Partial Multi-Label Learning with Meta Disambiguation

Ming-Kun Xie*
Nanjing University of Aeronautics and Astronautics
Nanjing, China
mkxie@nuaa.edu.cn

Feng Sun*
Nanjing University of Aeronautics and Astronautics
Nanjing, China
sunfeng@nuaa.edu.cn

Sheng-Jun Huang†
Nanjing University of Aeronautics and Astronautics
Nanjing, China
huangsj@nuaa.edu.cn

ABSTRACT

In partial multi-label learning (PML) problems, each instance is partially annotated with a candidate label set, which consists of multiple relevant labels and some noisy labels. To solve PML problems, existing methods typically try to recover the ground-truth information from partial annotations based on extra assumptions on the data structures. While the assumptions hardly hold in real-world applications, the trained model may not generalize well to varied PML tasks. In this paper, we propose a novel approach for partial multi-label learning with meta disambiguation (PML-MD). Instead of relying on extra assumptions, we try to disambiguate between ground-truth and noisy labels in a meta-learning fashion. On one hand, the multi-label classifier is trained by minimizing a confidence-weighted ranking loss, which distinctively utilizes the supervised information according to the label quality; on the other hand, the confidence for each candidate label is adaptively estimated with its performance on a small validation set. To speed up the optimization, these two procedures are performed alternately with an online approximation strategy. Comprehensive experiments on multiple datasets and varied evaluation metrics validate the effectiveness of the proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

KEYWORDS

partial multi-label learning, candidate label set, disambiguation, ranking loss, meta learning

ACM Reference Format:
Ming-Kun Xie, Feng Sun, and Sheng-Jun Huang. 2021. Partial Multi-Label Learning with Meta Disambiguation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467259

*Both authors contributed equally to this research.
†Correspondence to: Sheng-Jun Huang.

KDD '21, August 14–18, 2021, Virtual Event, Singapore.

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467259

Figure 1: An example of partial multi-label learning. The image is partial-labeled by annotators with different level of expertise on the crowdsourcing platform. In the candidate label set, building, tree, window, light and bicycle are ground-truth labels while people, cloud and flower are noisy labels.

1 INTRODUCTION

In multi-label learning problems, each instance is associated with multiple class labels simultaneously [45]. For example, a fragment of movie may consist of multiple characters [22]; a piece of document can be categorized into multiple topics [21]; and an image may be annotated with multiple tags [4]. The goal of multi-label learning is to train a classification model based on precisely annotated training data that can accurately predict all the relevant labels for an unseen example.

Typical multi-label learning methods assume that each instance has been precisely annotated with all of its relevant labels. Unfortunately, in many real-world scenarios, instead of ground-truth labels, one can only get access to a candidate label set, which contains multiple relevant labels and some other noisy labels. For instance, as shown in Figure 1, on the crowdsourcing platform, labelers with different level of expertise may annotate the image with multiple candidate labels, among which only some of them are correct ones. The problem has been recently formulized as a learning framework called partial multi-label learning (PML) [32, 35, 42].

In order to solve PML problems, a straightforward method is to transform the original problem to a standard multi-label learning problem by simply treating all candidate labels as relevant ones. Then, one can employ off-and-shelf multi-label learning methods to solve the transformed problems. Unfortunately, such methods may be seriously misled by the noisy labels in the candidate set.

Recently, a number of approaches have been proposed to improve the practical performance of PML. These methods usually tackle the partial-labeled data with disambiguation strategy, which recovers the ground-truth labeling information from the candidate labels. The main idea is to maintain a confidence for each candidate label

to measure how likely it can be a ground-truth label based on the structure information of data. For example, some methods focus on the sparsity assumption of the noisy labels, which utilizes the sparsity constraint to recover either the noise label matrix [27] or weight matrix of the noisy label identifier [35]. The smooth assumption is utilized either to achieve credible label elicitation [42] or perform label enhancement [36] while the label correlation is employed to recover label confidences [32, 36].

Despite the advances these methods have achieved, a potential limitation is that they perform disambiguation based on extra assumptions, which hardly hold in many real-world scenarios. For example, in practice, instead of sparsity, the candidate set may consist of a number of noisy labels, since the ground-truth labeling information may be corrupted heavily in the extreme case. The smooth assumption based on Euclidean distance may be ineffective in the high-dimensional feature space.

To tackle these challenges, in this paper, we argue that in order to accurately recover the ground-truth information, it is more effective and reliable to perform disambiguation in a meta learning fashion by utilizing a small validation set. Specifically, a confidence-weighted ranking loss is developed to train the multi-label classifier on the partial-labeled data, where the confidences measure the relevance orders of each label pair. To achieve disambiguation, we adaptively estimate the confidence for each candidate label guided by its performance on the validation set. These two procedures are performed iteratively by using the online approximation strategy to save the computational cost. Our empirical studies demonstrate that the proposed method can outperform state-of-the-art methods on different datasets with regard to various evaluation metrics.

2 RELATED WORK

Partial multi-label learning is a recently proposed weakly-supervised learning framework, which is particularly relevant to two popular learning scenarios: multi-label learning [45] and partial label learning [6].

Multi-label learning (MLL) studies the problem where each instance is associated with a set of valid labels simultaneously. Based on the order of label correlations [45] that learning techniques have exploited, existing multi-label learning approaches can be roughly classified into three categories: first-order, second-order and high-order methods. The first-order is the most straightforward approach for multi-label learning, which decomposes the multi-label learning task into a series of independent binary classification problems [3, 44]. However, the first-order approaches ignore the label correlations, which are fundamental information for solving multi-label learning problems. The second-order approaches tackle the multi-label learning task by considering the correlations between two labels [8, 14]. The high-order approaches consider the high-order correlation among all labels [5, 15, 16], which has stronger correlation-modeling capabilities than first-order and second-order approaches. The purposes of both MLL and PML are to learn a multi-label model which can predict all relevant labels for an unseen instance. However, the task of PML is more challenging than MLL because only the candidate label set with ambiguous information is given in PML setting.

Partial label learning (PLL) studies the problem where each instance is associated with a set of candidate labels, but only one of them is valid. Existing partial label learning approaches can be roughly grouped into two families. One aims to disambiguate the candidate label set [6, 9, 10, 38, 39], the other tries to transform partial labeling problems into traditional supervised learning problems [31, 43]. Recently, some works study PLL problems from the theoretical perspective. In [19], authors study the learnability of PLL problems. Consistent PLL learning methods are proposed in [11]. Both PLL and PML aim to learn a classification model from partially labeled training examples. Nevertheless, the task of PML is more challenging than PLL, because there are multiple ground-truth labels in the candidate label set for PML problems.

In order to solve partial multi-label learning problems, the most commonly used technique is the disambiguation, which recovers the ground-truth labels from the candidate label set. For instance, in [32], two effective methods PML-*lc* and PML-*fp* are proposed to learn the multi-label classifier and estimate the confidences simultaneously. Low-rank matrix approximation technique is used to recover the ground-truth labeling information in [40]. In [27], authors propose to achieve disambiguation by decomposing the observed matrix into two matrices, i.e., the ground-truth label matrix and noisy label matrix. A two-stage method PARTICLE [42] first recovers the credible labels with high labeling confidences and then utilizes an iterative label propagation procedure to train a multi-label classifier with credible labels. In [35], authors first consider the generation process of noisy labels in the candidate set and utilize this information to solve PML problems. In [37], authors borrow the idea of adversarial training to solve PML problems. DRAMA proposes to employ the gradient boosting model to fit the label confidences learned with smooth assumption [30]. The label compression technique is used to solve PML problems [41]. In [20], authors utilize the graph matching mechanism to recover ground-truth labels. The unbiased risk estimator for PML is proposed in [34]. Authors first extend PML into semi-supervised learning setting in [33]. Despite the advances these methods have achieved, a potential limitation is that most of these methods achieve disambiguation based on extra assumptions, which cannot hold in practice.

Our approach achieves disambiguation for candidate labels by adaptively estimating confidences in a meta learning manner [1, 18, 29]. The main idea of meta learning is to optimize a meta-loss on the validation set, which has been widely used in few-shot learning scenarios [23, 24, 28], where only a few of labeled exampels for each class label. Similar to previous works [12, 25, 26], the proposed method performs the meta-objective optimization by using the online approximation strategy. However, different from these methods, our method first considers learning to disambiguate for candidate labels.

3 THE PROPOSED APPROACH

For each partial-labeled example, let $\boldsymbol{x} \in \mathcal{X}$ be a feature vector and $\hat{\boldsymbol{y}} \subseteq \mathcal{Y}$ be its corresponding candidate label set, where $\mathcal{X} \subset \mathbb{R}^d$ is the feature space and $\mathcal{Y} = \{1, 2, ..., q\}$ is the target space with q possible class labels. Note that in our setting, besides relevant labels, the candidate label set may contain multiple noisy labels, i.e., for each instance, we have a ground-truth label set $\boldsymbol{y} \subseteq \hat{\boldsymbol{y}}$. We

denote by $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ the output of the neural network for instance $\boldsymbol{x}$, where $\boldsymbol{\theta}$ is the net parameter. Let $f_j(\boldsymbol{x}, \boldsymbol{\theta})$ be the j-th component of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$. Our goal is to learn a multi-label classifier based on a given training set $D = \{(\boldsymbol{x}_1, \hat{\boldsymbol{y}}_1), (\boldsymbol{x}_2, \hat{\boldsymbol{y}}_2), ..., (\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)\}$ with n examples that can accurately predict all the ground-truth labels for an unseen instance. In the following content, we will primarily introduce the objective function based on confidence-weighted ranking loss for tackling partial-labeled data; then, we discuss how to adaptively estimate confidences for candidate labels in a meta-learning manner; finally, to save the computational cost, we propose to use the online approximation strategy for optimizing the parameters iteratively.

3.1 The objective Function

In order to solve multi-label learning problems, one intuitive method is to decompose the task into q independent binary classification problems, where q is the number of class labels. Unfortunately, such methods neglect the correlation among labels, which is a fundamental information for solving multi-label problems. Among various multi-label losses, the ranking loss focuses on the relevance orders of label pairs, which considers the second-order label correlation. Given a real-valued decision function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) = \{f_1(\boldsymbol{x}, \boldsymbol{\theta}), f_2(\boldsymbol{x}, \boldsymbol{\theta}), ..., f_q(\boldsymbol{x}, \boldsymbol{\theta})\}$, where the corresponding net parameter is $\boldsymbol{\theta}$, the ranking loss can be defined as:

$$l(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}), \hat{\boldsymbol{y}}) = \sum_{j \in \hat{\boldsymbol{y}}} \sum_{k \notin \hat{\boldsymbol{y}}} I[f_j(\boldsymbol{x}, \boldsymbol{\theta}) < f_k(\boldsymbol{x}, \boldsymbol{\theta})], \tag{1}$$

where $I(\cdot)$ is the indicator function, which outputs 1 if the condition holds while 0, otherwise. Unfortunately, it is difficult to directly optimize the loss function defined in Eq.(1), since it usually yields a NP-hard problem owing to its non-convexity and discontinuity. To tackle the challenge, a feasible solution in practice is to alternatively consider a convex surrogate loss, which can be optimized efficiently. With respective to the ranking loss, a common choice of surrogate loss is the following hinge loss, which has been shown as an optimal choice among all convex surrogate losses [2]:

$$\mathcal{L}(D, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \in \hat{\boldsymbol{y}}} \sum_{k \notin \hat{\boldsymbol{y}}} \ell\left(f_j(\boldsymbol{x}_i, \boldsymbol{\theta}) - f_k(\boldsymbol{x}_i, \boldsymbol{\theta})\right), \tag{2}$$

where $\ell(z) = \max(0, 1 - z)$ is the hinge loss.

However, the objective function defined in Eq.(2) simply treats all candidate labels as relevant ones, which may be misled by false positive labels in the candidate label set and obtains a classification model with the poor generalization performance. To solve the problem, for instance $\boldsymbol{x}_i$, we introduce a label confidence vector $\boldsymbol{p}_i$, where the confidence $p_{ij} \in [0, 1]$ estimates how likely the j-th label can be a ground-truth label for instance $\boldsymbol{x}_i$. Note that for the non-candidate label j, since it is for sure irrelevant to instance $\boldsymbol{x}_i$, its confidence always satisfies $p_{ij} = 0$. Now, for the label pair (j, k) with $j \in \hat{\boldsymbol{y}}_i$ and $k \notin \hat{\boldsymbol{y}}_i$, without loss of generality, we assume that j is a false positive label, i.e., $j \notin \boldsymbol{y}_i$, where $\boldsymbol{y}_i$ is the relevant label set of $\boldsymbol{x}_i$. In such case, based on Eq.(2), the loss is over-estimated due to excessive loss calculated on label pair (j, k), since their loss should have been 0. To calibrate losses calculated by Eq.(2), we propose to re-weight the loss of each label pair (j, k) by their confidence difference $\Delta_{jk}^i = \max(0, p_{ij} - p_{ik})$, which measures their relevance ordering to instance $\boldsymbol{x}_i$. Here, the operator $\max(0, \cdot)$ is used to make sure that the loss is non-negative and we omit the superscript i when it is clear for the context. In this paper, the procedure is called disambiguation strategy for candidate labels. Finally, the objective function can be written as

$$\mathcal{L}_{train}(D, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \in \hat{\boldsymbol{y}}} \sum_{k \notin \hat{\boldsymbol{y}}} \Delta_{jk} \ell\left(f_j(\boldsymbol{x}_i, \boldsymbol{\theta}) - f_k(\boldsymbol{x}_i, \boldsymbol{\theta})\right). \tag{3}$$

Algorithm 1 Partial Multi-Label Learning with Meta Disambiguation

Input:
  D: the training dataset
  D_{val}: the validation dataset
  T: the max iteration
Process:
  Initialize the net parameter $\boldsymbol{\theta}$ and confidence $\boldsymbol{\rho}$.
  For: $t = 1 : T$
    Sample a minibatch of training samples $D_b = \{(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i)\}_{i=1}^{b}$ from training set D.
    Update $\hat{\boldsymbol{\theta}}^{(t)}$ with Eq.(5)
    Update $\boldsymbol{\rho}^{(t)}$ with Eq.(6-8),
    Update $\boldsymbol{\theta}^{(t)}$ with Eq.(9)
  End For
Output: the net parameter $\boldsymbol{\theta}$ and confidences $\boldsymbol{\rho}$.

3.2 Meta Disambiguation

As mentioned in Section 1, the performance of existing PML methods relies on the specific assumption, which can capture the structure information of data. However, in practice, such methods may suffer the poor generalization performance, since the unknown structure information can be varied in diverse datasets, which is hard to be modeled with the specific assumption. In order to tackle the challenge, different from previous methods, we propose a general framework to achieve disambiguation for candidate labels in a meta-learning fashion [25, 26].

Specifically, besides the training dataset, we assume that a small validation set $D_{val} = \{(\boldsymbol{x}_i^v, \boldsymbol{y}_i^v)\}_{i=1}^{m}$ with m examples is available during the training phase, where it satisfies $m \ll n$. Note for each example in the validation set, all of its relevant labels have been precisely annotated in advance. In the experiment section, we will further study the influence of the size of validation set on the performance of the proposed method. For notational simplicity, let $\boldsymbol{\rho} = \{\boldsymbol{p}_i\}_{i=1}^{n}$ denote the set of confidences for all training examples. Intuitively, for the label pair (j, k), the optimal confidences can precisely measure their relevance ordering, which leads to a small loss with Eq.(3) on the clean validation data. The observation tells us that the optimal $\boldsymbol{\rho}^*$ can be determined by its performance on the validation set, which can be formulated as a meta-objective function:

$$\mathcal{L}_{val}(D_{val}, \boldsymbol{\theta}^*(\boldsymbol{\rho})) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j \in \hat{\boldsymbol{y}}} \sum_{k \notin \hat{\boldsymbol{y}}} \ell\left(f_j(\boldsymbol{x}_i^v, \boldsymbol{\theta}^*(\boldsymbol{\rho})) - f_k(\boldsymbol{x}_i^v, \boldsymbol{\theta}^*(\boldsymbol{\rho}))\right), \tag{4}$$

where $\mathcal{L}_{val}$ represents the loss with respective to confidence $\boldsymbol{\rho}$ by calculating with Eq.(3) on validation set D_{val}.

In order to obtain the optimal parameter $\boldsymbol{\theta}^*$ and confidence $\boldsymbol{\rho}^*$, a straightforward method is to employ an alternating optimization procedure, which optimizes $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$ alternatively until both of them converge or exceed the maximal iteration. Specifically, it requires an external loop consists of two internal loops, where one for optimizing $\boldsymbol{\theta}$ by minimizing the objective function (Eq.(3)) at training step t:

$$\boldsymbol{\theta}^{(t)} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{train}(D, \boldsymbol{\theta}),$$

while the other for optimizing $\boldsymbol{\rho}$ by minimizing the meta-objective function (Eq.(4)):

$$\boldsymbol{\rho}^{(t)} = \arg\min_{\boldsymbol{\rho}} \mathcal{L}_{val}(D_{val}, \boldsymbol{\theta}^{(t)}(\boldsymbol{\rho})).$$

However, such method is infeasible in practice due to its high computational complexity, since both of these two procedures can be very expensive.

3.3 Online Approximation Optimization

Inspired by previous works [12, 13, 25, 26], to guarantee the efficiency for updating $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$, we utilize an online approximation strategy [12, 25] to update two sets of parameters simultaneously in an iterative fashion. Specifically, at every t iteration of training, we usually sample a mini-batch of training examples $D_b = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{b}$ to update the net parameter $\boldsymbol{\theta}$, where b is the size of mini-batch. Based on the parameter $\boldsymbol{\theta}^{(t-1)}$ on the last iteration, the procedure of updating parameter $\hat{\boldsymbol{\theta}}^{(t)}$ can be formulated as the gradient decent toward the direction of the expected loss on the mini-batch with respective to $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}}^{(t)}(\boldsymbol{\rho}) = \boldsymbol{\theta}^{(t-1)} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{train}(D_b, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}^{(t-1)}}, \tag{5}$$

where α is the learning rate.

For each label pair (j, k), the performance on the validation set depends on their relevance ordering of confidences at training step t. Following the main idea of previous works[12, 17], based on the parameter $\hat{\boldsymbol{\theta}}^{(t)}$, we can obtain the confidence $\hat{\boldsymbol{\rho}}^{(t)}$ by taking a single gradient descent step toward the direction of the expected meta-loss on the validation set with respective to $\boldsymbol{\rho}$.

$$\hat{\boldsymbol{\rho}}^{(t)} = \boldsymbol{\rho}^{(t-1)} - \beta \nabla_{\boldsymbol{\rho}} \mathcal{L}_{val}(D_{val}, \boldsymbol{\theta}^{(t)}(\boldsymbol{\rho}))\big|_{\boldsymbol{\rho}^{(t-1)}}, \tag{6}$$

where β is the decent step size on $\boldsymbol{\rho}$. To guarantee a non-negative loss, we need to make sure that each confidence $\hat{\boldsymbol{p}}_i^{(t)} \in \hat{\boldsymbol{\rho}}^{(t)}$ is non-negative:

$$\hat{p}_{ij}^{(t)} = \max(\hat{p}_{ij}^{(t)}, 0), \forall j \in [q]. \tag{7}$$

Then, we further normalize $\hat{p}_{ij}^{(t)}$ into $[0, 1]$ to obtain $p_{ij}^{(t)}, \forall i \in [n], j \in [q]$:

$$p_{ij}^{(t)} = \frac{\hat{p}_{ij}}{\max_j \hat{p}_{ij}}. \tag{8}$$

Accordingly, let $\boldsymbol{\rho}^{(t)} = \{\boldsymbol{p}_i^{(t)}\}_{i=1}^{n}$ denote the set of updated confidences at training step t.

Finally, guided by the newly updated confidences $\boldsymbol{\rho}^{(t)}$, we can update the net parameter $\boldsymbol{\theta}^{(t)}$ by using the gradient decent as follows:

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{train}(D_b, \boldsymbol{\theta}(\boldsymbol{\rho}^{(t)}))\big|_{\boldsymbol{\theta}^{(t-1)}}. \tag{9}$$

Table 1: Characteristics of the Experimental Datasets.

Data	# ins	# fea	# label	card	batch size	domain
music_emotion	6833	98	11	2.42	200	music
mirflickr	10433	100	7	1.77	500	image
medical	978	1449	45	1.25	100	text
enron	1702	1001	53	4.27	100	text
image	2000	294	5	1.24	100	image
scene	2407	294	6	1.07	100	image
yeast	2417	103	14	4.24	100	biology
slashdot	3782	1079	22	1.18	200	text
tmc	15000	500	22	2.23	500	text
mediamill	15000	120	50	4.22	500	video

We repeat these procedures until it exceeds the maximal iteration. The main procedures of PML-MD are summarized in Algorithm 1.

4 EXPERIMENTS

4.1 Experimental Setting

We perform experiments on totally 10 datasets [1], including 2 real-world PML datasets and 8 synthetic datasets. These datasets focus on a large range of applications: image, scene and mirflickr for image annotation, medical, slashdot, enron and tmc for text categorization, music_emotion for music recognition, mediamill for video annotation as well as yeast for protein function prediction. The detailed characteristics of these datasets are summarized in Table 1. For each dataset, we randomly sample 80% data for training and 20% data for testing. In our experiments, we assume that the size of validation set is the same as the mini-batch size. We additionally sample a mini-batch size of examples from training data as the validation set.

There are different criteria for evaluating the performance of multi-label learning. In our experiments, we employ five commonly used criteria, including *hamming loss*, *ranking loss*, *one error*, *coverage* and *average precision*. More details about these evaluation metrics can be found in [45]. For the hamming loss, ranking loss, one error and coverage metrics, the smaller the 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 method, we compare two variants of PML-MD:

- PML-MD. It trains a multi-label classifier with Algorithm 1 by using SGD method.
- PML-MD+. It also trains a multi-label classifier with Algorithm 1. Different from PML-MD, it additionally uses the validation set to update the net parameter $\boldsymbol{\theta}$.

with five state-of-the-art PML approaches: PML-NI [35], PML-LRS [27], fPML [40], PARMAP [42] and PARVLS [42]. as well as the baseline method:

- Baseline. It trains a multi-label classifier with the ranking loss defined as Eq.(2) by using SGD method on both training and validation data.

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

Table 2: Comparison results between the proposed method and comparing methods under low-level label noise, where •/○ indicates whether the proposed method is significantly superior/inferior to the comparing methods via paired t-test (at 0.05 significance level).

Data	PML-MD	PML-MD+	Baseline	PMLNI	PMLLRS	PARMAP	PARVLS	fPML
Hamming loss (the smaller, the better)								
medical	.022 ± .002•	**.017 ± .000**	.020 ± .000•	.021 ± .002	.024 ± .001•	.035 ± .002•	.032 ± .001•	.027 ± .000•
enron	**.070 ± .009**	.073 ± .004	.074 ± .006	.082 ± .002	.173 ± .007•	.137 ± .010•	.132 ± .032•	.112 ± .008•
image	.206 ± .008	.208 ± .004	.204 ± .004	.232 ± .023•	.209 ± .016	**.203 ± .011**	.317 ± .028•	.248 ± .004•
scene	.147 ± .012	.141 ± .003	**.139 ± .004**	.169 ± .016•	.139 ± .004	.232 ± .038•	.167 ± .018•	.179 ± .002•
yeast	.244 ± .010•	.268 ± .013•	.291 ± .010•	.238 ± .012•	.215 ± .004	.216 ± .003	**.202 ± .005**	.307 ± .002•
slashdot	.018 ± .002	**.016 ± .000**	.020 ± .002•	.020 ± .002•	.021 ± .003•	.018 ± .000•	.077 ± .004•	.054 ± .000•
tmc	.068 ± .000•	.066 ± .001	.070 ± .001•	**.066 ± .000**	.074 ± .001•	.082 ± .001•	.068 ± .000•	.101 ± .000•
mediamill	.086 ± .004•	.080 ± .002	.079 ± .002	**.065 ± .009**	.076 ± .001	.066 ± .003	.069 ± .000	.088 ± .000•
Ranking loss (the smaller, the better)								
medical	.056 ± .012	**.055 ± .010**	.076 ± .012•	.083 ± .020	.105 ± .021•	.067 ± .009	.199 ± .025•	.098 ± .014•
enron	**.165 ± .012**	.170 ± .003	.212 ± .028•	.293 ± .016•	.293 ± .029•	.204 ± .005•	.404 ± .047•	.305 ± .025•
image	**.194 ± .009**	.199 ± .010	.200 ± .018	.239 ± .032	.219 ± .025	.220 ± .014•	.448 ± .048•	.287 ± .021•
scene	**.092 ± .006**	.097 ± .004	.119 ± .009•	.163 ± .011•	.145 ± .018•	.262 ± .057•	.192 ± .035•	.149 ± .009•
yeast	**.174 ± .007**	.185 ± .010•	.182 ± .007	.210 ± .014•	.203 ± .010•	.184 ± .003	.197 ± .005•	.212 ± .015•
slashdot	**.038 ± .007**	.043 ± .007•	.067 ± .005•	.085 ± .011•	.077 ± .007•	.050 ± .005•	.267 ± .083•	.085 ± .009•
tmc	.056 ± .001	**.054 ± .001**	.084 ± .010•	.086 ± .002•	.085 ± .007•	.106 ± .004•	.099 ± .002•	.098 ± .007•
mediamill	.080 ± .002•	.071 ± .001	**.070 ± .000**	.154 ± .011•	.155 ± .011•	.116 ± .004•	.139 ± .003•	.164 ± .014•
One error (the smaller, the better)								
medical	**.247 ± .034**	.266 ± .034	.339 ± .023•	.410 ± .067•	.423 ± .034•	.295 ± .023	.623 ± .036•	.318 ± .009•
enron	.660 ± .036	.677 ± .015	.723 ± .028	.707 ± .037	.705 ± .021	**.576 ± .034**	.758 ± .130	.628 ± .043○
image	.358 ± .015	**.354 ± .019**	.359 ± .029	.413 ± .039•	.412 ± .040•	.393 ± .030	.621 ± .051•	.475 ± .022•
scene	**.269 ± .023**	.284 ± .011	.288 ± .008	.385 ± .019•	.379 ± .014•	.465 ± .038•	.333 ± .051	.364 ± .028•
yeast	.225 ± .014	.244 ± .027	.234 ± .016	.266 ± .008•	.239 ± .016	.231 ± .008	**.201 ± .018**	.235 ± .012
slashdot	.321 ± .014•	.326 ± .021•	.329 ± .016•	.114 ± .020	.134 ± .018	**.074 ± .003**	.875 ± .166•	.096 ± .009
tmc	.238 ± .010•	.230 ± .006•	.214 ± .003	.219 ± .004	.217 ± .004	.271 ± .003•	**.210 ± .004**	.241 ± .006•
mediamill	.185 ± .012•	.172 ± .002•	.174 ± .004•	**.128 ± .003**	.131 ± .004	.140 ± .007	.195 ± .003•	.145 ± .008•
Coverage (the smaller, the better)								
medical	.077 ± .015	**.076 ± .014**	.101 ± .015•	.105 ± .022	.127 ± .020•	.083 ± .012	.219 ± .027•	.121 ± .018•
enron	**.319 ± .012**	.330 ± .010	.372 ± .028•	.498 ± .014•	.502 ± .040•	.414 ± .012•	.623 ± .049•	.539 ± .029•
image	**.210 ± .005**	.217 ± .006	.211 ± .015	.247 ± .024	.224 ± .019	.212 ± .011	.359 ± .059•	.287 ± .023•
scene	**.091 ± .006**	.096 ± .004	.115 ± .012•	.153 ± .012•	.137 ± .017•	.239 ± .050•	.152 ± .037•	.141 ± .006•
yeast	**.458 ± .007**	.473 ± .008	.484 ± .021•	.517 ± .022•	.523 ± .026•	.494 ± .009•	.490 ± .009•	.530 ± .024•
slashdot	**.047 ± .009**	.056 ± .011•	.087 ± .008•	.127 ± .016•	.117 ± .012•	.084 ± .007•	.269 ± .071•	.124 ± .013•
tmc	.141 ± .000	**.140 ± .003**	.194 ± .017•	.199 ± .004•	.199 ± .012•	.229 ± .007•	.205 ± .003•	.216 ± .014•
mediamill	.265 ± .007•	.240 ± .003	**.239 ± .002**	.434 ± .026•	.440 ± .032•	.339 ± .007•	.396 ± .009•	.458 ± .034•
Average precision (the greater, the better)								
medical	**.801 ± .029**	.772 ± .023	.700 ± .020•	.663 ± .066•	.635 ± .032•	.737 ± .025	.445 ± .036•	.712 ± .007•
enron	**.561 ± .025**	.557 ± .003	.512 ± .031	.332 ± .022•	.327 ± .023•	.417 ± .012•	.255 ± .081•	.350 ± .030•
image	**.764 ± .011**	.763 ± .010	.761 ± .018	.728 ± .028	.736 ± .026•	.743 ± .015•	.559 ± .061•	.679 ± .013•
scene	**.836 ± .013**	.827 ± .005	.814 ± .006•	.755 ± .013•	.766 ± .013•	.671 ± .043•	.748 ± .044•	.770 ± .013•
yeast	**.754 ± .011**	.737 ± .016	.747 ± .008	.720 ± .009•	.735 ± .006	.742 ± .003	.747 ± .007	.732 ± .015
slashdot	**.913 ± .009**	.912 ± .007	.885 ± .008•	.844 ± .015•	.842 ± .016•	.889 ± .005•	.271 ± .142•	.855 ± .008•
tmc	.788 ± .005•	**.797 ± .003**	.773 ± .011•	.766 ± .005•	.770 ± .008•	.726 ± .006•	.762 ± .001•	.752 ± .010•
mediamill	.722 ± .002•	**.738 ± .002**	.737 ± .002	.662 ± .011•	.658 ± .007•	.672 ± .006•	.631 ± .008•	.643 ± .012•

To make a fair comparison, for PML methods, the multi-label classifier is trained based on both the training and validation data. We also use a linear classifier as the base model for PML-MD. We use SGD as the optimizer for 500 epochs and the learning rate is set as 0.01. The ℓ_2-regularization term is added with the parameter of 0.0001. For the comparing methods, parameters are determined by the performance on validation set if no default value given in their literature.

Table 3: Comparison results between the proposed method and comparing methods under high-level label noise, where •/○ indicates whether the proposed method is significantly superior/inferior to the comparing methods via paired t-test (at 0.05 significance level).

Data	PML-MD	PML-MD+	Baseline	PMLNI	PMLLRS	PARMAP	PARVLS	fPML
Hamming loss (the smaller, the better)								
medical	.027 ± .000•	**.020 ± .000**	.024 ± .002•	.034 ± .013	.041 ± .002•	.073 ± .002•	.340 ± .011•	.027 ± .000•
enron	**.068 ± .004**	.070 ± .007	.076 ± .005	.096 ± .004•	.187 ± .008•	.149 ± .013•	.435 ± .024•	.111 ± .011•
image	.214 ± .008	**.213 ± .003**	.224 ± .007	.280 ± .030•	.329 ± .015•	.346 ± .027•	.529 ± .068•	.247 ± .002•
scene	.160 ± .004	**.156 ± .004**	.161 ± .004•	.337 ± .070•	.206 ± .036•	.300 ± .062•	.347 ± .056•	.178 ± .002•
yeast	**.239 ± .004**	.271 ± .013•	.285 ± .011•	.434 ± .074•	.299 ± .029•	.273 ± .017•	.381 ± .020•	.302 ± .003•
slashdot	.021 ± .002	**.017 ± .001**	.025 ± .002•	.053 ± .002•	.052 ± .008•	.027 ± .004•	.369 ± .035•	.055 ± .000•
tmc	.073 ± .000	**.072 ± .001**	.083 ± .004•	.091 ± .003•	.131 ± .009•	.095 ± .012•	.401 ± .039•	.101 ± .000•
mediamill	.083 ± .004•	**.077 ± .002**	.080 ± .003	.831 ± .089•	.099 ± .005•	.078 ± .007	.450 ± .050•	.088 ± .000•
Ranking loss (the smaller, the better)								
medical	**.070 ± .021**	.076 ± .010	.125 ± .028•	.226 ± .043•	.231 ± .028•	.172 ± .027•	.234 ± .025•	.245 ± .032•
enron	**.183 ± .004**	.185 ± .012	.277 ± .020•	.387 ± .014•	.384 ± .018•	.287 ± .036•	.420 ± .026•	.388 ± .022•
image	.236 ± .030	**.233 ± .011**	.245 ± .019	.309 ± .046•	.408 ± .062•	.383 ± .016•	.426 ± .038•	.409 ± .058•
scene	**.110 ± .007**	.126 ± .009•	.230 ± .027•	.273 ± .059•	.278 ± .061•	.347 ± .103•	.267 ± .053•	.304 ± .040•
yeast	**.176 ± .002**	.178 ± .005	.201 ± .009•	.297 ± .051•	.283 ± .033•	.249 ± .022•	.257 ± .010•	.285 ± .045•
slashdot	**.039 ± .003**	.046 ± .005	.070 ± .014•	.121 ± .023•	.121 ± .015•	.070 ± .010•	.108 ± .006•	.092 ± .008•
tmc	**.063 ± .001**	.064 ± .003	.175 ± .041•	.175 ± .034•	.217 ± .039•	.140 ± .018•	.212 ± .007•	.217 ± .025•
mediamill	.080 ± .002•	.071 ± .000	**.069 ± .001**	.233 ± .022•	.248 ± .046•	.175 ± .023•	.240 ± .028•	.233 ± .013•
One error (the smaller, the better)								
medical	**.289 ± .045**	.342 ± .024•	.516 ± .087•	.811 ± .031•	.801 ± .041•	.665 ± .034•	.773 ± .077•	.686 ± .075•
enron	.720 ± .040	.733 ± .016	.831 ± .027•	.835 ± .025•	.829 ± .012•	**.705 ± .042**	.797 ± .022•	.749 ± .029
image	.420 ± .043	**.416 ± .026**	.417 ± .024	.522 ± .052•	.704 ± .028•	.616 ± .023•	.517 ± .043•	.616 ± .055•
scene	**.320 ± .012**	.340 ± .008•	.517 ± .044•	.597 ± .072•	.585 ± .107•	.625 ± .101•	.417 ± .070•	.621 ± .091•
yeast	.232 ± .003	**.229 ± .017**	.250 ± .006	.360 ± .110	.371 ± .148	.334 ± .072	.271 ± .048	.343 ± .120
slashdot	.327 ± .003	.333 ± .006	.349 ± .021	.526 ± .073•	.477 ± .087•	.145 ± .026○	.141 ± .025○	**.096 ± .012**
tmc	**.257 ± .005**	.264 ± .018	.372 ± .081•	.437 ± .078•	.573 ± .063•	.356 ± .061•	.404 ± .037•	.400 ± .078•
mediamill	.187 ± .011	**.170 ± .005**	.175 ± .008	.286 ± .068•	.209 ± .033	.220 ± .057	.197 ± .009•	.283 ± .065•
Coverage (the smaller, the better)								
medical	**.092 ± .024**	.100 ± .013	.153 ± .029•	.255 ± .044•	.260 ± .025•	.200 ± .030•	.253 ± .028•	.273 ± .034•
enron	.336 ± .012	**.335 ± .025**	.432 ± .024•	.588 ± .029•	.601 ± .023•	.497 ± .043•	.635 ± .021•	.603 ± .023•
image	.243 ± .027	**.237 ± .011**	.256 ± .022	.300 ± .036•	.371 ± .052•	.354 ± .011•	.336 ± .021•	.382 ± .047•
scene	**.107 ± .007**	.123 ± .007•	.207 ± .023•	.241 ± .048•	.246 ± .053•	.309 ± .086•	.197 ± .054•	.268 ± .033•
yeast	**.457 ± .005**	.467 ± .005•	.507 ± .021•	.624 ± .047•	.593 ± .028•	.561 ± .039•	.548 ± .018•	.586 ± .046•
slashdot	**.050 ± .006**	.060 ± .008	.091 ± .021•	.159 ± .025•	.159 ± .015•	.103 ± .011•	.146 ± .008•	.138 ± .013•
tmc	**.151 ± .002**	.153 ± .005	.313 ± .044•	.311 ± .040•	.349 ± .045•	.270 ± .021•	.345 ± .009•	.356 ± .024•
mediamill	.263 ± .004•	.240 ± .002•	**.235 ± .003**	.528 ± .019•	.562 ± .054•	.458 ± .037•	.564 ± .040•	.541 ± .029•
Average precision (the greater, the better)								
medical	**.759 ± .037**	.711 ± .014•	.543 ± .071•	.307 ± .041•	.295 ± .048•	.426 ± .032•	.324 ± .057•	.365 ± .052•
enron	**.534 ± .011**	.526 ± .016	.429 ± .020•	.218 ± .013•	.227 ± .011•	.308 ± .038•	.222 ± .018•	.251 ± .017•
image	.722 ± .030	**.727 ± .016**	.718 ± .016	.655 ± .037•	.544 ± .034•	.592 ± .011•	.605 ± .020•	.570 ± .047•
scene	**.807 ± .007**	.790 ± .009•	.661 ± .027•	.612 ± .052•	.615 ± .061•	.567 ± .091•	.675 ± .070•	.581 ± .057•
yeast	.747 ± .003	**.747 ± .008**	.723 ± .006•	.618 ± .061•	.620 ± .052•	.651 ± .028•	.688 ± .013•	.610 ± .049•
slashdot	**.910 ± .006**	.906 ± .004	.877 ± .018•	.616 ± .056•	.640 ± .058•	.835 ± .018•	.798 ± .019•	.846 ± .002•
tmc	**.770 ± .002**	.770 ± .009	.600 ± .067•	.565 ± .054•	.482 ± .053•	.645 ± .050•	.536 ± .028•	.543 ± .047•
mediamill	.722 ± .003•	**.738 ± .001**	.738 ± .003	.501 ± .045•	.514 ± .042•	.583 ± .037•	.516 ± .025•	.483 ± .027•

Regarding the last 8 multi-label datasets, to construct candidate label sets for training instances, the irrelevant labels can be flipped with a probability. To further validate the effectiveness of the proposed method under different levels of label noise, we define two levels of label noise, i.e., low-level label noise and high-level label noise:

Table 4: Comparison results between the proposed method and comparing methods on real-world PML datasets, where •/○ indicates whether the proposed method is significantly superior/inferior to the comparing methods via paired t-test (at 0.05 significance level).

Data	PML-MD	PML-MD+	Baseline	PMLNI	PMLLRS	PARMAP	PARVLS	fPML
Hamming loss (the smaller, the better)								
music_emotion	**.192 ± .003**	.192 ± .004	.201 ± .008	.227 ± .002•	.230 ± .003•	.256 ± .003•	.243 ± .004•	.220 ± .001•
mirflickr	.128 ± .002	**.123 ± .004**	.142 ± .002•	.166 ± .002•	.185 ± .007•	.152 ± .002•	.224 ± .005•	.252 ± .001•
Ranking loss (the smaller, the better)								
music_emotion	**.223 ± .005**	.228 ± .010	.235 ± .003•	.251 ± .005•	.257 ± .007•	.324 ± .004•	.358 ± .005•	.293 ± .017•
mirflickr	**.059 ± .002**	.060 ± .004	.066 ± .001•	.122 ± .002•	.104 ± .004•	.103 ± .004•	.219 ± .012•	.148 ± .023•
One error (the smaller, the better)								
music_emotion	**.393 ± .012**	.399 ± .019	.427 ± .007•	.489 ± .006•	.527 ± .009•	.584 ± .014•	.584 ± .016•	.515 ± .013•
mirflickr	**.113 ± .006**	.123 ± .008	.141 ± .004•	.297 ± .008•	.218 ± .010•	.147 ± .006•	.403 ± .137•	.239 ± .047•
Coverage (the smaller, the better)								
music_emotion	**.393 ± .006**	.398 ± .009	.400 ± .004	.414 ± .006•	.416 ± .009•	.486 ± .005•	.493 ± .007•	.453 ± .017•
mirflickr	**.173 ± .003**	.174 ± .002	.178 ± .001	.228 ± .002•	.214 ± .003•	.215 ± .004•	.302 ± .004•	.251 ± .019•
Average precision (the greater, the better)								
music_emotion	**.652 ± .008**	.648 ± .012	.627 ± .005•	.599 ± .003•	.587 ± .007•	.531 ± .007•	.523 ± .005•	.579 ± .006•
mirflickr	**.897 ± .004**	.894 ± .005	.887 ± .003•	.790 ± .003•	.827 ± .008•	.849 ± .004•	.678 ± .005•	.795 ± .026•

Table 5: Friedman statistics F_F in terms of each evaluation metric and the critical value at 0.05 significance level (# comparing algorithms = 8, # datasets = 18).

Evaluation metric	F_F	critical value
Hamming Loss	8.2930	
Ranking loss	28.9205	
One Error	4.1730	2.1310
Coverage	24.6596	
Average Precision	23.3200	

- Low-level label noise: each class label can be flipped from an irrelevant label to a candidate one with a probability randomly sampled from {0.2,0.3,0.4,0.5}.
- High-level label noise: each class label can be flipped from an irrelevant label to a candidate one with a probability randomly sampled from {0.5,0.6,0.7,0.8}.

For each level of label noise, we repeat 5 experiments and report the averaging results.

4.2 Comparison Result

Table 2 and Table 3 report comparison results between the proposed PML-MD and the comparing methods on synthetic datasets under low-level and high-level label noise, respectively. For each dataset, the paired t-test based on five repeats is conducted to show whether the performance of PML-MD is significantly different from the comparing methods. From the results, it can be observed that: 1) under low-level label noise, PML-MD achieves the best performance on almost all cases in terms of *ranking loss*, *coverage* and *average precision*; 2) under low-level label noise, PML-MD shows comparable performance with PML-NI and PARTICLE in terms of *hamming loss* and *one error*; 3) under high-level label noise, PML-MD significantly outperforms the comparing methods on almost all cases in terms of five evaluation metrics; 4) the performance of PML-MD is generally better than PML-MD+, especially under high-level label noise. One possible reason is that for PML-MD+, the model may be over-fitted to the validation set, which has been directly used to train the multi-label classifier. This leads to the inaccurate estimation of label confidences for PML-MD+. From the results, it seems that the baseline method outperforms PML methods in some cases. One possible explanation is that compared to PML methods, the baseline method can exploit validation examples more effectively.

We also perform experiments on real-world PML datasets to validate the practical usefulness of the proposed method. Table 4 reports comparison results between the proposed PML-MD and comparing methods. Similarly, the paired t-test based on five repeats is conducted to show whether the proposed method is significantly different from the comparing methods. From the results, it can be observed that the performance of PML-MD is significantly superior to the comparing methods in all cases. In general, the performance of PML-MD is better than PML-MD+. These results further validate the effectiveness of the proposed PML-MD in real-world applications.

Furthermore, *Friedman test* [7] is utilized as a statistical test to demonstrate the relative performance among the comparing methods. Table 5 reports the Friedman statistic F_F and the corresponding critical value with respective to each evaluation metric (# comparing methods = 8, # datasets = 18). For each evaluation metric, the null hypothesis of indistinguishable performance among the comparing algorithms is rejected at 0.05 significance level. From the table, it can be observed that the Friedman statistic F_F is significantly larger than the critical value in terms of five evaluation metrics, which indicates the performances of the comparing methods are significantly different.

Finally, we perform the post-hoc *Bonferroni-Dunn test* to illustrate the relative performance among the comparing methods.

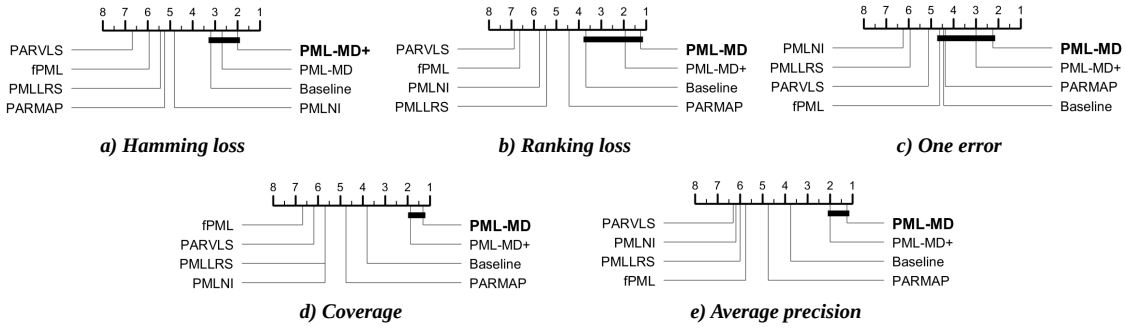
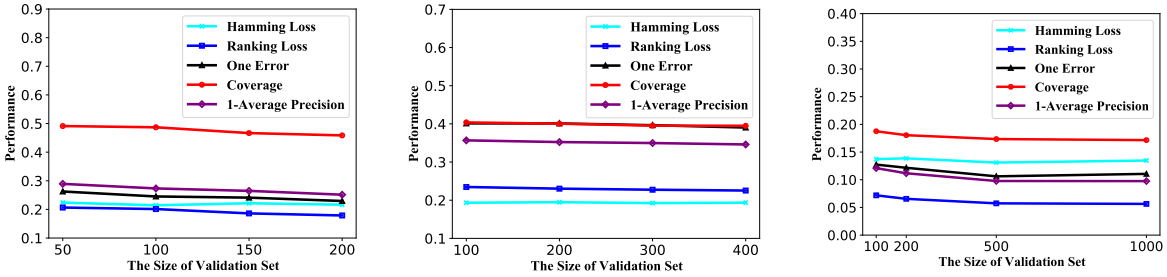


Figure 2: Comparison of PML-MD (control algorithm) against five comparing algorithms with the Bonferroni-Dunn test. Algorithms not connected with PML-MD in the CD diagram are considered to have a significantly different performance from the control algorithm (CD = 2.4905 at 0.05 significance level).



(a) yeast (a) music_emotion (c) mirflickr

Figure 3: Performance of PML-MD with varying size of validation set.

Specifically, we treat PML-MD as the control method whose averaging rank difference against the comparing methods is calibrated with the *critical difference* (CD). Accordingly, the performance of PML-MD is deemed to be significantly different from the comparing methods in the case that their averaging ranks differ by at least one CD (CD = 2.4905 in our experiments: # comparing algorithms = 8, # datasets = 18). As shown in Figure 2, PML-MD methods achieve the best (lowest) averaging rank in terms of all evaluation metrics, where the averaging rank of each comparing algorithm is marked along the axis (lower ranks to the right). The effectiveness of the proposed PML-MD method is convincingly validated by these experimental results.

4.3 Study On the Size of Validation Data

In this section, we study the influence of the size of validation set on the performance of PML-MD. Figure 3 illustrates the performance curves of PML-MD as the size of validation set changes among {50,100,150,200} on yeast, {100,200,300,400} on music_emotion and {100,200,500,1000} on mirflickr, in terms of five evaluation metrics. From the figures, it can be observed that the peroformance of PML-MD is also acceptable when the size of validation set is relative small, such as 50 for yeast as well as 100 for music_emotion and mirflickr. The observation discloses that the meta loss on the validation set can be regarded as a regularization term, which encourages the model to achieve precise confidence estimation. Therefore, in practice, in order to obtain the promising performance, one only needs to precisely annotate a few of instances in advance.

5 CONCLUSION

In this paper, we propose a new PML method by learning to disambiguate for candidate labels. Different from existing methods that perform disambiguation based on extra assumptions of data, we adaptively estimate the confidence for each candidate label guided by the meta-objective on a small additional validation set. The multi-label classifier is trained by minimizing the ranking loss weighted by confidences. By using the online approximation strategy, these two objectives are optimized in a pipeline iteratively to guarantee the time efficiency. Extensive experimental results validate the effectiveness of the proposed method. In the future, we will design more powerful base models to further improve the performance of PML-MD framework.

6 ACKNOWLEDGMENTS

This research was supported by the National Key R&D Program of China (2020AAA0107000), NSFC (62076128, 61732006) and the China University S&T Innovation Plan Guided by the Ministry of Education.

REFERENCES

[1] Marcin Andrychowicz, Misha Denil, Sergio Gómez Colmenarejo, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 3988–3996.

[2] Shai Ben-David, David Loker, Nathan Srebro, and Karthik Sridharan. 2012. Minimizing the misclassification error rate using a surrogate convex loss. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*. 83–90.

[3] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. 2004. Learning multi-label scene classification. *Pattern recognition* 37, 9 (2004), 1757–1771.

[4] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. 2019. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5177–5186.

[5] Weiwei Cheng and Eyke Hüllermeier. 2009. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning* 76, 2-3 (2009), 211–225.

[6] Timothee Cour, Ben Sapp, and Ben Taskar. 2011. Learning from partial labels. *The Journal of Machine Learning Research* 12 (2011), 1501–1536.

[7] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.

[8] André Elisseeff, Jason Weston, et al. 2001. A kernel method for multi-labelled classification.. In *NIPS*, Vol. 14. 681–687.

[9] Lei Feng and Bo An. 2018. Leveraging latent label distributions for partial label learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2107–2113.

[10] Lei Feng and Bo An. 2019. Partial label learning with self-guided retraining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3542–3549.

[11] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. 2020. Provably Consistent Partial-Label Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 1126–1135.

[13] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. 2018. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*. PMLR, 1568–1577.

[14] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. 2008. Multilabel classification via calibrated label ranking. *Machine learning* 73, 2 (2008), 133–153.

[15] Shantanu Godbole and Sunita Sarawagi. 2004. Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 22–30.

[16] Shuiwang Ji, Lei Tang, Shipeng Yu, and Jieping Ye. 2008. Extracting shared subspace for multi-label classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 381–389.

[17] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*. PMLR, 1885–1894.

[18] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences* 40 (2017).

[19] Liping Liu and Thomas Dietterich. 2014. Learnability of the superset label learning problem. In *International Conference on Machine Learning*. PMLR, 1629–1637.

[20] Gengyu Lyu, Songhe Feng, and Yidong Li. 2020. Partial multi-label learning via probabilistic graph matching mechanism. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 105–113.

[21] Andrew Kachites McCallum. 1999. Multi-label text classification with a mixture model trained by EM. In *AAAI 99 workshop on text learning*. Citeseer.

[22] Stefano Pini, Marcella Cornia, Federico Bolelli, Lorenzo Baraldi, and Rita Cucchiara. 2019. M-VAD names: a dataset for video captioning with naming. *Multimedia Tools and Applications* 78, 10 (2019), 14007–14027.

[23] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. (2016).

[24] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. 2018. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676* (2018).

[25] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*. PMLR, 4334–4343.

[26] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379* (2019).

[27] Lijuan Sun, Songhe Feng, Tao Wang, Congyan Lang, and Yi Jin. 2019. Partial multi-label learning by low-rank and sparse decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5016–5023.

[28] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2019. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 403–412.

[29] Sebastian Thrun and Lorien Pratt. 2012. *Learning to learn*. Springer Science & Business Media.

[30] Haobo Wang, Weiwei Liu, Yang Zhao, Chen Zhang, Tianlei Hu, and Gang Chen. 2019. Discriminative and Correlative Partial Multi-Label Learning.. In *IJCAI*. 3691–3697.

[31] Xuan Wu and Min-Ling Zhang. 2018. Towards Enabling Binary Decomposition for Partial Label Learning.. In *IJCAI*. 2868–2874.

[32] Ming-Kun Xie and Sheng-Jun Huang. 2018. Partial Multi-Label Learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[33] Ming-Kun Xie and Sheng-Jun Huang. 2020. Semi-Supervised Partial Multi-Label Learning. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 691–700.

[34] Ming-Kun Xie and Sheng-Jun Huang. 2021. CCMN: A General Framework for Learning with Class-Conditional Multi-Label Noise. *arXiv preprint arXiv:2105.07338* (2021).

[35] Ming-Kun Xie and Sheng-Jun Huang. 2021. Partial multi-label learning with noisy label identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[36] Ning Xu, Yun-Peng Liu, and Xin Geng. 2020. Partial multi-label learning with label distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6510–6517.

[37] Yan Yan and Yuhong Guo. 2019. Adversarial partial multi-label learning. *arXiv preprint arXiv:1909.06717* (2019).

[38] Yan Yan and Yuhong Guo. 2020. Partial label learning with batch label correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6575–6582.

[39] Fei Yu and Min-Ling Zhang. 2016. Maximum margin partial label learning. In *Asian conference on machine learning*. PMLR, 96–111.

[40] Guoxian Yu, Xia Chen, Carlotta Domeniconi, Jun Wang, Zhao Li, Zili Zhang, and Xindong Wu. 2018. Feature-induced partial multi-label learning. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1398–1403.

[41] Tingting Yu, Guoxian Yu, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. 2020. Partial multi-label learning using label compression. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 761–770.

[42] Min-Ling Zhang and Jun-Peng Fang. 2020. Partial multi-label learning via credible label elicitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[43] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. 2017. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering* 29, 10 (2017), 2155–2167.

[44] Min-Ling Zhang and Zhi-Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition* 40, 7 (2007), 2038–2048.

[45] Min-Ling Zhang and Zhi-Hua Zhou. 2013. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* 26, 8 (2013), 1819–1837.