymbol{A}/\mu\|_{\mathrm{F}}^{2}, \tag{8}$$

which can be solved in a closed form

$$\boldsymbol{H}_{k+1}=(\mathbf{Y}\mathbf{X}^{\top}+\mu\boldsymbol{W}_{k}+\mu\boldsymbol{S}_{k}+\boldsymbol{A})(\mathbf{X}\mathbf{X}^{\top}+\lambda\boldsymbol{I}+\mu\boldsymbol{I})^{-1}. \tag{9}$$

With $\boldsymbol{H}$ fixed, the variables $\boldsymbol{W}$ and $\boldsymbol{S}$ can be optimized by solving following problem:

$$\min_{\boldsymbol{W},\boldsymbol{S}}\beta\|\boldsymbol{W}\|_{\mathrm{tr}}+\gamma\|\boldsymbol{S}\|_{1}+\frac{\mu}{2}\|\boldsymbol{H}-\boldsymbol{W}-\boldsymbol{S}+\boldsymbol{A}/\mu\|_{\mathrm{F}}^{2},$$

which is a robust PCA (RPCA) problem [46], and the optimizing rules are given

$$\begin{aligned}\boldsymbol{W}_{k+1}&=\mathcal{T}_{\beta/\mu}[\boldsymbol{H}_{k}-\boldsymbol{S}_{k}+\boldsymbol{A}_{k}/\mu_{k}] \\ \boldsymbol{S}_{k+1}&=\mathcal{S}_{\gamma/\mu}[\boldsymbol{H}_{k}-\boldsymbol{W}_{k+1}+\boldsymbol{A}_{k}/\mu_{k}].\end{aligned}$$

Here, $\mathcal{T}$ is the single value thresholding operator [47], which first performs singular value decomposition on $\boldsymbol{H}_{k}-\boldsymbol{S}_{k}+\boldsymbol{A}_{k}/\mu_{k}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\top}$, then the solution is given by $\mathbf{U}\widehat{\boldsymbol{\Sigma}}\mathbf{V}^{\top}$, where $\widehat{\Sigma}_{ii}=\max(0,\Sigma_{ii}-\beta/\mu)$. $\mathcal{S}_{\gamma/\mu}$ is the shrinkage operator, which is defined as $\mathcal{S}_{\omega}(a)=(a-\omega)_{+}-(-a-\omega)_{+}$.

At last, the Lagrange multiplier matrix $\boldsymbol{A}$ and penalty parameter μ are updated based on following rules:

$$\begin{aligned}\boldsymbol{A}_{k+1}&=\boldsymbol{A}_{k}+\mu(\boldsymbol{H}_{k+1}-\boldsymbol{W}_{k+1}-\boldsymbol{S}_{k+1}) \\ \mu_{k+1}&=\min(\mu_{\max},\rho\mu_{k}),\end{aligned}$$

where $\mu_{\max}$ is the maximum value of μ and ρ is a positive updating parameter which are defined by users. Algorithm 1 summarizes the main steps of PML-NI method.

Theorem 1. *For Algorithm 1, if $\{\mu_k\}$ is nondecreasing and $\sum_{k=1}^{+\infty}\mu_k^{-1}=+\infty$, then $(\boldsymbol{W}_k,\boldsymbol{S}_k)$ converge to an optimal solution $(\boldsymbol{W}^*,\boldsymbol{S}^*)$ to the PML-NI problem.*

Proof. According to [46], as $\boldsymbol{H}_{k+1}^{*}-\boldsymbol{W}_{k+1}^{*}-\boldsymbol{S}_{k+1}^{*}=\mu_{k}^{-1}(\boldsymbol{A}_{k+1}^{*}-\boldsymbol{A}_{k}^{*})$,by the boundedness of $\boldsymbol{A}_{k}^{*}$ and as $k\to+\infty$, we have $\boldsymbol{H}^{*}=\boldsymbol{W}^{*}+\boldsymbol{S}^{*}$. Therefore, the last term of Eq. (8) equals zero and $\boldsymbol{H}^{*}$ becomes the optimal solution of equation $\|\mathbf{Y}-\boldsymbol{H}\mathbf{X}\|_{\mathrm{F}}^{2}+\frac{\lambda}{2}\|\boldsymbol{H}\|_{\mathrm{F}}^{2}$. Accordingly, $(\boldsymbol{W}_k,\boldsymbol{S}_k)$ converge to an optimal solution $(\boldsymbol{W}^*,\boldsymbol{S}^*)$ to the PML-NI problem. □

Algorithm 1. PML-NI Method

```
Input: Feature matrix $\mathbf{X}\in\mathbb{R}^{d\times n}$, Observed label matrix
  $\mathbf{Y}\in\{0,1\}^{q\times n}$ as well as balancing parameters $\lambda$, $\beta$ $\gamma$
Output: Learned weight matrix $(\boldsymbol{W}^*,\boldsymbol{S}^*)$
Initialize $\mu_0 > 0, \rho > 1$ and $k=0$
while not converged do
    Updating $\boldsymbol{H}_{k+1}$ according to Eq. (9)
    Updating $\boldsymbol{W}_{k+1}=\mathcal{T}_{\beta/\mu}[\boldsymbol{H}_k-\boldsymbol{S}_k+\boldsymbol{A}_k/\mu_k]$
    Updating $\boldsymbol{S}^{k+1}=\mathcal{S}_{\gamma/\mu}[\boldsymbol{H}_k-\boldsymbol{W}_{k+1}+\boldsymbol{A}_k/\mu_k]$
    Updating $\boldsymbol{A}_{k+1}=\boldsymbol{A}_k+\mu_k(\boldsymbol{H}_{k+1}-\boldsymbol{W}_{k+1}-\boldsymbol{S}_{k+1})$
    Updating $\mu_{k+1}=\min(\mu_{\max},\rho\mu_k)$
    $k\leftarrow k+1$
end while
return result
```

3.3 Generalization Bound

In this section, we first provide the Rademacher complexity [48] for PML-NI, which is a commonly used tool for performing comprehensive analysis of data-dependent risk bounds.

Definition 1. *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{X}$ to [0,1] and $S=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}\in\mathcal{X}^n$ a fixed sample of size n. Then, the empirical Rademacher complexity of $\mathcal{G}$ with respective to sample S is defined as*

$$\widehat{\mathcal{R}}_{S}(\mathcal{G})=\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}}\frac{1}{n}\sum_{i=1}^{n}\sigma_{i}g(x_{i})\right], \tag{10}$$

where $\boldsymbol{\sigma}=(\sigma_1,\ldots,\sigma_n)$ are Rademacher variables with σ_i independent uniform random variable taking value in $\{-1,+1\}$.

Lemma 1. *Let $\mathcal{H}=\mathcal{W}\times\mathcal{S}$ be the family of functions for PML-NI with linear functions $(\boldsymbol{W},\boldsymbol{S})\in\mathcal{H}$. For the square loss function ℓ, the Rademacher complexity of the proposed algorithm can be bounded by*

$$\widehat{\mathcal{R}}_{S}(\ell\circ\mathcal{H})\leq\frac{\sqrt{2}(2q)}{n}\mathbb{E}\left[\sup_{(\boldsymbol{W},\boldsymbol{S})\in\mathcal{H}}\left\langle\boldsymbol{W}^{\top}+\boldsymbol{S}^{\top},\widehat{\mathbf{X}}\right\rangle\right], \tag{11}$$

where $\widehat{\mathbf{X}}$ will be defined later.

Proof. Based on the Definition 1, the Rademacher complexity with respective to $\mathcal{H}$ and ℓ can be rewritten as

$$\widehat{\mathcal{R}}_{S}(\ell\circ\mathcal{H})=\frac{1}{n}\mathbb{E}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{n}\sigma_{i}\ell(h(\mathbf{x}_{i}),\mathbf{y}_{i})\right],$$

where the subscript $\boldsymbol{\sigma}$ is omitted for notational simplicity and $h\in\mathcal{H}$ is a real value function. According to the contraction inequality for Rademacher complexity (see Theorem 3 of [49]), note that the square loss is $2q$-Lipschitz for PML-NI, then the Rademacher complexity defined in above equation can be bounded by

$$\widehat{\mathcal{R}}_S(\ell \circ \mathcal{H}) \leq \frac{\sqrt{2}(2q)}{n} \mathbb{E}\left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{n} \sum_{j=1}^{q} \sigma_{ij} h_j(\mathbf{x}_i)\right],$$

where $h_j(\mathbf{x}_i)$ corresponds the jth component of $h(\mathbf{x}_i)$, $[\sigma_{ij}]_{n\times q}$ are $n \times q$ Rademacher variables with σ_{ij} independent uniform random variable taking value in $\{-1, +1\}$. Accordingly, the Rademacher complexity of PML-NI algorithm can be bounded by

$$\widehat{\mathcal{R}}_S(\ell \circ \mathcal{H}) \leq \frac{\sqrt{2}(2q)}{n} \mathbb{E}\left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{n} \sum_{j=1}^{q} \sigma_{ij} (\boldsymbol{w}_j + \boldsymbol{s}_j) \cdot \mathbf{x}_i\right],$$

where $\boldsymbol{w}_j$ and $\boldsymbol{s}_j$ denote the jth rows of $\boldsymbol{W}$ and $\boldsymbol{S}$, respectively, which are the feature mappings vector for the jth label. To further simplify the notations, $\widehat{\mathbf{X}}_j = \sum_{i=1}^{n} \sigma_{ij}\mathbf{x}_i$ is used to represent the weight summation of feature vectors for jth label. By arranging $\widehat{\mathbf{X}}_j$ for each of q labels one-by-one in columns, we obtain the weight summation matrix $\widehat{\mathbf{X}} \in \mathbb{R}^{d\times q}$. Accordingly, the right side of the above equation can be concretely rewritten as follows:

$$\frac{\sqrt{2}(2q)}{n} \mathbb{E}\left[\sup_{(\boldsymbol{W},\boldsymbol{S})\in\mathcal{H}} \left\langle \boldsymbol{W}^\top + \boldsymbol{S}^\top, \widehat{\mathbf{X}} \right\rangle\right],$$

which finishes proof of Lemma 1. □

Theorem 2. *Let $\mathcal{H} = \mathcal{W} \times \mathcal{F}$ be a family of functions for PML-NI with q outputs, and $(\boldsymbol{W}, \boldsymbol{S}) \in \mathcal{H}$ be linear functions learned on $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$ by PML-NI. The learning algorithm further encourages that $rank(\boldsymbol{W}) \leq \epsilon$ and $\|\boldsymbol{S}\|_1 \leq \sigma$. Then the Rademacher complexity of PML-NI with square loss ℓ satisfies*

$$\widehat{\mathcal{R}}_S(\ell \circ \mathcal{H}) \leq \frac{2\sqrt{2}q(\sqrt{q}\epsilon + q\sigma)}{\sqrt{n}},$$

where we assume that $\|\mathbf{x}\| \leq 1$.

Proof. According to Lemma 1, we have

$$\begin{aligned}
\widehat{\mathcal{R}}_S(\ell \circ \mathcal{H}) &\leq \frac{2\sqrt{2}q}{n} \mathbb{E}\left[\sup_{(\boldsymbol{W},\boldsymbol{S})\in\mathcal{H}} \left\langle \boldsymbol{W}^\top + \boldsymbol{S}^\top, \widehat{\mathbf{X}} \right\rangle\right] \\
&\leq \frac{2\sqrt{2}q}{n} \mathbb{E}\left[\sup_{(\boldsymbol{W},\boldsymbol{S})\in\mathcal{H}} (\|\boldsymbol{W}\|_* + \|\boldsymbol{S}\|_*) \cdot \|\widehat{\mathbf{X}}\|_{\mathrm{F}}\right] \\
&\leq \frac{2\sqrt{2}q}{n} \mathbb{E}\left[\sup_{(\boldsymbol{W},\boldsymbol{S})\in\mathcal{H}} \left(\epsilon + \sqrt{q}\|\boldsymbol{S}\|_1\right) \cdot \|\widehat{\mathbf{X}}\|_{\mathrm{F}}\right].
\end{aligned}$$

It is also easy to prove the bound as follows:

$$\mathbb{E}_{\boldsymbol{\sigma}}\|\widehat{\mathbf{X}}\|_{\mathrm{F}}^2 = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{j=1}^{q} \|\widehat{\mathbf{X}}_j\|_2^2\right] = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{j=1}^{q} \left\|\sum_{i=1}^{n} \sigma_{ij}\mathbf{x}_i\right\|_2^2\right] \leq nq.$$

Then we have

$$\widehat{\mathcal{R}}_S(\ell \circ \mathcal{H}) \leq \frac{2\sqrt{2}q(\sqrt{q}\epsilon + q\sigma)}{\sqrt{n}},$$

which finishes the proof of Theorem 2. □

To conclude the Theorem 2, the following Lemma is introduced to show the relationship between the risk of an algorithm and its Rademacher complexity.

Lemma 2. *[48] Let $\mathcal{G}$ be a family of functions. For a loss function ℓ bounded by Θ, then for any $\delta > 0$, with probability at least $1 - \delta$, for all $g \in \mathcal{G}$ such that*

$$\mathcal{L}_{\mathcal{D}}(g) \leq \mathcal{L}_S(g) + \widehat{\mathcal{R}}_S(\ell \circ \mathcal{G}) + 3\Theta\sqrt{\frac{\log 2/\delta}{2n}},$$

where $\mathcal{L}_{\mathcal{D}}(g)$ and $\mathcal{L}_S(g)$ are risk and empirical risk with respective to f.

Lemma 2 motivates us to acquire a smaller Rademacher complexity when designing the algorithm. To emphasize the superiority of the proposed algorithm, let us first consider a typical algorithm $\widehat{W}$ for solving PML problems, which ignore the noisy labels in the candidate set and still minimize $rank(\widehat{W})$ to exploit label correlations. Without considering the negative influence of the noisy labels , the Rademacher complexity is likely to be large due to the high-rankness of $\widehat{W}$. Compared to the algorithm discussed above, according to Theorem 2, the Rademacher complexity of PML-NI can be bounded by two parts corresponding to low-rankness of $\boldsymbol{W}$ and sparsity of $\boldsymbol{S}$, respectively. Among them, the latter part, i.e., the sparsity of $\boldsymbol{S}$ can highly alleviate the negative influence of the noisy labels, thus cut down the rankness of the classifier $\boldsymbol{W}$ and further obtains a lower bound of Rademacher complexity, which leads to a good generalization performance.

4 Multi-Instance Partial Multi-Label Learning

In this section, we extend PML-NI to multi-instance partial multi-label learning. Compared to traditional PML, MIPML is more naturally for representing partial-labeled data since each object can be described by high-level representation *instance*, which makes the task of identifying noisy labels based on the representations more achievable. In MIPML problems, each example is represented by a bag of instances and associated with a set of candidate labels. Our goal is to train a classifier based on training examples with candidate label sets which can predict all the relevant labels for a unseen bag. To solve MIPML problems, one straightforward method is to simply treat all candidate labels as relevant ones. Then the MIPML problems can be solved by standard multi-instance multi-label learning algorithms, e.g., MIMLSVM [11], MIMLfast [38] and so on. Obviously, such methods will be over-fitting due to the noisy labels in the candidate set. To solve the problem, in this paper, we propose a new method for multi-instance partial multi-label learning with noisy label identification (MIPML-NI), which adapts PML-NI method to identify noisy labels based on ambiguous instances while recover the ground-truth labeling information.

In MIPML, we are given a set of training examples $D = \{B_i, \mathbf{y}_i\}_{i=1}^{n}$, where $B_i = \{\mathbf{x}_{i,j}\}_{j=1}^{m_i}$ is a bag which consists of m_i instances and $\mathbf{y}_i = [y_{i1}, \ldots, y_{iq}] \in \{0, 1\}^q$ is the label vector of bag B_i. Here, $y_{ij} = 1$ indicates the label j is candidate to the ith bag B_i. Let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] \in \{0, 1\}^{q\times n}$ denote the observed label matrix. The goal is to learn a classifier based on D which can predict all relevant labels for any unseen bag.

Inspired by [11], [37], we first represent a bag of instances by a feature vector

$$\phi(B) = [sim(B, \boldsymbol{c}_1), \ldots, sim(B, \boldsymbol{c}_k)], \tag{12}$$

where $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$ are k prototypes of all the instances, and sim is a function to measure the similarity between bag B and prototype $\boldsymbol{c}_j, \forall \in \{1, \ldots, k\}$, where the larger value of $sim(B, \boldsymbol{c}_j)$, the more similar the bag B and $\boldsymbol{c}_j$. In this paper, the prototypes are instantiated by centroids of clusters obtained by using k-means. Specifically, we employ the Gaussian distance as the similarity function, i.e., $sim(B, \boldsymbol{c}) = \min_{\mathbf{x} \in B} \exp(-\frac{\|\mathbf{x}-\boldsymbol{c}\|_2^2}{\delta})$, where δ is set to be the averaging distance between the instances in a cluster. For notational simplicity, we denote the feature vector $\phi(B_i)$ for bag B_i as ϕ_i and the matrix $\Phi = [\phi_1, \ldots, \phi_n] \in \mathbb{R}^{k \times n}$ is obtained by arranging feature vectors of all bags.

In such case, each prototype can be treated as a group of similar instances. Accordingly, the task of identifying ambiguous content can be transformed into the task of identifying ambiguous instances, which can be formulated as following:

$$\begin{aligned} &\mathbf{Y}_n \approx \boldsymbol{S}\Phi \\ &s.t. \quad card(\boldsymbol{S}) \le \sigma, \end{aligned}$$

where $\boldsymbol{S}$ is the noisy label identifier for multi-instance partial-labeled data. By further generalizing the Eq. (6), the optimization problem of MIPML-NI can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{S}} \frac{1}{2}\|\mathbf{Y} - \boldsymbol{H}\Phi\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\boldsymbol{W}\|_{\mathrm{F}}^2 + \beta\|\boldsymbol{W}\|_{\mathrm{tr}} + \gamma\|\boldsymbol{S}\|_1 \\ s.t. \quad \boldsymbol{H} = \boldsymbol{W} + \boldsymbol{S}. \end{aligned} \tag{13}$$

Here, $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k] \in \mathbb{R}^{q \times k}$ is the multi-instance multi-label classifier. The optimization problem can be effectively solved by Algorithm 1. Note that theoretical results in Section 3.3 can be easily applied to MIPML-NI method with slight modifications. The noisy label identifier $\boldsymbol{S}$ of MIPML-NI method can identify the noisy labels based on ambiguous instances and thus alleviate their negative influence. Therefore, the Rademacher complexity of the model will be lowered due to the low-rankness of the multi-instance multi-label classifier $\boldsymbol{W}$. Since the Rademacher complexity is a kind of data-dependent complexity, the only need is to re-compute the Rademacher complexity for MIPML-NI method as done as in the case of PML-NI method.

5 Experiments

The experiments for PML tasks are first reported, followed by the experiments for MIPML tasks.

5.1 Study on PML Data

5.1.1 Experimental Setting

We perform experiments on totally 15 data sets.[1] These data sets spanned a broad range of applications: *image*, *scene* and *corel16k* for image annotation, *music_emotion*, *music_style* and *birds* for music recognition, *genbase*, *YeastCC*, *YeastMF* and *YeastBP* for protein classification as well as *medical*, *slashdot*, *enron*, *bibtex* and *tmc2007* for text categorization. We also did some pre-processing to facilitate the partially labeling as in [5], [9]. Specifically, for data sets with too many class labels (more than 100 in our experiments), their rare labels are filtered out to keep under 15 labels, and instances without any relevant labels are filtered out.

There are different criteria for evaluating the performances of multi-label learning. In our experiments, we employ five commonly used criteria including *ranking loss*, *hamming loss*, *one error*, *coverage* and *average precision*. More detail about these evaluation metrics can be found in [1]. For the *ranking loss*, *hamming loss*, *one error* and *coverage* metrics, the smaller value, the better the performance. For the *average precision* metric, the larger the value, the better the performance.

To validate the effectiveness of the proposed PML-NI[2] method, we compare with four state-of-the-art PML algorithms as well as two well-established MLL approaches:

- PARTICLE [9]. It transforms the PML task into a multi-label problem through a label propagation procedure. Then a calibrated label ranking model is induced to instantiate two PML methods PAR-VLS and PAR-MAP.
- PML-LRS [10]. It utilizes low-rank and sparse decomposition scheme to capture the ground-truth label matrix and irrelevant label matrix from the observed candidate label matrix.
- fPML [50]. It employ the low-rank approximation of the observed instance-label association matrix to estimate the labeling confidence and then trains multi-label classifier.
- ML-*k*NN [7]. It is a nearest neighbor based multi-label classification method. ML-*k*NN is a very popular baseline method in multi-label learning literature owing to its simplicity.
- CPLST [8]. It is a typical label embedding approach in MLL, which integrates the concepts of principal component analysis and canonical correlation analysis.

For PML-NI, parameter λ is selected from $\{1, 10, 100\}$ by 3-fold cross validation, and the other two parameters are set $\beta = 0.5$ and $\gamma = 0.5$, respectively. In our experiments, although the parameter λ is determined by cross validation, the algorithm is generally not very sensitive to the parameter and it tends to obtain decent performances with a default value, such as $\lambda = 10$. For the other comparing methods, parameters are determined in the same way if no default value given in their literature.

For the last 10 data sets, to construct partial multi-label assignments for the training data, we simulate the annotation process by using a svm classifier trained on original supervised multi-label data sets as the human annotator. Specifically, a svm classifier is first trained on the multi-label data set. Then, for each instance $\mathbf{x}_i$ of the data set, we add the irrelevant noisy labels of $\mathbf{x}_i$ with $\alpha\%$ number of ground-truth labels according to their probabilities to be relevant labels predicted by the svm classifier and the $\alpha\%$ is

[1] Publicly available at: http://mulan.sourceforge.net/datasets.html and http://meka.sourceforge.net/#datasets

[2] Source code available at: http://milkxie.github.io/code/PMLNI code.zip

TABLE 1
Experimental Results of Each Comparing Approach in Terms of *Ranking Loss*,
Where ●/○ Indicates Whether PML-NI is Superior/Inferior to the Other Method

Data	α%	PML-NI	PAR-VLS	PAR-MAP	PML-LRS	fPML	ML-*k*NN	CPLST
music_emotion		.243 ± .004	.261 ± .007●	.245 ± .006●	.256 ± .002●	.261 ± .004●	.364 ± .009●	.257 ± .006●
music_style		.140 ± .007	.161 ± .005●	.161 ± .006●	.148 ± .006●	.154 ± .006●	.232 ± .006●	.157 ± .005●
YeastCC		.158 ± .015	.432 ± .031●	.261 ± .036●	.169 ± .007●	.420 ± .020●	.357 ± .010●	.404 ± .010●
YeastMF		.205 ± .011	.388 ± .052●	.299 ± .022●	.227 ± .013●	.392 ± .014●	.357 ± .011●	.363 ± .004●
YeastBP		.185 ± .009	.413 ± .022●	.255 ± .006●	.206 ± .013●	.412 ± .007●	.354 ± .009●	.401 ± .004●
birds	50%	.206 ± .026	.438 ± .058●	.285 ± .021●	.302 ± .018●	.287 ± .017●	.324 ± .040●	.252 ± .012●
	100%	.217 ± .020	.400 ± .046●	.298 ± .017●	.323 ± .028●	.307 ± .028●	.322 ± .019●	.283 ± .031●
	150%	.240 ± .020	.466 ± .066●	.307 ± .026●	.330 ± .014●	.326 ± .028●	.331 ± .030●	.293 ± .013●
genbase	50%	.003 ± .004	.025 ± .013●	.012 ± .006●	.017 ± .004●	.008 ± .007●	.008 ± .004●	.050 ± .010●
	100%	.006 ± .004	.059 ± .030●	.010 ± .004●	.017 ± .003●	.009 ± .003●	.011 ± .004●	.063 ± .018●
	150%	.007 ± .004	.017 ± .008●	.011 ± .004●	.031 ± .008●	.016 ± .007●	.027 ± .007●	.075 ± .016●
medical	50%	.017 ± .008	.157 ± .034●	.071 ± .015●	.048 ± .013●	.054 ± .011●	.047 ± .008●	.089 ± .008●
	100%	.018 ± .007	.155 ± .035●	.074 ± .017●	.049 ± .008●	.053 ± .011●	.047 ± .008●	.097 ± .010●
	150%	.019 ± .005	.147 ± .029●	.073 ± .013●	.053 ± .005●	.045 ± .010●	.049 ± .005●	.102 ± .015●
image	50%	.177 ± .013	.195 ± .045●	.267 ± .102●	.187 ± .010●	.213 ± .019●	.186 ± .016●	.189 ± .019●
	100%	.176 ± .019	.198 ± .042●	.267 ± .099●	.182 ± .014●	.203 ± .012●	.190 ± .012●	.189 ± .010●
	150%	.184 ± .010	.205 ± .059●	.265 ± .139●	.185 ± .015●	.228 ± .010●	.212 ± .013●	.196 ± .013●
slashdot	50%	.041 ± .003	.150 ± .032●	.047 ± .008●	.041 ± .004●	.047 ± .003●	.048 ± .008●	.048 ± .004●
	100%	.039 ± .005	.149 ± .036●	.047 ± .009●	.042 ± .007●	.047 ± .007●	.047 ± .006●	.055 ± .005●
	150%	.037 ± .004	.175 ± .016●	.047 ± .009●	.047 ± .006●	.049 ± .009●	.048 ± .005●	.066 ± .011●
enron	50%	.172 ± .011	.318 ± .070●	.188 ± .047●	.163 ± .021○	.164 ± .009○	.180 ± .007●	.301 ± .019●
	100%	.171 ± .015	.376 ± .088●	.216 ± .048●	.168 ± .012○	.177 ± .014●	.190 ± .011●	.294 ± .011●
	150%	.172 ± .013	.366 ± .077●	.209 ± .047●	.171 ± .021○	.176 ± .013●	.196 ± .011●	.297 ± .017●
scene	50%	.105 ± .005	.154 ± .028●	.198 ± .041●	.106 ± .011●	.114 ± .006●	.115 ± .007●	.165 ± .015●
	100%	.106 ± .006	.153 ± .035●	.187 ± .057●	.107 ± .009●	.112 ± .007●	.122 ± .012●	.162 ± .010●
	150%	.126 ± .013	.178 ± .028●	.200 ± .038●	.122 ± .009○	.128 ± .008●	.146 ± .019●	.219 ± .007●
bibtex	50%	.041 ± .004	.080 ± .002●	.057 ± .001●	.042 ± .002●	.077 ± .008●	.115 ± .008●	.115 ± .010●
	100%	.033 ± .005	.095 ± .006●	.062 ± .004●	.035 ± .004●	.060 ± .009●	.136 ± .019●	.138 ± .002●
	150%	.033 ± .001	.098 ± .007●	.064 ± .004●	.035 ± .003●	.062 ± .007●	.143 ± .011●	.151 ± .006●
corel16K	50%	.221 ± .004	.288 ± .002●	.236 ± .003●	.214 ± .003○	.229 ± .005●	.264 ± .007●	.229 ± .004●
	100%	.226 ± .007	.334 ± .008●	.262 ± .005●	.226 ± .004	.242 ± .003●	.273 ± .002●	.239 ± .005●
	150%	.227 ± .006	.326 ± .007●	.258 ± .003●	.228 ± .001●	.244 ± .003●	.275 ± .007●	.237 ± .005●
tmc2007	50%	.046 ± .001	.087 ± .014●	.057 ± .008●	.046 ± .001	.063 ± .001●	.075 ± .004●	.080 ± .002●
	100%	.047 ± .002	.082 ± .014●	.057 ± .009●	.047 ± .002●	.064 ± .004●	.079 ± .002●	.081 ± .001●
	150%	.050 ± .001	.107 ± .023●	.060 ± .010●	.050 ± .002●	.066 ± .004●	.082 ± .001●	.086 ± .001●

varied in the range $\{50\%, 100\%, 150\%\}$. To examine the performance of the proposed approaches, we performed experiments with all possible percentages of the noisy labels. In the following content, we will show details of three groups of experiments on these totally 35 data sets.

5.1.2 Comparison Results

We follow the setting in [9] to only report detailed results of each comparing methods in terms of *ranking loss* and *average precision* in Tables 1 and 2, while similar results can be observed in terms of other evaluation metrics. When compare PML-NI approach with other methods, our algorithm shows significant superiority. It achieves the best performance in most cases. Among the five comparing approaches, PML-LRS shows some superiority, and is better than PML-NI with three cases on *enron*, one case on *scene* as well as one case on *corel16k* in terms of *ranking loss* and two cases on *scene* in terms of *average precision*, while losses for other cases. fPML outperforms PML-NI with one case on *enron* in terms of *ranking loss*, while losses for other cases. ML-*k*NN is better than PML-NI with one case on *scene* in terms of *average precision*.

To validate the effectiveness of PML-NI for real applications, we also perform experiments on real-world PML data sets *music_emotion* and *music_style*. The results show that PML-NI achieves the best results in almost all cases except for the data set *music_emotion* where PAR-VAL achieves comparable performance than PML-NI in terms of *average precision*.

Furthermore, we also use *Friedman test* [51] as the statistical test to analyze the relative performance among the comparing approaches. Assume that there are k algorithms and N data sets. Let r_i^j denotes the rank of jth algorithm on the ith data set. The average ranks of algorithms $R_j = \frac{1}{N}\sum_i r_i^j$ is used for Friedman test comparison. Under the null-hypothesis, which indicates that all the algorithms have equivalent performance, the Friedman statistic F_F with respective to the F-distribution with $(k-1)(N-1)$ degree of freedom can be defined

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2}, \tag{14}$$

TABLE 2
Experimental Results of Each Comparing Approach in Terms of *Average Precision*, Where ●/○ Indicates Whether PML-NI is Superior/Inferior to the Other Method

Data	α%	PML-NI	PAR-VLS	PAR-MAP	PML-LRS	fPML	ML-kNN	CPLST
music_emotion		.614 ± .005	.605 ± .006●	.614 ± .011	.589 ± .006●	.586 ± .004●	.506 ± .009●	.595 ± .007●
music_style		.737 ± .009	.716 ± .010●	.677 ± .015●	.714 ± .008●	.706 ± .012●	.658 ± .009●	.717 ± .011●
YeastCC		.601 ± .021	.214 ± .024●	.399 ± .045●	.574 ± .020●	.176 ± .008●	.320 ± .006●	.129 ± .006●
YeastMF		.482 ± .012	.237 ± .023●	.291 ± .014●	.447 ± .009●	.246 ± .004●	.281 ± .011●	.274 ± .005●
YeastBP		.429 ± .010	.136 ± .029●	.278 ± .038●	.380 ± .016●	.085 ± .004●	.184 ± .012●	.130 ± .009●
birds	50%	.483 ± .006	.413 ± .034●	.395 ± .024●	.371 ± .030●	.387 ± .033●	.370 ± .037●	.451 ± .015●
	100%	.451 ± .042	.416 ± .042●	.386 ± .024●	.352 ± .033●	.368 ± .033●	.366 ± .037●	.410 ± .033●
	150%	.424 ± .023	.392 ± .033●	.369 ± .023●	.344 ± .031●	.348 ± .025●	.352 ± .017●	.387 ± .040●
genbase	50%	.983 ± .010	.895 ± .022●	.968 ± .020●	.860 ± .022●	.977 ± .014●	.948 ± .011●	.738 ± .028●
	100%	.969 ± .017	.819 ± .039●	.965 ± .019●	.851 ± .025●	.951 ± .018●	.920 ± .055●	.723 ± .030●
	150%	.947 ± .024	.897 ± .042●	.960 ± .010○	.785 ± .049●	.894 ± .040●	.773 ± .069●	.612 ± .020●
medical	50%	.866 ± .037	.703 ± .021●	.737 ± .029●	.738 ± .034●	.796 ± .013●	.737 ± .014●	.592 ± .027●
	100%	.854 ± .007	.680 ± .020●	.714 ± .031●	.724 ± .020●	.792 ± .016●	.734 ± .014●	.568 ± .027●
	150%	.797 ± .013	.673 ± .013●	.675 ± .018●	.665 ± .014●	.741 ± .029●	.664 ± .032●	.498 ± .031●
image	50%	.779 ± .016	.770 ± .055●	.734 ± .076●	.765 ± .013●	.734 ± .027●	.767 ± .015●	.766 ± .019●
	100%	.781 ± .023	.767 ± .051●	.735 ± .077●	.772 ± .016●	.750 ± .010●	.763 ± .016●	.769 ± .007●
	150%	.772 ± .011	.760 ± .068●	.709 ± .150●	.770 ± .016●	.713 ± .011●	.732 ± .009●	.757 ± .015●
slashdot	50%	.896 ± .006	.799 ± .104●	.884 ± .015●	.893 ± .006●	.881 ± .009●	.883 ± .014●	.832 ± .011●
	100%	.895 ± .010	.636 ± .167●	.885 ± .016●	.893 ± .012●	.879 ± .011●	.882 ± .011●	.805 ± .016●
	150%	.885 ± .007	.493 ± .011●	.884 ± .014●	.844 ± .010●	.876 ± .012●	.878 ± .012●	.697 ± .015●
enron	50%	.549 ± .010	.297 ± .132●	.432 ± .068●	.528 ± .022●	.497 ± .023●	.450 ± .017●	.350 ± .004●
	100%	.491 ± .023	.271 ± .129●	.398 ± .081●	.474 ± .019●	.452 ± .013●	.412 ± .016●	.346 ± .013●
	150%	.461 ± .026	.264 ± .120●	.397 ± .058●	.453 ± .021●	.435 ± .021●	.395 ± .017●	.326 ± .022●
scene	50%	.819 ± .004	.787 ± .037●	.756 ± .049●	.824 ± .017○	.811 ± .005●	.832 ± .011○	.750 ± .023●
	100%	.824 ± .008	.783 ± .048●	.761 ± .075●	.820 ± .016●	.811 ± .010●	.822 ± .014●	.752 ± .010●
	150%	.797 ± .012	.760 ± .043●	.743 ± .048●	.801 ± .014○	.793 ± .015●	.794 ± .021●	.682 ± .010●
bibtex	50%	.888 ± .013	.810 ± .009●	.831 ± .006●	.888 ± .007	.801 ± .015●	.748 ± .009●	.733 ± .017●
	100%	.886 ± .012	.763 ± .010●	.816 ± .011●	.874 ± .013●	.815 ± .017●	.708 ± .028●	.621 ± .008●
	150%	.888 ± .005	.761 ± .010●	.816 ± .009●	.873 ± .006●	.805 ± .007●	.697 ± .019●	.598 ± .015●
corel16K	50%	.511 ± .008	.473 ± .003●	.484 ± .003●	.511 ± .004●	.497 ± .004●	.456 ± .010●	.500 ± .003●
	100%	.484 ± .010	.453 ± .006●	.454 ± .007●	.481 ± .007●	.472 ± .005●	.436 ± .004●	.476 ± .005●
	150%	.486 ± .005	.458 ± .004●	.455 ± .009●	.479 ± .005●	.473 ± .006●	.433 ± .009●	.475 ± .007●
tmc2007	50%	.804 ± .002	.731 ± .033●	.783 ± .022●	.803 ± .006●	.780 ± .004●	.746 ± .008●	.747 ± .002●
	100%	.803 ± .005	.737 ± .035●	.785 ± .021●	.802 ± .005●	.778 ± .007●	.729 ± .004●	.738 ± .005●
	150%	.793 ± .003	.676 ± .033●	.760 ± .036●	.792 ± .005●	.773 ± .008●	.710 ± .005●	.721 ± .002●

where,

$$\chi_F^2 = \frac{12N}{k(k-1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]. \tag{15}$$

Table 3 reports the Friedman statistics F_F and the corresponding critical value with respective to each evaluation metric (# comparing algorithms $k = 7$, # data sets $N = 35$). For each evaluation metric, the null hypothesis of indistinguishable performance among the comparing algorithm is rejected at 0.05 significance level.

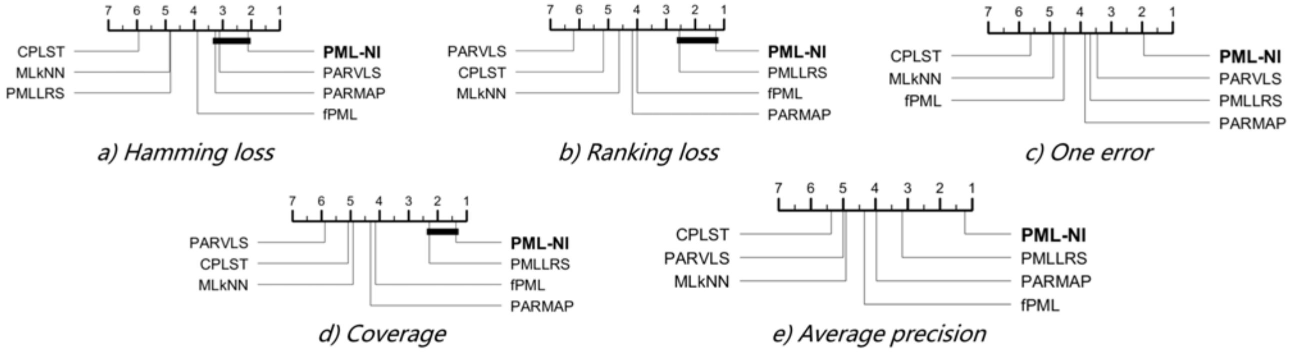
Then, the post-hoc *Bonferroni-Dunn test* [51] is utilized to illustrate the relative performance among comparing approaches. Here, PML-NI is regarded as the control method whose average rank difference against the comparing algorithm is calibrated with the *critical difference* (CD)

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}}, \tag{16}$$

where critical value $q_\alpha = 2.638$ at 0.05 significance level. Accordingly, PML-NI is deemed to have significantly different performance to one comparing algorithm if their average ranks differ by at least one CD (CD = 1.3623 in our experiment: # comparing algorithms $k = 7$, # data sets $N = 35$). Fig. 2 shows the CD diagrams ([51]) on each evaluation metric, where the average rank of each comparing algorithm is marked along the axis (lower ranks to the right). In each subfigure, any comparing algorithms whose average rank is within one CD to that of PML-NI is interconnected to each other with a thick line. From the figure, it can

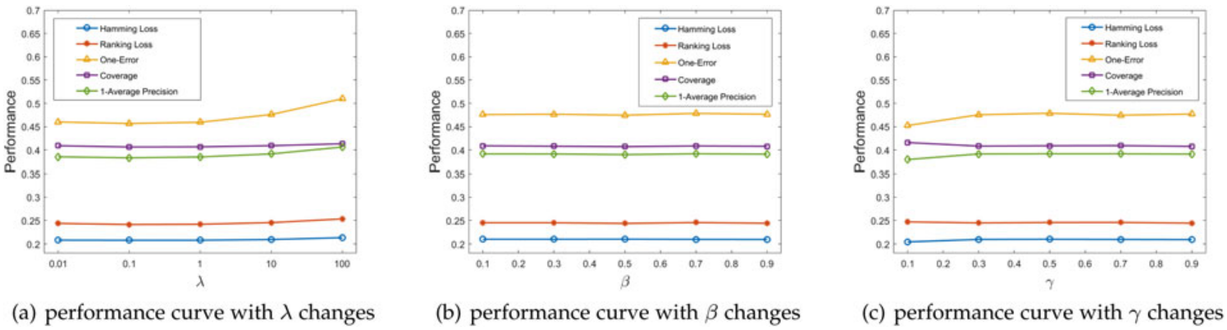
TABLE 3
Friedman Statistics F_F in Terms of Each Evaluation Metric and the Critical Value at 0.05 Significance Level (# comparing algorithms $k = 7$, # data sets $N = 35$)

Evaluation metric	F_F	critical value
Hamming Loss	19.1880	2.2852
Ranking loss	46.1843	
One Error	14.5149	
Coverage	42.3920	
Average Precision	26.2625	



a) Hamming loss b) Ranking loss c) One error d) Coverage e) Average precision

Fig. 2. Comparison of PML-NI (control algorithm) against five comparing algorithms with the Bonferroni-Dunn test. Algorithms not connected with PML-NI in the CD diagram are considered to have a significantly different performance from the control algorithm (CD = 1.3623 at 0.05 significance level).



(a) performance curve with λ changes (b) performance curve with β changes (c) performance curve with γ changes

Fig. 3. Results of PML-NI with varying value of trade-off parameters on *music_emotion*.

be observed that: 1) PML-NI achieves the best (lowest) average rank in terms of all evaluation metrics and significantly outperforms the comparing methods in terms of *one-error* and *average precision*; 2) PML-NI is significantly better than the comparing methods other than PMLLRS in terms of *ranking loss* and *coverage*; 3) PML-NI is significantly better than the comparing methods other than PARVLS and PARMAP in terms of *hamming loss*. These experimental results convincingly validate the significance of the superiority for our PML-NI approach.

5.1.3 Sensitive Analysis

In this section, we study the influences of three balancing parameters, λ, β and γ for the proposed approach on the real-world data sets. We conducted experiments by varying one parameter while keeping the other two parameters fixed. Due to the page limit, we only show the experimental results which are measured by the five evaluation metrics on real-world data set *music_emotion* in Fig. 3, while the results on real-world data set *music_style* are reported on supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3059290. As we can see, in general, performance is not sensitive to the parameters except for the parameter λ, whose performance will be significantly degraded when the value of λ is too large (approximates to 100 in the experiment). Therefore we can safely set them in a wide range in practice.

5.2 Study on MIPML Data

5.2.1 Experimental Setting

MIPML is a new learning framework and there is no method designed specifically for MIPML problems. To show effectiveness of the proposed MIPML-NI[3] method, we compare with multi-instance multi-label methods, which treat all candidate labels as relevant. The following state-of-art methods are compared: MIMLfast [38], EnMIMLNNmetric [52], KISAR [37], DBA [32], MIML-kNN [33] and MIMLSVM [11]. For MIPML-NI, parameter λ is selected from $\{1, 10, 100\}$ by 3-fold cross validation, and the other two parameters are set $\beta = 0.5$ and $\gamma = 0.5$, respectively. In our experiments, although the parameter λ is determined by cross validation, the algorithm is generally not very sensitive to the parameter and it tends to obtain decent performance with a default value, such as $\lambda = 10$. For the other comparing methods, parameters are determined in the same way if no default value given in their literature.

We perform the experiments on totally 7 data sets. Among them, *Scene* and *Reuters* are two benchmark datases which are often used in MIML tasks. *Scene* [53] contains 2,000 images for scene classification with 5 possible class labels. *Reuters* is constructed by using Reuters-21578 data set [54]. *Corel5K* consists 5,000 images and 260 class labels for image classification. The other data sets can be found in [36]. *Letter Frost* and *Letter Carroll* are constructed based on the UCI Letter Recognition dataset [55]. *Bird Song* is constructed for bird audio classification. *MSRC v2* is constructed by using a subset of the Microsoft Research Cambridge (MSRC) image dataset [56].

To construct MIPML data sets based on multi-instance multi-label data, we simulate the annotation process by using a MIMLSVM classifier trained on original supervised

[3] Source code available at: http://milkxie.github.io/code/MIPMLNI code.zip

TABLE 4
Experimental Results of Each Comparing Approach in Terms of *Ranking Loss*, Where ●/○ Indicates Whether MIPML-NI is Superior/Inferior to the Other Method

Data	$\alpha\%$	PML-NI	MIMLfast	EnMIMLNNmetric	KISAR	DBA	MIML-kNN	MIMLSVM
letterF	50%	.163 ± .029	.195 ± .017●	.199 ± .026●	.200 ± .024●	.796 ± .073●	.253 ± .039●	.250 ± .044●
	100%	.185 ± .020	.219 ± .025●	.215 ± .024●	.223 ± .046●	.796 ± .036●	.265 ± .040●	.242 ± .020●
	150%	.197 ± .032	.255 ± .015●	.256 ± .008●	.239 ± .037●	.824 ± .030●	.288 ± .015●	.238 ± .020●
letterC	50%	.160 ± .057	.202 ± .034●	.205 ± .025●	.199 ± .028●	.852 ± .038●	.237 ± .036●	.257 ± .033●
	100%	.178 ± .025	.224 ± .046●	.239 ± .017●	.193 ± .037●	.844 ± .032●	.273 ± .042●	.255 ± .014●
	150%	.181 ± .024	.237 ± .035●	.256 ± .009●	.214 ± .019●	.845 ± .037●	.313 ± .042●	.241 ± .021●
MSRC	50%	.105 ± .015	.174 ± .013●	.124 ± .016●	.118 ± .018●	.679 ± .037●	.188 ± .012●	.115 ± .017●
	100%	.118 ± .008	.184 ± .024●	.152 ± .014●	.126 ± .010●	.664 ± .040●	.193 ± .020●	.118 ± .019
	150%	.138 ± .014	.202 ± .038●	.203 ± .008●	.137 ± .014○	.672 ± .036●	.224 ± .018●	.121 ± .016○
Reuters	50%	.020 ± .003	.045 ± .003●	.029 ± .004●	.022 ± .004●	.075 ± .007●	.028 ± .003●	.034 ± .009●
	100%	.023 ± .003	.050 ± .006●	.041 ± .005●	.023 ± .003○	.073 ± .008●	.032 ± .004●	.036 ± .009●
	150%	.029 ± .007	.071 ± .005●	.083 ± .005●	.031 ± .004●	.093 ± .014●	.043 ± .006●	.063 ± .011●
Bird Song	50%	.071 ± .012	.271 ± .040●	.161 ± .024●	.080 ± .008●	.478 ± .010●	.076 ± .010●	.193 ± .014●
	100%	.080 ± .013	.306 ± .043●	.188 ± .028●	.081 ± .014●	.491 ± .011●	.082 ± .006●	.205 ± .025●
	150%	.088 ± .006	.302 ± .043●	.239 ± .005●	.095 ± .011●	.487 ± .044●	.097 ± .012●	.224 ± .016●
Scene	50%	.181 ± .010	.261 ± .024●	.248 ± .008●	.182 ± .006●	.362 ± .015●	.195 ± .006●	.224 ± .018●
	100%	.190 ± .008	.249 ± .026●	.272 ± .010●	.193 ± .020●	.364 ± .019●	.213 ± .015●	.228 ± .011●
	150%	.203 ± .017	.274 ± .017●	.330 ± .011●	.268 ± .016●	.383 ± .026●	.239 ± .005●	.310 ± .006●
Corel5K	50%	.142 ± .006	.146 ± .007●	.229 ± .004●	.233 ± .009●	.884 ± .006●	.182 ± .005●	.228 ± .009●
	100%	.150 ± .002	.155 ± .003●	.253 ± .007●	.248 ± .006●	.879 ± .004●	.196 ± .010●	.238 ± .003●
	150%	.154 ± .012	.164 ± .004●	.273 ± .006●	.261 ± .006●	.882 ± .007●	.198 ± .008●	.253 ± .004●

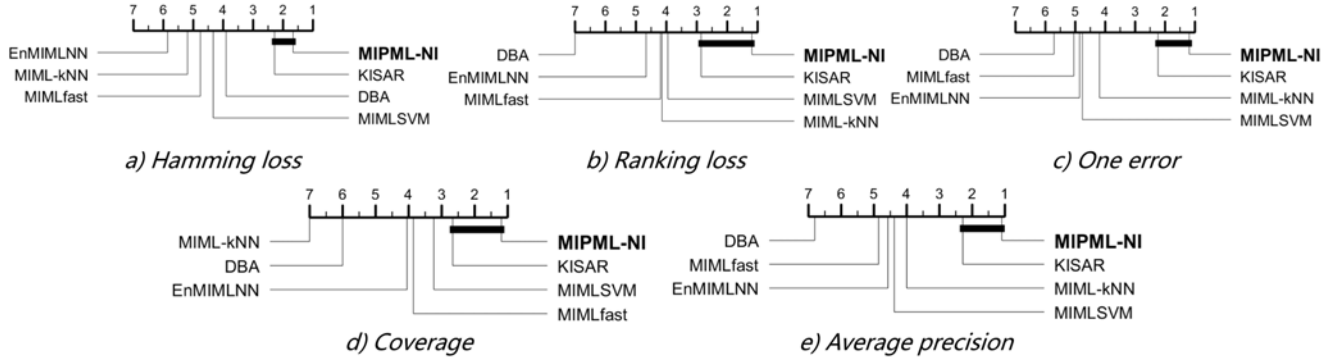
MIML data sets as the human annotator. Specifically, a MIMLSVM classifier is first trained on the MIML data set. Then, for each instance $\mathbf{x}_i$ of the data set, we add irrelevant noisy labels of $\mathbf{x}_i$ with $\alpha\%$ number of ground-truth labels according to their probabilities to be relevant labels predicted by the MIMLSVM classifier and $\alpha\%$ is varied in the range of $\{50\%, 100\%, 150\%\}$.

5.2.2 Comparison Results

Similar to Section 5.1.2, Tables 4 and 5 illustrate detailed results of each comparing method in terms of *ranking loss* and *average precision*, while similar results can be observed in terms of other evaluation metrics. As shown in the table, it can be observed that our method MIPML-NI achieves the best performance in most cases. Among six comparing

TABLE 5
Experimental Results of Each Comparing Approach in Terms of *Average Precision*, Where ●/○ Indicates Whether MIPML-NI is Superior/inferior to the Other Method

Data	$\alpha\%$	PML-NI	MIMLfast	EnMIMLNNmetric	KISAR	DBA	MIML-kNN	MIMLSVM
letterF	50%	.670 ± .049	.591 ± .023●	.599 ± .029●	.632 ± .059●	.313 ± .067●	.531 ± .046●	.517 ± .064●
	100%	.630 ± .048	.560 ± .051●	.572 ± .038●	.595 ± .060●	.319 ± .046●	.518 ± .044●	.491 ± .020●
	150%	.604 ± .054	.518 ± .045●	.502 ± .032●	.546 ± .067●	.282 ± .021●	.457 ± .062●	.442 ± .029●
letterC	50%	.648 ± .100	.571 ± .028●	.584 ± .033●	.607 ± .029●	.287 ± .025●	.537 ± .054●	.497 ± .038●
	100%	.633 ± .047	.551 ± .066●	.541 ± .038●	.602 ± .058●	.282 ± .023●	.484 ± .063●	.486 ± .040●
	150%	.594 ± .063	.534 ± .054●	.485 ± .011●	.544 ± .031●	.298 ± .034●	.432 ± .055●	.483 ± .033●
MSRC	50%	.708 ± .026	.581 ± .015●	.697 ± .037●	.687 ± .043●	.415 ± .033●	.560 ± .018●	.702 ± .027●
	100%	.701 ± .014	.577 ± .045●	.630 ± .029●	.662 ± .019●	.427 ± .037●	.553 ± .044●	.696 ± .018●
	150%	.665 ± .031	.550 ± .040●	.568 ± .026●	.648 ± .042●	.415 ± .031●	.521 ± .032●	.690 ± .020○
Reuters	50%	.961 ± .005	.922 ± .007●	.946 ± .007●	.955 ± .007●	.914 ± .006●	.951 ± .007●	.933 ± .018●
	100%	.957 ± .005	.914 ± .011●	.932 ± .010●	.954 ± .007●	.924 ± .005●	.948 ± .008●	.933 ± .011●
	150%	.948 ± .011	.887 ± .006●	.861 ± .007●	.942 ± .006●	.902 ± .010●	.931 ± .007●	.894 ± .014●
Bird Song	50%	.856 ± .026	.508 ± .043●	.710 ± .031●	.850 ± .014●	.468 ± .034●	.855 ± .013●	.614 ± .027●
	100%	.831 ± .020	.478 ± .056●	.670 ± .040●	.844 ± .026○	.452 ± .030●	.828 ± .013●	.606 ± .039●
	150%	.833 ± .009	.492 ± .057●	.602 ± .024●	.816 ± .020●	.474 ± .062●	.812 ± .019●	.607 ± .019●
Scene	50%	.781 ± .010	.704 ± .026●	.715 ± .010●	.780 ± .009●	.640 ± .012●	.766 ± .005●	.734 ± .018●
	100%	.772 ± .014	.711 ± .031●	.690 ± .015●	.769 ± .020●	.638 ± .019●	.756 ± .017●	.731 ± .006●
	150%	.758 ± .015	.689 ± .017●	.635 ± .013●	.688 ± .012●	.626 ± .016●	.722 ± .012●	.648 ± .007●
Corel5K	50%	.399 ± .006	.308 ± .004●	.307 ± .003●	.346 ± .007●	.076 ± .003●	.339 ± .009●	.324 ± .009●
	100%	.396 ± .008	.300 ± .005●	.277 ± .010●	.336 ± .006●	.079 ± .003●	.326 ± .020●	.317 ± .006●
	150%	.384 ± .008	.297 ± .010●	.252 ± .012●	.325 ± .005●	.077 ± .005●	.323 ± .012●	.306 ± .009●



a) Hamming loss *b) Ranking loss* *c) One error* *d) Coverage* *e) Average precision*

Fig. 4. Comparison of MIPML-NI (control algorithm) against five comparing algorithms with the Bonferroni-Dunn test. Algorithms not connected with MIPML-NI in the CD diagram are considered to have a significantly different performance from the control algorithm (CD = 1.7587 at 0.05 significance level).

methods, KISAR shows some superiority and achieves better performances with two cases in terms of *ranking loss* on *MSRC* and *Reuters*, respectively, as well as one case on *Bird song* in terms of *average precision*, while loss for the other cases. MIMLSVM achieves comparable performance than MIPML-NI on *MSRC*, where MIMLSVM outperforms MIPML-NI with two cases in terms of *ranking loss* and *average precision*, respectively, and is comparable to MIPML-NI with one case, while loss for other cases.

Furthermore, the Friedman test and post-hoc Bonferroni-Dunn are employed to statistically analyze the relative performance among the comparing approaches, respectively. Table 6 reports the Friedman statistics F_F in terms of each evaluation metric (# comparing algorithms $k = 7$, # data sets $N = 21$). From the table, it can be observed that the null hypothesis of indistinguishable performance among the comparing algorithms is clearly rejected at 0.05 significance level. Fig. 4 illustrate the CD diagrams on each evaluation metric (CD = 1.7587 in our experiment: # comparing algorithms $k = 7$, # data set $N = 21$). From the figure, it can be observed that MIPML-NI achieves the best (lowest) average rank and significantly outperforms the comparing methods other than KISAR in terms of all evaluation metrics.

6 Conclusion

In this paper, we disclose the phenomenon that noise labels are usually caused by some ambiguous contents of the example. Based on this observation, we extend our preliminary research [57], and propose to learn partial multi-label problems in a novel strategy by exploiting the potential connections between noisy labels and feature contents. Under the supervision of the observed label matrix, the proposed PML-NI approach jointly learn the multi-label classifier and noisy label identifier by incorporating the label correlation exploitation and feature-induced noise model. Considering multi-instance partial-labeled data, we propose a new learning framework called multi-instance partial multi-label learning and further extend PML-NI into MIPML setting by identifying noisy labels based on ambiguous instances. Theoretical analysis as well as experiments results validate that the proposed approaches are superior to state-of-the-art approaches. In the future, we plan to improve the PML-NI method by considering various forms of noisy labels and utilizing more powerful learning models.

TABLE 6
Friedman Statistics F_F in Terms of Each Evaluation Metric and the Critical Value at 0.05 Significance Level (# comparing algorithms $k = 7$, # data sets $N = 21$)

Evaluation metric	F_F	critical value
Hamming Loss	19.8580	2.4876
Ranking loss	40.2341	
One Error	28.2155	
Coverage	98.5029	
Average Precision	54.4304	

Acknowledgments

This work was supported by the National Key R&D Program of China (2020AAA0107000), NSFC (62076128) and the China University S&T Innovation Plan Guided by the Ministry of Education.

References

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

[2] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, “Multi-label classification of music into emotions,” in *Proc. 9th Int. Conf. Music Inf. Retrieval*, 2008, pp. 325–330.

[3] J. Lin, Q. Su, P. Yang, S. Ma, and X. Sun, “Semantic-unit-based dilated convolution for multi-label text classification,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 4554–4564.

[4] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, “Multi-label image recognition with graph convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5177–5186.

[5] M. Xie and S. Huang, “Partial multi-label learning,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4302–4309.

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

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

[8] Y.-N. Chen and H.-T. Lin, “Feature-aware label space dimension reduction for multi-label classification,” in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1529–1537.

[9] M.-L. Zhang and J.-P. Fang, “Partial multi-label learning via credible label elicitation,” *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 3518–3525, 2019.

[10] T. Wang, C. Lang, L. Sun, S. Feng, and Y. Jin, “Partial multi-label learning by low-rank and sparse decomposition,” in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 5016–5023.

[11] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, “Multi-instance multi-label learning,” *Artif. Intell.*, vol. 176, no. 1, pp. 2291–2320, 2012.

[12] Y. Guo and S. Gu, “Multi-label classification using conditional dependency networks,” in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1300–1305.

[13] S.-J. Huang, Z.-H. Zhou, and Z. Zhou, “Multi-label learning by exploiting label correlations locally,” in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 949–955.

[14] Y. Li, Y. Song, and J. Luo, “Improving pairwise ranking for multi-label image classification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1837–1845.

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

[16] Y. Grandvalet and Y. Bengio, “Learning from partial labels with minimum entropy,” *Cirano Working Papers*, 2004.

[17] R. Jin and Z. Ghahramani, “Learning with multiple labels,” in *Proc. 15th Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 897–904.

[18] B. An and L. Feng, “Partial label learning with self-guided retraining,” in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 3542–3549. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33013542

[19] M. Zhang, B. Zhou, and X. Liu, “Partial label learning via feature-aware disambiguation,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1335–1344.

[20] Y. Yan and Y. Guo, “Partial label learning with batch label correction,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 6575–6582.

[21] F. Yu and M. Zhang, “Maximum margin partial label learning,” *Mach. Learn.*, vol. 106, no. 4, pp. 573–593, 2017.

[22] T. Cour, B. Sapp, and B. Taskar, “Learning from partial labels,” *J. Mach. Learn. Res.*, vol. 12, pp. 1501–1536, 2011.

[23] E. Hüllermeier and J. Beringer, “Learning from ambiguously labeled examples,” *Lecture Notes Comput. Sci.*, vol. 10, no. 5, pp. 419–439, 2006.

[24] M. Zhang and F. Yu, “Solving the partial label learning problem: An instance-based approach,” in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 4048–4054.

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

[26] Q. Zhang, Y. Zhong, and M. Zhang, “Feature-induced labeling information enrichment for multi-label learning,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4446–4453.

[27] Y. Yan and Y. Guo, “Adversarial partial multi-label learning,” 2019, *arXiv: 1909.06717*.

[28] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, “Discriminative and correlative partial multi-label learning,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3691–3697.

[29] S. He, K. Deng, L. Li, S. Shu, and L. Liu, “Discriminatively relabel for partial multi-label learning,” in *Proc. IEEE Int. Conf. Data Mining*, 2019, pp. 280–288.

[30] Z.-S. Chen, X. Wu, Q.-G. Chen, Y. Hu, and M.-L. Zhang, “Multi-view partial multi-label learning with graph-based disambiguation,” in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 3553–3560.

[31] M.-K. Xie and S.-J. Huang, “Semi-supervised partial multi-label learning,” in *Proc. IEEE Int. Conf. Data Mining*, 2020, pp. 691–700.

[32] S.-H. Yang, H. Zha, and B.-G. Hu, “Dirichlet-bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora,” in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 2143–2150.

[33] M.-L. Zhang, “A k-nearest neighbor based multi-instance multi-label learning algorithm,” in *Proc. 22nd IEEE Int. Conf. Tools Artif. Intell.*, 2010, vol. 2, pp. 207–212.

[34] M.-L. Zhang and Z.-J. Wang, “MIMLRBF: RBF neural networks for multi-instance multi-label learning,” *Neurocomputing*, vol. 72, no. 16/18, pp. 3951–3956, 2009.

[35] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, and Z. Wang, “Joint multi-label multi-instance learning for image classification,” in *Proc. 2008 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[36] F. Briggs, X. Z. Fern, and R. Raich, “Rank-loss support instance machines for MIML instance annotation,” in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 534–542.

[37] Y.-F. Li, J.-H. Hu, Y. Jiang, and Z.-H. Zhou, “Towards discovering what patterns trigger what labels,” in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1012–1018.

[38] S.-J. Huang, W. Gao, and Z.-H. Zhou, “Fast multi-instance multi-label learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2614–2627, Nov. 2019.

[39] A. T. Pham, R. Raich, and X. Z. Fern, “Dynamic programming for instance annotation in multi-instance multi-label learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2381–2394, Dec. 2017.

[40] J. Feng and Z.-H. Zhou, “Deep MIML network,” in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1884–1890.

[41] C.-T. Nguyen, X. Wang, J. Liu, and Z.-H. Zhou, “Labeling complicated objects: Multi-view multi-instance multi-label learning,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2013–2019.

[42] Y. Zhu, K. M. Ting, and Z.-H. Zhou, “Discover multiple novel labels in multi-instance multi-label learning,” in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2977–2983.

[43] Y. Pei and X. Z. Fern, “Constrained instance clustering in multi-instance multi-label learning,” *Pattern Recognit. Lett.*, vol. 37, pp. 107–114, 2014.

[44] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Found. Comput. Math.*, Springer, vol. 9, no. 6, pp. 717–772, 2009.

[45] E. Candes and T. Tao, “Decoding by linear programming,” 2005, *arXiv preprint math/0502327*.

[46] Z. Lin, M. Chen, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” 2010, *CoRR*, vol. abs/1009.5055, 2010. [Online]. Available: http://arxiv.org/abs/1009.5055

[47] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[48] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, Massachusetts, USA: MIT Press, 2018.

[49] A. Maurer, “A vector-contraction inequality for rademacher complexities,” in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2016, pp. 3–17.

[50] G. Yu *et al.*, “Feature-induced partial multi-label learning,” in *Proc. IEEE Int. Conf. Data Mining*, 2018, pp. 1398–1403.

[51] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[52] J.-S. Wu, S.-J. Huang, and Z.-H. Zhou, “Genome-wide protein function prediction through multi-instance multi-label learning,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 5, pp. 891–902, Sep./Oct. 2014.

[53] Z.-H. Zhou and M.-L. Zhang, “Multi-instance multi-label learning with application to scene classification,” in *Proc. 19th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1609–1616.

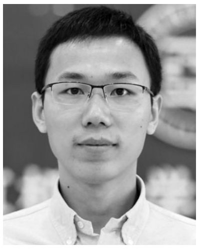
[54] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[55] P. W. Frey and D. J. Slate, “Letter recognition using holland-style adaptive classifiers,” *Mach. Learn.*, vol. 6, no. 2, pp. 161–182, 1991.

[56] J. Winn, A. Criminisi, and T. Minka, “Object categorization by learned universal visual dictionary,” in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1800–1807.

[57] M.-K. Xie and S.-J. Huang, “Partial multi-label learning with noisy label identification,” in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 6454–6461.

Ming-Kun Xie received the BSc degree, in 2018. He is currently working toward the master’s degree from the MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing University of Aeronautics and Astronautics, China. He has served as a PC member of IJCAI 2020 and AAAI 2021, also a reviewer of the *IEEE Transactions on Neural Networks and Learning Systems*. His research interests includes machine learning. Particularly, he is interested in multi-label learning and weakly-supervised learning.



Sheng-Jun Huang received the BSc and PhD degrees in computer science from Nanjing University, China, in 2008 and 2014, respectively. He is currently a professor at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His main research interests include machine learning and data mining. He has been selected to the Young Elite Scientists Sponsorship Program by CAST, in 2016, and won the China Computer Federation Outstanding Doctoral Dissertation Award, in 2015, the KDD Best Poster Award at the, in 2012, and the Microsoft Fellowship Award, in 2011. He is a junior associate editor of the *Frontiers of Computer Science*.