SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2025, Vol. 68, Iss. 3, 132104:1–132104:15
https://doi.org/10.1007/s11432-023-4230-2

Partial multi-label learning via label-specific feature corrections

Jun-Yi HANG[1,2] & Min-Ling ZHANG[1,2*]

[1]*School of Computer Science and Engineering, Southeast University, Nanjing 210096, China*
[2]*Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 210096, China*

Received 17 August 2023/Revised 25 November 2024/Accepted 26 November 2024/Published online 24 January 2025

Abstract Partial multi-label learning (PML) allows learning from rich-semantic objects with inaccurate annotations, where a set of candidate labels are assigned to each training example but only some of them are valid. Existing approaches rely on disambiguation to tackle the PML problem, which aims to correct noisy candidate labels by recovering the ground-truth labeling information ahead of prediction model induction. However, this dominant strategy might be suboptimal as it usually needs extra assumptions that cannot be fully satisfied in real-world scenarios. Instead of label correction, we investigate another strategy to tackle the PML problem, where the potential ambiguity in PML data is eliminated by correcting instance features in a label-specific manner. Accordingly, a simple yet effective approach named PASE, i.e., partial multi-label learning via label-specific feature corrections, is proposed. Under a meta-learning framework, PASE learns to exert label-specific feature corrections so that potential ambiguity specific to each class label can be eliminated and the desired prediction model can be induced on these corrected instance features with the provided candidate labels. Comprehensive experiments on a wide range of synthetic and real-world data sets validate the effectiveness of the proposed approach.

Keywords machine learning, multi-label learning, partial multi-label learning, label-specific features, feature correction

Citation Hang J-Y, Zhang M-L. Partial multi-label learning via label-specific feature corrections. Sci China Inf Sci, 2025, 68(3): 132104, https://doi.org/10.1007/s11432-023-4230-2

1 Introduction

Partial multi-label learning (PML) deals with the problem where each training example is associated with a set of candidate labels, among which only some labels are valid [1]. The need to learn from such inaccurate supervision arises with the popularity of crowdsourcing platforms, where sets of candidate labels provided by multiple annotators inevitably contain some irrelevant labels due to potentially unreliable annotators (Figure 1).

Disambiguation is a common strategy to tackle the PML problem, which aims to correct noisy candidate labels by recovering ground-truth labeling information. As a challenging preprocessing procedure before inducing the desired prediction model, disambiguation is generally realized by introducing extra assumptions on data structures. For instance, the smoothness assumption is widely employed to elicit the labeling confidence for each candidate label via label propagation [2,3], label enhancement [4,5], or cluster assignment [1,6], etc. Meanwhile, some other approaches hypothesize that noisy labels in candidate labels may be sparse [7,8], so that ground-truth labels can be identified by imposing sparsity constraints on underlying noisy labels. However, these extra assumptions cannot always be satisfied in real-world scenarios, which may lead to failure in disambiguation and further result in error accumulation in the induced prediction model. Under such a background, an interesting question naturally arises: Can the predominance of disambiguation be challenged for the PML problem?

Intuitively, a PML example can be dealt with properly in two different strategies. One strategy is to correct an example's labeling information to align with its features, so that potential ambiguity in this PML example can be eliminated to behave like a clean multi-label example. Actually, this is just the methodology behind disambiguation. In contrast, the other strategy is to correct the example's features, so that the semantic information embodied in the corrected features can be altered to be consistent

* Corresponding author (email: zhangml@seu.edu.cn)

Figure 1 (Color online) Example of partial multi-label learning. Among the 7 candidate labels provided by crowdsourcing annotators, irrelevant labels for the image exist, including cow, flower, and people.

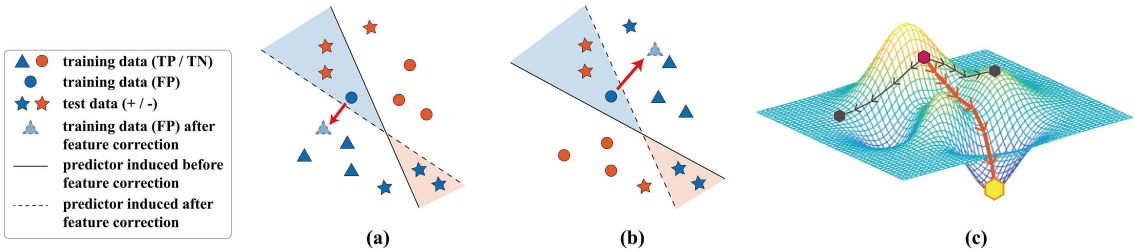

Figure 2 (Color online) Intuitive illustration of our approach. Considering the labeling information of each class label, PML training examples can be grouped into three categories, i.e., examples with true positive (TP) labels, true negative (TN) labels, and false positive (FP) labels. Such inaccurately annotated training examples often lead to unsatisfactory predictor (solid line), which generalizes poorly on clean test data. (a) Data distribution of label l_1. By correcting the example's features to align the with provided candidate labels, potential ambiguity in training examples can be eliminated, so that the desired predictor (dotted line) can be induced. (b) Data distribution of label l_2. As shown in (a) and (b), the optimal feature corrections (red arrows) for example should be considered in a label-specific manner to account for each class label's own characteristics (caused by differences in data distributions and correctness of annotations, etc.). (c) The loss landscape on an additional validation set in the predictor's parameter space, where different correction procedures can lead to totally different predictors (hexagons). Under the meta-learning framework, the optimal feature corrections are learned to ensure the induced predictor (the yellow hexagon) can achieve minimal empirical risk on this validation set.

with provided candidate labels. However, such a strategy has been rarely investigated in current PML literature.

Following this strategy, we challenge the predominance of disambiguation with a simple yet effective label-specific feature correction procedure. Figure 2 provides an illustrative example of the basic idea behind our work. Instead of directly correcting noisy labels by disambiguation as existing approaches do, we attempt to eliminate potential ambiguity in PML examples by correcting the instance into the right position in the feature space. Since each class label may possess its own characteristics, such a right position to correct an instance might be different among class labels (as shown in Figure 2). Therefore, it is reasonable to consider label-specific feature corrections to thoroughly eliminate potential ambiguity specific to each class label so that the desired prediction model can be induced.

Here, the major challenge lies in that such a right position can be hard to determine when the learning algorithm is only accessible to inaccurately annotated examples. To tackle this challenge, the meta-learning framework [9] provides a natural and principled choice: by formalizing the learning process as a bilevel optimization problem, the right position or the optimal correction can be learned to decrease classification error on an additional clean validation set. Specifically, a bilevel optimization problem is constructed in PASE, which is solved via an alternating procedure to learn to exert label-specific feature corrections and induce the prediction model iteratively. In each iteration, the feature correction procedure is optimized to eliminate potential ambiguity in training examples guided by empirical risk on the validation set, while the prediction model is updated on these corrected training instances with provided candidate labels. Comprehensive experiments on diversified benchmark data sets show that PASE performs better than well-established PML approaches.

The contributions of this paper can be summarized as follows.

- We investigate a new strategy for tackling the PML problem, i.e., disambiguation by feature cor-

rection, which is an indispensable complement to the existing disambiguation strategy based on label correction.

• Considering the distinct characteristics of each class label, we introduce the idea of label-specific features (a supervised representation learning technique) to correct instances in a label-specific manner, which extends the applicable scenarios of label-specific features from supervised learning to weakly-supervised learning.

• We provide a comprehensive evaluation on a total of 21 benchmark data sets, clearly demonstrating the superiority of PASE and the effectiveness of our feature correction strategy.

The rest content is organized as follows. Section 2 briefly reviews related work. Section 3 presents details of the proposed PASE approach. Section 4 reports experimental results over a wide range of synthetic and real-world data sets. Section 5 concludes this paper.

2 Related work

As an emerging weakly-supervised learning problem [10, 11], partial multi-label learning has been extensively studied in recent years. Here, a brief review of two closely related learning problems, i.e., partial label learning [12] and multi-label learning [13, 14], is made first.

2.1 Partial label learning

Partial label learning (PLL) deals with the problem where a set of candidate labels are assigned to each training example but only one of them is valid. To learn from such inaccurate supervision, disambiguating between ground-truth labels and noisy ones is a prevalent strategy adopted by the majority of existing approaches. Generally speaking, these approaches can be roughly grouped into two categories, i.e., identification-based disambiguation and averaging-based disambiguation. In identification-based disambiguation, recovery of the gound-truth label is formalized as a latent variable inference task [15–19], where an expectation-maximization procedure can be constructed to identify the ground-truth label and induce prediction model alternatively. While in averaging-based disambiguation, all candidate labels are treated equally and prediction is made by averaging model outputs on each candidate label [12, 20, 21].

Recent years have witnessed the emergence of several disambiguation-free approaches. For example, some problem transformation approaches tackle the PLL problem by directly transforming the original weakly-supervised problem into well-established supervised counterparts [22, 23] with error-correcting outputs codes (ECOC) or binary decomposition techniques. While some other approaches [24, 25] turn to design provably consistent risk functions to derive the desired prediction model from PLL data. Both PLL and PML are expected to learn from inaccurately annotated training examples. Nevertheless, PML is a more complicated problem than PLL since the number of ground-truth labels is concealed from the learning algorithm.

2.2 Multi-label learning

With the same goal to induce a multi-label predictor as PML, multi-label learning (MLL) learns from rich-semantic objects with accurate annotations. To tackle the MLL problem, a feasible strategy is to model label correlations during the learning process. Generally speaking, these approaches can be roughly grouped into three categories, namely first-order approaches [26,27], second-order approaches [28,29], and high-order approaches [30, 31].

Complementary to label correlation exploitation, label-specific features have been proven to be another effective strategy to improve MLL. The basic idea is to learn from multi-label data with tailored features accounting for the distinct discriminative characteristics of each class label. Such tailored features can be constructed by performing prototype-based label-specific feature transformation, which captures the underlying data properties of each class label with clustering analysis [32–35]. Alternatively, label-specific features can also be generated by retaining a feature subset as the most pertinent features for each class label [36–39]. In this paper, we extend label-specific features to a weakly-supervised learning scenario, which is notable yet underexplored in related literature. As demonstrated in the later section, the intuitively simple label-specific feature correction procedure is rather effective to tackle the PML problem.

2.3 Partial multi-label learning

In PML, existing studies mainly rely on disambiguation to induce prediction models. Conceptually speaking, disambiguation can be regarded as a preprocessing procedure before inducing the desired multi-label predictor. For instance, a two-stage pipeline is investigated in [2–5], where the ground-truth labeling information is recovered by either propagating candidate labels in an instance-similarity graph, or performing label enhancement with label correlation and smoothness constraints. Such a two-stage pipeline can be adapted to a multi-round one, where the labeling confidence of each candidate label and the prediction model are optimized iteratively [1,6]. While some other studies try to perform disambiguation and induce prediction model simultaneously [7, 8, 40–43], which can be regarded as a co-regularization procedure to improve each other. It is worth noting that these approaches learn from PML data by manipulating the label space.

Nevertheless, it is still an underexplored direction to tackle the PML problem via manipulating the feature space. Some studies jointly consider the manipulations in the feature space and the label space [44–48], but they still rely on disambiguation to deal with inaccurate annotations. The only available feature manipulation approach [49] regards the task of PML as a feature completion problem, where the missing features are completed with smoothness and low-rank assumptions. However, these extra assumptions on the data structure hardly hold in real-world scenarios [42], with no guarantee to complete the missing features properly. We eliminate the dependence on these extra assumptions by formalizing the learning process under a meta-learning framework [9, 50, 51] and further demonstrate that it is essential to consider distinct characteristics of each class label via generalizing the idea of label-specific features from MLL. We will detail our approach in Section 3.

3 Pase approach

3.1 Preliminaries

Formally, let $\mathcal{X} = \mathbb{R}^d$ denote the feature space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ denote the label space with q class labels. A partial multi-label example is defined as $(\boldsymbol{x}, S)$, where $\boldsymbol{x} \in \mathcal{X}$ is its feature vector and $S \subseteq \mathcal{Y}$ is its set of candidate labels. As a basic assumption of PML, the ground-truth labels reside in the candidate label set, i.e., $Y \subseteq S$, and are concealed from the learning algorithm. Given a partial multi-label data set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) | 1 \leqslant i \leqslant m\}$, the goal of PML is to derive a multi-label predictor $h : \mathcal{X} \to 2^{\mathcal{Y}}$ which can accurately predict all the relevant labels for an unseen instance.

For notation brevity, a q-dimensional indicator vector $\boldsymbol{s} \in \{0,1\}^q$ denotes the set of candidate labels S, where $s_k = 1$ indicates $l_k \in S$ and $s_k = 0$ otherwise. Similarly, a q-dimensional indicator vector $\boldsymbol{y} \in \{0,1\}^q$ denotes the set of ground-truth labels Y.

3.2 Learning framework

We firstly provide a general learning framework for our novel strategy to deal with the PML problem, i.e., label-specific feature corrections. Under this framework, instance features are corrected in a label-specific manner to eliminate potential ambiguity specific to each class label. With corrected instance features, the desired multi-label predictor can be directly induced by empirical risk minimization on given training set $\mathcal{D}$ as follows:

$$\min_{\boldsymbol{\Theta}, \boldsymbol{\phi}} \mathcal{L}^{\text{train}}(\mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\phi}; \boldsymbol{\Psi}) = \min_{\boldsymbol{\Theta}, \boldsymbol{\phi}} \frac{1}{m} \sum_{(\boldsymbol{x}, \boldsymbol{s}) \sim \mathcal{D}} \sum_{k=1}^{q} \ell(f_k(g_k(e_{\boldsymbol{\phi}}(\boldsymbol{x}); \boldsymbol{\psi}_k); \boldsymbol{\theta}_k), s_k), \tag{1}$$

where $f_k(\cdot; \boldsymbol{\theta}_k)$ is the prediction model for label l_k parameterized by $\boldsymbol{\theta}_k$, and $g_k(\cdot; \boldsymbol{\psi}_k)$ denotes the feature correction function specific to label l_k which is parameterized by $\boldsymbol{\psi}_k$. For brevity, their parameter sets $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_q\}$ and $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_q\}$ are abbreviated as $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, respectively. While $e_{\boldsymbol{\phi}} : \mathbb{R}^d \to \mathbb{R}^{d_z}$ is an embedding function parameterized by $\boldsymbol{\phi}$, which is shared among all the class labels. Such an embedding function makes the above learning framework general enough to accommodate some high-dimensional raw data[1)] and also provides an implicit way to capture some common knowledge among class labels.

1) For example, the embedding function can be implemented as a CNN backbone to deal with raw image data.

As an essential building block, the label-specific feature correction function can be implemented in diversified ways. In this paper, we opt to instantiate the correction function with neural networks for the sake of simplicity,

$$g_k(e_{\boldsymbol{\phi}}(\boldsymbol{x});\boldsymbol{\psi}_k) = w_k(e_{\boldsymbol{\phi}}(\boldsymbol{x});\boldsymbol{\psi}_k)\cdot e_{\boldsymbol{\phi}}(\boldsymbol{x}) + b_k(e_{\boldsymbol{\phi}}(\boldsymbol{x});\boldsymbol{\psi}_k), \tag{2}$$

where $w_k(\cdot;\boldsymbol{\psi}_k)$ and $b_k(\cdot;\boldsymbol{\psi}_k)$ are two small hyper-networks to produce scaling and translation factors respectively for correcting an input instance into a right position in the feature space. It is worth noting that such an instantiation only represents an intuitive implementation and is not meant to be the best possible practice for feature correction. We have studied some variants of this instantiation in Appendix B and leave further explorations as future work.

However, there is no oracle to provide the just proper feature corrections when solving the optimization problem in (1). Jointly optimizing parameters for the prediction model and the feature correction function by minimizing training error will lead to a model deviating from the desired multi-label predictor, as there exists a gap between the distributions of training and test data.

To tackle this challenge, we propose to formalize the learning process as a bi-level optimization problem, which provides a concise and principled way to ensure the quality of feature correction: the optimal feature correction procedure should lead to a prediction model with small error on the clean validation data,

$$\begin{aligned} &\min_{\boldsymbol{\Psi}} \mathcal{L}^{\mathrm{val}}(\mathcal{D}^v, \boldsymbol{\Theta}^*(\boldsymbol{\Psi}), \boldsymbol{\phi}^*(\boldsymbol{\Psi})) \\ &\text{s.t. } \boldsymbol{\Theta}^*(\boldsymbol{\Psi}), \boldsymbol{\phi}^*(\boldsymbol{\Psi}) = \arg\min_{\boldsymbol{\Theta},\boldsymbol{\phi}} \mathcal{L}^{\mathrm{train}}(\mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\phi}; \boldsymbol{\Psi}), \end{aligned} \tag{3}$$

where $\boldsymbol{\Theta}^*(\boldsymbol{\Psi})$ and $\boldsymbol{\phi}^*(\boldsymbol{\Psi})$ are the model parameters induced by empirical risk minimization on training set given correction parameter $\boldsymbol{\Psi}$. $\mathcal{L}^{\mathrm{val}}(\mathcal{D}^v, \boldsymbol{\Theta}^*(\boldsymbol{\Psi}), \boldsymbol{\phi}^*(\boldsymbol{\Psi}))$ denotes the meta-objective on a small validation set $\mathcal{D}^v = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \leqslant i \leqslant n\}$ with accurately annotated examples, which is instantiated as the empirical risk on validation set $\mathcal{D}^v$ in this paper,

$$\mathcal{L}^{\mathrm{val}}(\mathcal{D}^v, \boldsymbol{\Theta}, \boldsymbol{\phi}) = \frac{1}{n} \sum_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}^v} \sum_{k=1}^{q} \ell(f_k(e_{\boldsymbol{\phi}}(\boldsymbol{x});\boldsymbol{\theta}_k), y_k). \tag{4}$$

Here, no feature correction is exerted when computing the empirical risk on the validation set, since the goal of correction is to deal with inherent ambiguity in training data.

3.3 Optimization procedure

To solve the bi-level optimization problem in (3), an efficient online approximation optimization method [50, 52] is adapted to learn to exert label-specific feature corrections and induce the prediction model in an alternating procedure. Specifically, we iteratively optimize one set of parameters while keeping the other set of parameters fixed until the maximal iteration is reached.

Update feature correction parameter $\boldsymbol{\Psi}$. With fixed model parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\phi}$, the feature correction parameter $\boldsymbol{\Psi}$ can be updated with a two-step stochastic gradient descent procedure which minimizes empirical risks on training set and validation set in succession.

At the t-th iteration of training, we first move one step forward to optimize empirical risk on the training set as follows:

$$\tilde{\boldsymbol{\Theta}}^{(t)}(\boldsymbol{\Psi}) = \boldsymbol{\Theta}^{(t-1)} - \eta\nabla_{\boldsymbol{\Theta}}\mathcal{L}^{\mathrm{train}}(\mathcal{B}, \boldsymbol{\Theta}^{(t-1)}, \boldsymbol{\phi}^{(t-1)}; \boldsymbol{\Psi}^{(t-1)}), \tag{5}$$

$$\tilde{\boldsymbol{\phi}}^{(t)}(\boldsymbol{\Psi}) = \boldsymbol{\phi}^{(t-1)} - \eta\nabla_{\boldsymbol{\phi}}\mathcal{L}^{\mathrm{train}}(\mathcal{B}, \boldsymbol{\Theta}^{(t-1)}, \boldsymbol{\phi}^{(t-1)}; \boldsymbol{\Psi}^{(t-1)}), \tag{6}$$

where η is the learning rate for the inner-objective, i.e., empirical risk on training set, and $\mathcal{B}$ denotes a batch of training examples sampled from training set $\mathcal{D}$. This step is a look-ahead step to explore the influence of the feature correction parameter $\boldsymbol{\Psi}$ on model parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\phi}$, where the temporarily-updated model parameter $\tilde{\boldsymbol{\Theta}}^{(t)}$ (respectively $\tilde{\boldsymbol{\phi}}^{(t)}$) is explicitly expressed as a function of the feature correction parameter $\boldsymbol{\Psi}$, i.e., $\tilde{\boldsymbol{\Theta}}^{(t)}(\boldsymbol{\Psi})$ (respectively $\tilde{\boldsymbol{\phi}}^{(t)}(\boldsymbol{\Psi})$).

Successively, one gradient step is taken to optimize the feature correction parameter $\boldsymbol{\Psi}$ via evaluating the empirical risk of the temporarily-updated model parameterized by $\tilde{\boldsymbol{\Theta}}^{(t)}(\boldsymbol{\Psi})$ and $\tilde{\boldsymbol{\phi}}^{(t)}(\boldsymbol{\Psi})$ on validation set,

$$\boldsymbol{\Psi}^{(t)} = \boldsymbol{\Psi}^{(t-1)} - \mu\nabla_{\boldsymbol{\Psi}}\mathcal{L}^{\mathrm{val}}(\mathcal{B}^v, \tilde{\boldsymbol{\Theta}}^{(t)}(\boldsymbol{\Psi}), \tilde{\boldsymbol{\phi}}^{(t)}(\boldsymbol{\Psi})), \tag{7}$$

Algorithm 1 Pseudocode of the optimization procedure for PASE.

Inputs: $\mathcal{D}$: training set; $\mathcal{D}^v$: validation set; η: learning rate for the inner-objective; μ: learning rate for the meta-objective; b: batch size; T: maximal iteration;

Outputs:

Model parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\phi}$;

Process:

Initialize model parameters $\boldsymbol{\Theta}^{(0)}$, $\boldsymbol{\phi}^{(0)}$, and feature correction parameter $\boldsymbol{\Psi}^{(0)}$;
for $t = 1 : T$ **do**
  Sample a batch of training examples $\mathcal{B}$ from training set $\mathcal{D}$;
  Sample a batch of validation examples $\mathcal{B}^v$ from validation set $\mathcal{D}^v$;
  Evaluate empirical risk on training batch with current parameters: $\mathcal{L}_1^{\text{train}} \leftarrow \mathcal{L}^{\text{train}}(\mathcal{B}, \boldsymbol{\Theta}^{(t-1)}, \boldsymbol{\phi}^{(t-1)}; \boldsymbol{\Psi}^{(t-1)})$;
  Create a temporary model via gradient descent:

$$\begin{cases} \tilde{\boldsymbol{\Theta}}^{(t)}(\boldsymbol{\Psi}) \leftarrow \boldsymbol{\Theta}^{(t-1)} - \eta \nabla_{\boldsymbol{\Theta}} \mathcal{L}_1^{\text{train}}, \\ \tilde{\boldsymbol{\phi}}^{(t)}(\boldsymbol{\Psi}) \leftarrow \boldsymbol{\phi}^{(t-1)} - \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}_1^{\text{train}}; \end{cases}$$

  Evaluate empirical risk on validation batch with updated parameters: $\mathcal{L}_2^{\text{val}} \leftarrow \mathcal{L}^{\text{val}}(\mathcal{B}^v, \tilde{\boldsymbol{\Theta}}^{(t)}(\boldsymbol{\Psi}), \tilde{\boldsymbol{\phi}}^{(t)}(\boldsymbol{\Psi}))$;
  Update correction parameter via gradient descent: $\boldsymbol{\Psi}^{(t)} \leftarrow \boldsymbol{\Psi}^{(t-1)} - \mu \nabla_{\boldsymbol{\Psi}} \mathcal{L}_2^{\text{val}}$;
  Evaluate empirical risk on training batch with currently optimal correction parameter: $\mathcal{L}_3^{\text{train}} \leftarrow \mathcal{L}^{\text{train}}(\mathcal{B}, \boldsymbol{\Theta}^{(t-1)}, \boldsymbol{\phi}^{(t-1)}; \boldsymbol{\Psi}^{(t)})$;
  Update model parameters via gradient descent:

$$\begin{cases} \boldsymbol{\Theta}^{(t)} \leftarrow \boldsymbol{\Theta}^{(t-1)} - \eta \nabla_{\boldsymbol{\Theta}} \mathcal{L}_3^{\text{train}}, \\ \boldsymbol{\phi}^{(t)} \leftarrow \boldsymbol{\phi}^{(t-1)} - \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}_3^{\text{train}}; \end{cases}$$

end for

where μ is the learning rate for the meta-objective, i.e., empirical risk on the validation set and $\mathcal{B}^v$ denotes a batch of validation examples sampled from validation set $\mathcal{D}^v$. With this step, $\boldsymbol{\Psi}^{(t)}$ can be regarded as the currently optimal correction parameter, which provides an optimization direction for model parameters towards better generalization performance on a clean validation set.

Update model parameters $\boldsymbol{\Theta}, \boldsymbol{\phi}$. While feature correction parameter $\boldsymbol{\Psi}$ is fixed, the optimization problem (3) can be stated as follows:

$$\min_{\boldsymbol{\Theta}, \boldsymbol{\phi}} \mathcal{L}^{\text{train}}(\mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\phi}; \boldsymbol{\Psi}). \tag{8}$$

With stochastic gradient descent, model parameters are updated with current correction parameter $\boldsymbol{\Psi}^{(t)}$,

$$\boldsymbol{\Theta}^{(t)} = \boldsymbol{\Theta}^{(t-1)} - \eta \nabla_{\boldsymbol{\Theta}} \mathcal{L}^{\text{train}}(\mathcal{B}, \boldsymbol{\Theta}^{(t-1)}, \boldsymbol{\phi}^{(t-1)}; \boldsymbol{\Psi}^{(t)}), \tag{9}$$

$$\boldsymbol{\phi}^{(t)} = \boldsymbol{\phi}^{(t-1)} - \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}^{\text{train}}(\mathcal{B}, \boldsymbol{\Theta}^{(t-1)}, \boldsymbol{\phi}^{(t-1)}; \boldsymbol{\Psi}^{(t)}). \tag{10}$$

Algorithm 1 summarizes the pseudocode of the above optimization procedure. This procedure can be easily implemented by mainstream deep learning frameworks (e.g., PyTorch), as all involved gradients can be efficiently computed by automatic differentiation techniques.

4 Experiments

4.1 Sanity check

To provide a visualized example of how the feature correction procedure helps with the induction of the desired prediction model, we first make a sanity check for PASE on a toy PML data set. In the data set, the instance has two-dimensional features and the label space consists of two labels. To generate this toy data set, we sample 200 examples per Gaussian distribution (Figure 3(a)). As shown in Figure 3(b), the data set is linearly separable for each class label, but the provided noisy labels will lead to corrupted decision boundaries. For example, for the first label, the desired decision boundary should separate the clusters with indices 1, 2, and 3 (positive clusters) from the clusters with indices 4 and 5 (negative clusters). While the actual decision boundary induced from the noisy data set makes a wrong separation for the 4th cluster.

We take out 10% examples as a hold-out validation set and employ PASE to eliminate potential ambiguity specific to each class label by learning label-specific feature corrections. Figure 3(c) shows the

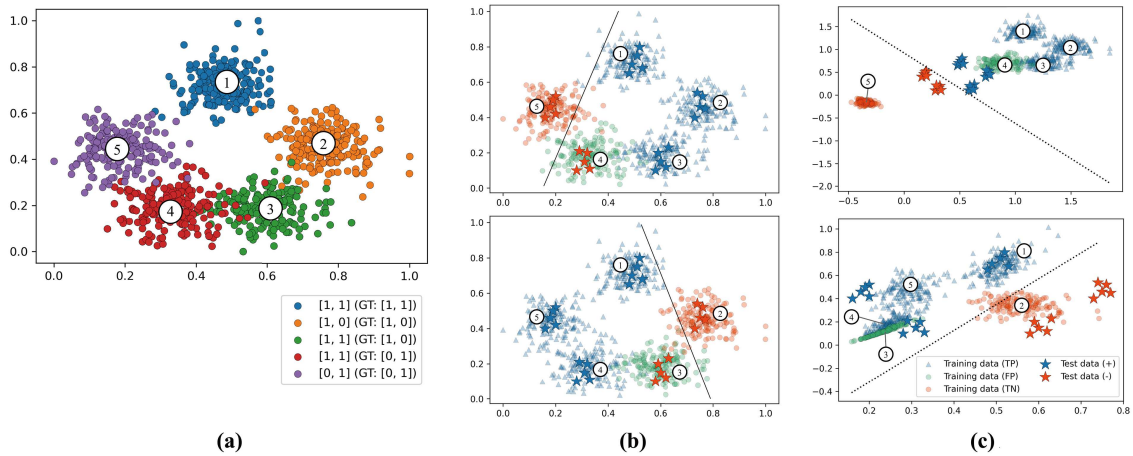

Figure 3 (Color online) Visualization for the working mechanism of PASE. (a) A toy PML data set is generated from 5 Gaussian distributions, where the provided noisy labels and the ground-truth labels (in parentheses) are shown in legend. (b) The distribution of training data before the label-specific feature correction procedure. The decision boundary (solid line) induced from noisy PML training examples generalizes poorly on clean test data. (c) The distribution of training data after label-specific feature correction procedure. By correcting features, training examples with wrong annotations (green dots) behave like clean ones (blue dots). Therefore, the desired decision boundary (dotted line) can be induced with no further need for disambiguation. In (b) and (c), the top rows show visualizations for the first label and the bottom rows for the second label. TP (respectively TN) means the provided label and the ground-truth label are both positive (respectively negative), while FP denotes the provided label is positive but the ground-truth label is negative.

distribution of training examples after feature correction. The correction procedure relocates instances in the feature space and eliminates potential ambiguity in training examples. For instance, for the first label, ambiguity in noisy examples with false positive labels (data in the 4th cluster) is thoroughly eliminated, so that they behave like other clean examples with true positive labels (data in the 1st, 2nd, and 3rd clusters). With these corrected training examples, the desired prediction model can be easily induced (in Figure 3(c)).

4.2 Comparative studies

4.2.1 *Experimental setup*

Data sets. Five real-world and a number of synthetic PML data sets are employed for comprehensive performance evaluation. Table 1 summarizes detailed properties of each data set. As shown in Table 1, the first five data sets are real-world PML data sets, which possess both inaccurate annotations collected from web users and clean annotations further examined by human labelers. While the last six data sets are multi-label benchmarks, including yeast, Corel16k-s1, delicious, iaprtc12, espgame, and mediamill.

Following [8,42], a synthetic PML data set is generated from a multi-label data set by injecting random label noises. Specifically, for data sets with more than 100 class labels, their rare labels are first filtered out to keep the 15 most frequent labels and instances with no relevant label removed. Then, label noises are injected into the set of candidate labels by randomly flipping an instance's irrelevant labels with a probability. For thorough evaluation, two levels of label noise are considered in our experiments.

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

Evaluation metrics. Five widely-used evaluation metrics [13] for multi-label learning are employed to evaluate the performance of each approach, including average precision, ranking loss, one-error, coverage, and hamming loss. Detailed definitions of these metrics can be found in [13].

Implementation details. The embedding function $e_{\boldsymbol{\phi}}$ is instantiated by a two-layer fully-connected neural network with ReLU activations, where the hidden dimensionality is set as 512. And the prediction model $f_k(\cdot;\boldsymbol{\theta}_k)$ for each class label is implemented as a linear model with sigmoid activation. In the feature correction function, the two hyper-networks $w_k(\cdot;\boldsymbol{\psi}_k)$ and $b_k(\cdot;\boldsymbol{\psi}_k)$ are implemented as lookup tables since they are only used in training. The loss function ℓ employed to compute the empirical risk is the binary cross entropy loss, which is a commonly-used loss for MLL problems. For network

Table 1 Characteristics of the experimental data sets. The first five are real-world PML data sets and the last six are multi-label data sets employed to generate synthetic PML data sets. a) http://palm.seu.edu.cn/zhangml/. b) http://mulan.sourceforge.net/datasets.html. c) http://lear.inrialpes.fr/people/guillaumin/data.php.

Dataset	#Examples	#Features	#Labels	Cardinality	Domain
YeastBP	6139	6139	217	5.54	Biology[a)]
YeastCC	6139	6139	50	1.35	Biology[a)]
YeastMF	6139	6139	39	1.01	Biology[a)]
Music_emotion	6833	98	11	2.42	Music[a)]
Music_style	6839	98	10	1.44	Music[a)]
yeast	2417	103	14	4.24	Biology[b)]
Corel16k-s1	13766	500	153	2.86	Images[b)]
delicious	16105	500	983	19.02	Text[b)]
iaprtc12	19627	1000	291	5.72	Images[c)]
espgame	20770	1000	268	4.69	Images[c)]
mediamill	43907	120	101	4.38	Video[b)]

optimization, Adam with a batch size of 128, weight decay of 10^{-4}, and momentums of 0.999 and 0.9 is employed for 500 epochs.

4.2.2 *Experimental results*

PASE[2)] is compared against six well-established PML approaches with parameter configurations suggested in respective literature.

• FPML [40]. FPML employs the low-rank approximation of the instance-label association matrix to estimate the labeling confidence and simultaneously induces a multi-label predictor. $[\lambda_1 = 0.1, \lambda_2 = 1, \lambda_3 = 10]$.

• PARVLS [2]. A two-step approach which elicits credible labels via label propagation and then induces multi-label predictor by virtual label splitting. $[k = 10$, $\alpha = 0.95$, and thr $= 0.9]$.

• NATAL [49]. NATAL regards the task of PML as a feature completion problem and completes missing features with smoothness and low-rank assumptions. [grid search for $\alpha, \gamma \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ and $\beta \in \{10^{-7}, 10^{-6}, \ldots, 10^{-3}\}$].

• PML-MD [42]: PML-MD disambiguates between ground-truth and noisy labels in a meta-learning fashion and learns multi-label predictor with a confidence-weighted ranking loss.

• UPML-HL [53]. The desired prediction model is induced by optimizing an unbiased estimator of the Hamming loss.

• UPML-RL [53]. The desired prediction model is induced by optimizing an unbiased estimator of the Ranking loss.

Following [42], we take out 10% examples in each data set as a hold-out validation set to perform meta-learning for PML-MD and our PASE approach. The remaining 90% examples are randomly split into training set and test set with a ratio of 9 : 1 for training and evaluation respectively. During training, PML-MD and PASE perform meta-learning on the noisy training set and the clean validation set. For fair comparison, the clean validation set is also employed to train other approaches. Training details can be found in Appendix A. All the deep approaches share the same neural network structure and the same optimizer with the learning rate[3)] searched in {1e−3, 3e−3, 1e−2, 3e−2}. For each data set, we repeat the random splitting process ten times and record the average predictive performance across ten training/test trials.

Tables 2 and 3 report detailed experimental results in terms of each evaluation metric. Furthermore, we conduct the Wilcoxon signed-ranks test [54] (at 0.05 significance level) to analyze whether PASE is statistically superior to other comparing approaches. The p-value statistics on each evaluation metric are reported in Table 4. Based on these results, it is impressive to observe the following.

• Across all evaluation metrics, PASE achieves the best performance in 79% cases over all real-world data sets and synthetic data sets under varied noise levels.

• Compared with UPML-HL and UPML-RL which rely on unbiased loss estimators to tackle the PML problem, PASE significantly outperforms them in terms of all evaluation metrics.

2) Code package of PASE is publicly available at: https://palm.seu.edu.cn/zhangml/files/PASE.rar.

3) For PASE, we set the meta learning rate μ as 1 in all experiments and only search the inner learning rate η.

Table 2 Predictive performance of each comparing approach (mean±std. deviation) in terms of average precision and ranking loss, where •/◦ indicates PASE is significantly superior/inferior to one comparing approach via paired t-test at 0.05 significance level. ↑ (↓) indicates the larger (smaller) the value, the better the performance. The best results are shown in bold.

Data set	Noise level	Average precision ↑						
		FPML	PARVLS	NATAL	PML-MD	UPML-HL	UPML-RL	PASE
YeastBP		0.284±0.019•	0.173±0.007•	0.354±0.019•	0.275±0.011•	0.209±0.018•	0.089±0.005•	**0.362±0.018**
YeastCC		0.462±0.019•	0.297±0.018•	0.561±0.024•	0.507±0.031•	0.540±0.021•	0.400±0.021•	**0.589±0.029**
YeastMF		0.327±0.018•	0.204±0.009•	0.406±0.025•	0.426±0.020•	0.386±0.019•	0.235±0.028•	**0.464±0.028**
Music_emotion		0.641±0.011•	0.617±0.010•	0.626±0.012•	0.669±0.010	0.658±0.012•	0.669±0.011•	**0.673±0.011**
Music_style		0.725±0.010•	0.730±0.012•	0.574±0.012•	0.738±0.009•	0.735±0.011•	0.729±0.011•	**0.751±0.010**
yeast	L	0.747±0.025•	0.750±0.025•	0.578±0.018•	0.755±0.025	0.750±0.025•	0.735±0.031•	**0.760±0.025**
	H	0.742±0.024•	0.723±0.021•	0.574±0.020•	0.747±0.023•	0.740±0.026•	0.731±0.024•	**0.754±0.026**
Corel16k-s1	L	0.479±0.014•	0.401±0.009•	0.442±0.010•	0.486±0.017	0.491±0.012•	0.476±0.010•	**0.500±0.010**
	H	0.457±0.013•	0.385±0.009•	0.407±0.009•	0.428±0.007•	0.451±0.013•	0.429±0.011•	**0.469±0.010**
delicious	L	0.619±0.006•	0.611±0.005•	0.595±0.007•	0.643±0.007•	0.623±0.011•	0.647±0.007•	**0.657±0.006**
	H	0.606±0.006•	0.587±0.004•	0.590±0.007•	0.609±0.007•	0.623±0.006•	0.619±0.007•	**0.629±0.006**
iaprtc12	L	0.588±0.008•	0.565±0.009•	0.460±0.010•	0.588±0.007•	0.582±0.008•	0.558±0.007•	**0.607±0.011**
	H	**0.573±0.009**◦	0.526±0.007•	0.442±0.010•	0.559±0.007	0.546±0.008•	0.530±0.009•	0.561±0.007
espgame	L	0.460±0.009•	0.423±0.006•	0.395±0.008•	0.495±0.007	0.476±0.013•	0.475±0.008•	**0.498±0.007**
	H	0.442±0.009•	0.387±0.006•	0.372±0.008•	**0.478±0.007**	0.449±0.010•	0.447±0.006•	**0.478±0.007**
mediamill	L	0.784±0.004•	0.784±0.004•	0.703±0.005•	0.780±0.003•	0.790±0.005•	0.781±0.004•	**0.814±0.004**
	H	0.774±0.004•	0.753±0.004•	0.672±0.006•	0.760±0.004•	0.784±0.004•	0.765±0.006•	**0.799±0.005**
Data set	Noise level	Ranking loss ↓						
		FPML	PARVLS	NATAL	PML-MD	UPML-HL	UPML-RL	PASE
YeastBP		0.272±0.011•	0.322±0.007•	0.244±0.013•	0.217±0.011•	0.300±0.031•	0.409±0.013•	**0.184±0.006**
YeastCC		0.237±0.011•	0.315±0.018•	0.173±0.014•	0.160±0.017•	0.189±0.014•	0.194±0.016•	**0.140±0.015**
YeastMF		0.325±0.011•	0.372±0.016•	0.299±0.020•	0.221±0.017•	0.280±0.017•	0.387±0.035•	**0.206±0.019**
Music_emotion		0.230±0.011•	0.241±0.009•	0.252±0.014•	**0.208±0.008**	0.227±0.010•	0.209±0.011	0.211±0.009
Music_style		0.139±0.007•	0.136±0.007	0.292±0.010•	**0.129±0.006**◦	0.145±0.009•	0.139±0.007•	0.132±0.006
yeast	L	0.184±0.019•	0.178±0.018•	0.389±0.017•	0.177±0.020	0.187±0.018•	0.188±0.024•	**0.172±0.020**
	H	0.189±0.018•	0.203±0.016•	0.393±0.018•	0.181±0.019	0.192±0.017•	0.192±0.020•	**0.180±0.020**
Corel16k-s1	L	0.285±0.007•	0.330±0.007•	0.321±0.008•	0.257±0.019	0.261±0.010	**0.251±0.007**	0.253±0.008
	H	0.308±0.007•	0.351±0.007•	0.368±0.008•	0.318±0.008•	0.292±0.010•	0.292±0.011	**0.284±0.010**
delicious	L	0.266±0.005•	0.273±0.002•	0.292±0.006•	0.240±0.005•	0.265±0.007•	0.238±0.005•	**0.233±0.006**
	H	0.278±0.005•	0.292±0.003•	0.295±0.007•	0.275±0.004•	0.266±0.004•	0.258±0.006	**0.257±0.006**
iaprtc12	L	0.217±0.005•	0.242±0.006•	0.326±0.008•	0.201±0.003	0.220±0.007•	0.217±0.005•	**0.196±0.010**
	H	0.231±0.005	0.279±0.007•	0.347±0.008•	**0.220±0.003**◦	0.245±0.009•	0.239±0.006•	0.228±0.006
espgame	L	0.304±0.007•	0.335±0.005•	0.385±0.006•	0.262±0.005	0.278±0.008•	0.266±0.005•	**0.259±0.003**
	H	0.324±0.008•	0.370±0.005•	0.415±0.006•	**0.270±0.005**◦	0.306±0.012•	0.292±0.006•	0.280±0.008
mediamill	L	0.140±0.003•	0.138±0.003•	0.219±0.004•	0.141±0.002•	0.144±0.006•	0.134±0.003•	**0.121±0.004**
	H	0.147±0.003•	0.168±0.003•	0.261±0.004•	0.152±0.003•	0.150±0.004•	0.142±0.003•	**0.134±0.004**

• Meanwhile, PASE achieves much better performance against approaches based on disambiguation strategy. Specifically, PASE is statistically superior to PML-MD in terms of all evaluation metrics except coverage. Note that PASE and PML-MD both tackle the PML problem under the meta-learning framework. The superior performance of PASE against PML-MD demonstrates it is a promising strategy to eliminate potential ambiguity in PML data via learning label-specific feature corrections.

4.3 Further analyses

4.3.1 *Ablation studies*

PASE deals with the PML problem by merely exerting label-specific corrections in the feature space. We have provided an intuitive explanation for the rationality of PASE in Figure 2. Here, ablation studies are further conducted to validate the effectiveness of this feature correction strategy for the PML problem. We implement two variants: PASE-sc and PASE-nc. PASE-sc learns to exert a feature correction shared among all class labels, with no consideration of the distinct characteristics of each class label. While

Table 3 Predictive performance of each comparing approach (mean±std. deviation) in terms of one-error, coverage and hamming loss, where •/◦ indicates PASE is significantly superior/inferior to one comparing approach via paired t-test at 0.05 significance level. ↑ (↓) indicates the larger (smaller) the value, the better the performance. The best results are shown in bold.

Data set	Noise level	One-error ↓						
		FPML	PARVLS	NATAL	PML-MD	UPML-HL	UPML-RL	PASE
YeastBP		0.783±0.015•	0.961±0.009•	**0.726±0.022**	0.806±0.017•	0.833±0.023•	0.957±0.008•	0.732±0.018
YeastCC		0.831±0.021•	0.946±0.008•	0.796±0.019•	0.816±0.013•	0.805±0.018•	0.864±0.014•	**0.779±0.021**
YeastMF		0.902±0.013•	0.978±0.005•	0.861±0.011•	0.857±0.011•	0.871±0.014•	0.950±0.008•	**0.843±0.014**
Music_emotion		0.431±0.018•	0.463±0.020•	0.443±0.020•	0.380±0.021•	0.389±0.020•	0.386±0.019•	**0.367±0.022**
Music_style		0.369±0.015•	0.367±0.019•	0.531±0.021•	0.353±0.016•	0.340±0.017•	0.357±0.021•	**0.325±0.016**
yeast	L	0.230±0.041	**0.225±0.044**	0.410±0.048•	0.241±0.026	0.249±0.043	0.243±0.037	0.240±0.039
	H	0.235±0.037	0.241±0.032	0.410±0.046•	0.238±0.034	0.244±0.046•	0.247±0.037•	**0.223±0.042**
Corel16k-s1	L	0.634±0.021•	0.721±0.010•	0.664±0.014•	0.653±0.017•	0.623±0.016•	0.665±0.020•	**0.610±0.016**
	H	0.655±0.021	0.742±0.013•	0.692±0.013•	0.707±0.013•	0.674±0.018•	0.713±0.016•	**0.649±0.012**
delicious	L	0.389±0.014•	0.392±0.011•	0.444±0.012•	0.372±0.019•	0.384±0.027•	0.360±0.014•	**0.344±0.012**
	H	0.409±0.013•	0.424±0.011•	0.449±0.013•	0.412±0.018•	**0.374±0.008**	0.401±0.018•	0.376±0.011
iaprtc12	L	0.460±0.014•	0.477±0.012•	0.628±0.013•	0.476±0.013•	0.466±0.013•	0.530±0.013•	**0.444±0.014**
	H	**0.478±0.016•**	0.518±0.009•	0.647±0.014•	0.511±0.012•	0.510±0.014•	0.559±0.014•	0.494±0.012
espgame	L	0.644±0.016•	0.687±0.010•	0.707±0.015•	0.609±0.009	0.619±0.019•	0.642±0.013•	**0.606±0.012**
	H	0.664±0.014•	0.728±0.011•	0.719±0.015•	0.629±0.011	0.654±0.018•	0.658±0.010•	**0.621±0.012**
mediamill	L	0.135±0.005•	0.133±0.006•	0.188±0.007•	0.136±0.006•	0.126±0.007•	0.138±0.005•	**0.106±0.005**
	H	0.146±0.006•	0.166±0.007•	0.188±0.007•	0.152±0.005•	0.124±0.004•	0.163±0.009•	**0.113±0.007**
Data set	Noise level	Coverage ↓						
		FPML	PARVLS	NATAL	PML-MD	UPML-HL	UPML-RL	PASE
YeastBP		0.350±0.017•	0.392±0.013•	0.316±0.017•	0.263±0.014•	0.397±0.027•	0.491±0.020•	**0.240±0.010**
YeastCC		0.133±0.011•	0.173±0.018•	0.101±0.008•	0.088±0.012•	0.104±0.012•	0.108±0.010•	**0.080±0.010**
YeastMF		0.161±0.011•	0.181±0.012•	0.142±0.015•	0.107±0.013	0.131±0.009•	0.193±0.028•	**0.100±0.012**
Music_emotion		0.391±0.009•	0.402±0.007•	0.414±0.013•	0.371±0.008	0.397±0.006•	**0.370±0.010**	0.375±0.008
Music_style		0.197±0.009•	0.194±0.008	0.350±0.011•	**0.186±0.006**◦	0.205±0.009•	0.198±0.006•	0.193±0.006
yeast	L	0.485±0.019•	0.481±0.021•	0.666±0.013•	0.468±0.026	0.488±0.021•	0.476±0.019•	**0.459±0.025**
	H	0.493±0.019•	0.527±0.018•	0.672±0.014•	**0.467±0.019**	0.496±0.019•	0.480±0.022	0.473±0.023
Corel16k-s1	L	0.388±0.007•	0.428±0.008•	0.431±0.007•	0.357±0.020	0.357±0.013	**0.345±0.008**◦	0.353±0.006
	H	0.411±0.008•	0.448±0.007•	0.477±0.007•	0.420±0.008•	0.390±0.011	0.391±0.014	**0.386±0.013**
delicious	L	0.576±0.005•	0.583±0.004•	0.593±0.006•	0.537±0.006•	0.569±0.005•	0.534±0.006	**0.530±0.006**
	H	0.588±0.005•	0.603±0.004•	0.597±0.006•	0.577±0.006•	0.571±0.004•	**0.555±0.008**	0.556±0.009
iaprtc12	L	0.364±0.006•	0.395±0.007•	0.482±0.007•	0.342±0.005	0.375±0.009•	0.356±0.006•	**0.341±0.013**
	H	0.381±0.006	0.436±0.008•	0.504±0.007•	**0.363±0.004**◦	0.400±0.013•	0.378±0.005	0.377±0.010
espgame	L	0.423±0.008•	0.452±0.006•	0.509±0.004•	**0.373±0.006**	0.397±0.008•	0.375±0.006	0.374±0.007
	H	0.444±0.008•	0.483±0.004•	0.536±0.004•	**0.381±0.006**◦	0.425±0.013•	0.404±0.006•	0.396±0.008
mediamill	L	0.411±0.005•	0.407±0.005•	0.492±0.006•	0.407±0.004•	0.418±0.014•	0.399±0.007•	**0.380±0.007**
	H	0.421±0.005•	0.447±0.006•	0.526±0.005•	0.424±0.006•	0.427±0.006•	**0.405±0.005**	0.407±0.009
Data set	Noise level	Hamming loss ↓						
		FPML	PARVLS	NATAL	PML-MD	UPML-HL	UPML-RL	PASE
YeastBP		0.025±0.001•	0.029±0.001•	0.068±0.008•	0.041±0.002•	0.025±0.001•	0.025±0.001•	**0.024±0.001**
YeastCC		0.026±0.002•	0.027±0.005•	0.037±0.006•	0.047±0.005•	0.026±0.002•	0.027±0.002•	**0.024±0.002**
YeastMF		0.027±0.002•	**0.025±0.002**	0.029±0.004•	0.039±0.002•	0.028±0.001•	0.026±0.002	**0.025±0.002**
Music_emotion		0.198±0.003•	0.204±0.005•	0.202±0.003•	0.194±0.004•	**0.188±0.004**	0.191±0.003•	0.189±0.004
Music_style		0.117±0.003•	0.119±0.003•	0.132±0.004•	0.125±0.004•	0.111±0.003•	0.114±0.004•	**0.110±0.003**
yeast	L	0.209±0.012	0.208±0.010	0.278±0.007•	0.214±0.012	**0.201±0.009**◦	0.224±0.031	0.207±0.012
	H	0.212±0.010	0.257±0.009•	0.281±0.008•	0.210±0.011	0.209±0.012	0.219±0.013•	**0.208±0.012**
Corel16k-s1	L	0.125±0.002•	0.136±0.002•	0.118±0.002•	0.119±0.003•	0.118±0.002•	0.118±0.002•	**0.117±0.001**
	H	0.125±0.001	0.219±0.004•	0.118±0.002•	0.118±0.002•	0.120±0.007	0.118±0.002•	**0.117±0.001**
delicious	L	0.253±0.002•	0.255±0.002•	0.265±0.003•	0.257±0.002•	0.248±0.006•	0.243±0.003•	**0.235±0.002**
	H	0.257±0.002•	0.298±0.002•	0.268±0.003•	0.273±0.004•	0.248±0.007	0.257±0.005•	**0.247±0.004**
iaprtc12	L	0.142±0.001•	0.140±0.001•	0.151±0.002•	0.145±0.002•	0.139±0.003•	0.145±0.002•	**0.136±0.001**
	H	0.144±0.002•	0.228±0.003•	0.152±0.002•	0.148±0.002•	0.143±0.003	0.149±0.002•	**0.142±0.002**
espgame	L	0.138±0.002•	0.138±0.003•	0.133±0.002•	0.131±0.002•	0.132±0.002•	0.132±0.002•	**0.130±0.002**
	H	0.139±0.002•	0.233±0.003•	0.133±0.002•	0.132±0.002•	0.132±0.002•	0.132±0.002•	**0.130±0.002**
mediamill	L	0.154±0.002•	0.153±0.002•	0.173±0.002•	0.159±0.002•	0.150±0.004•	0.159±0.002•	**0.141±0.002**
	H	0.158±0.002•	0.170±0.002•	0.172±0.002•	0.167±0.001•	0.151±0.003•	0.161±0.002•	**0.147±0.002**

PASE-nc further removes the feature correction procedure and directly induces a multi-label predictor with provided examples in both the training and validation sets.

Table 4 Summary of the Wilcoxon signed-ranks test for PASE against other comparing approaches at 0.05 significance level. p-values are shown in the brackets.

	FPML	PARVLS	NATAL	PML-MD	UPML-HL	UPML-RL
Average precision	win [4.2e−4]	win [2.9e−4]	win [2.9e−4]	win [4.4e−4]	win [2.9e−4]	win [2.9e−4]
Hamming loss	win [2.9e−4]	win [4.4e−4]	win [2.9e−4]	win [2.9e−4]	win [6.0e-3]	win [2.9e−4]
One-error	win [9.2e−4]	win [3.5e−4]	win [3.5e−4]	win [2.9e−4]	win [3.5e−4]	win [2.9e−4]
Coverage	win [2.9e−4]	win [2.9e−4]	win [2.9e−4]	tie [8.4e−2]	win [2.9e−4]	win [1.2e−2]
Ranking loss	win [2.9e−4]	win [2.9e−4]	win [2.9e−4]	win [1.5e−2]	win [2.9e−4]	win [7.1e−4]

Table 5 Summary of the Wilcoxon signed-ranks test for PASE against its variants at 0.05 significance level. p-values are shown in the brackets.

	PASE-SC	PASE-NC	PASE-NM
Average precision	win [1.1e−3]	win [3.5e−4]	win [2.9e−4]
Hamming loss	win [4.4e−4]	win [2.9e−4]	win [4.3e−4]
One error	win [7.4e−3]	win [3.5e−4]	win [2.9e−4]
Coverage	win [7.1e−4]	win [2.9e−4]	win [2.9e−4]
Ranking loss	win [6.0e−4]	win [2.9e−4]	win [2.9e−4]

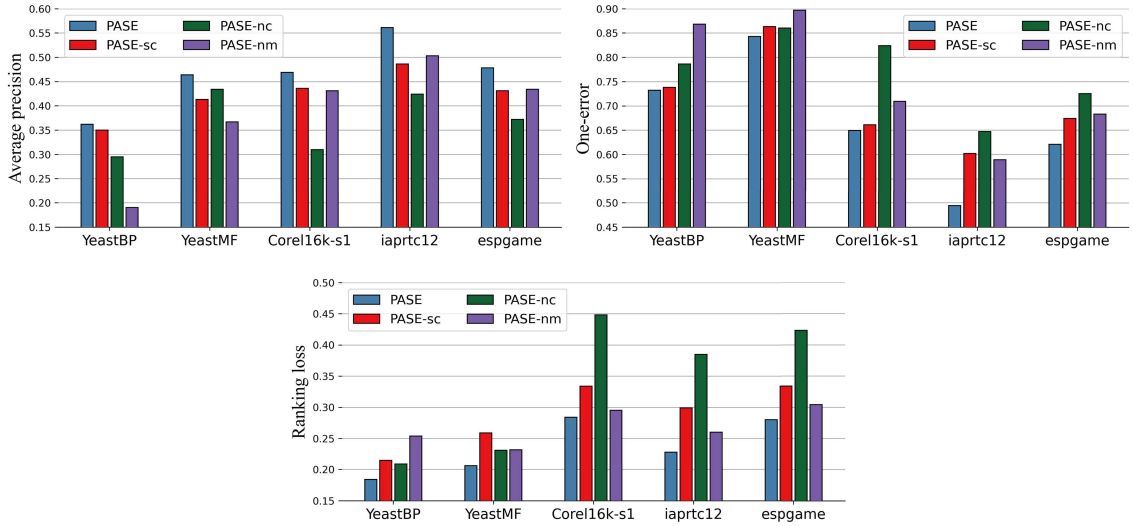

Figure 4 (Color online) Predictive performance of PASE and its variants in terms of average precision, one-error, and ranking loss.

We employ ten-fold cross validation on all the real-world and synthetic PML data sets. Table 5 summarizes the p-value statistics of the Wilcoxon signed-ranks test at a 0.05 significance level, and Figure 4 shows the detailed experimental results on representative data sets in terms of average precision, one-error, and ranking loss. A statistically significant performance degradation can be witnessed in these two variants, demonstrating that feature correction is an effective strategy for the PML problem and that the consideration of label-specific properties is important.

In PASE, we formalize the learning problem of the feature correction function and prediction model under a meta-learning framework. To validate its effectiveness, we implement a variant named PASE-nm, which gets rid of the meta-learning framework by jointly optimizing the feature correction function and prediction model with empirical risk minimization on both the training and validation sets. As shown in Table 5 and Figure 4, removing the meta-learning framework results in statistically significant performance degradation.

In Appendix B, we also conduct ablation studies on the specific instantiation of the feature correction function in (2). For simplicity, we implement the correction function as a simple affine transformation in PASE. To demonstrate that such an affine transformation is a reasonable choice to make PASE work well, we decompose it and employ its two components, namely, a scaling transformation and a translation transformation, to implement its two variants respectively. Experimental results validate the superiority of the considered affine transformation to these simplified variants.

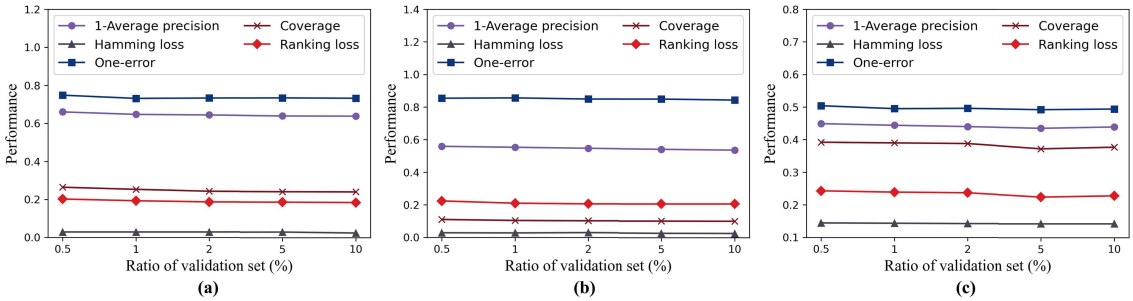

Figure 5 (Color online) Performance of PASE with varying sizes of validation sets. (a) YeastBP; (b) YeastMF; (c) iaprtc12.

4.3.2 *Study on the size of validation set*

Figure 5 presents some illustrative examples of how the performance of PASE changes when the size of the validation set changes. We take out $r\%$ examples in each data set as the validation set and alter the value of r in the range of $\{0.5, 1, 2, 5, 10\}$. The performance of PASE is relatively stable as the value of r changes, which serves as a desirable property for using PASE in practice. If there is no clean data, only a few instances need to be accurately annotated to obtain promising performance.

4.3.3 *Complexity analysis*

Let b be the batch size and $\hat{d}$ denote a proxy of the hidden dimensionality of the network. The time complexity of the label-specific feature correction procedure in PASE is $\mathcal{O}(bq\hat{d})$ with q class labels. Figure C.1 in Appendix C reports the empirical training and test time of each comparing approach, which shows that the time overhead of PASE is comparable to those of existing approaches.

4.3.4 *Scalability studies on large-scale data sets*

We further conduct scalability studies on two large-scale multi-label data sets: MS-COCO [55] and Visual Genome [56]. MS-COCO contains about 120000 images from 80 daily-life categories. As the official test set has no publicly available annotation, we follow the routine to train models on the 80000 training set and report the final performance on the 40000 official validation set. Visual Genome contains about 110000 images from 80138 categories. As most categories have few positive instances, we follow [57] to keep the 200 most frequent categories and randomly split the resulting data set (VG-200) into 100000 training sets and 10000 test sets. For both data sets, we randomly sample 10% examples from the training set as a hold-out validation set for meta-learning.

Similar to the way used in Subsection 4.2.1, a synthetic PML data set is generated from a multi-label data set by randomly flipping an instance's irrelevant labels into candidate ones. For thorough evaluation, two levels of label noise (i.e., low- and high-level label noise) are considered in our experiments.

We adopt the ImageNet-pretrained ResNet18 [58] as the backbone to deal with raw image data. As traditional approaches (FPML, PARVLS, and NATAL) cannot optimize the backbone during training, we only consider deep approaches in experiments for fair comparison.

For network optimization, we adopt AdamW [59] with a batch size of 32 and a 1-cycle policy [60] with a maximal learning rate of 2.5e−5. The weight decay factor is set to 10^{-2}. Following [57], we resize the input image to 512×512 and randomly choose a number from $\{512, 448, 384, 320, 256\}$ as the width and height to crop a patch. Then, the cropped patch is further resized to 448×448 and randomly horizontally flipped.

We repeat training/test trials three times with different random seeds and report the average predictive performance in Table 6. Impressively, PASE still achieves much better performance than existing approaches on large-scale multi-label data sets. Although PASE and PML-MD both tackle the PML problem in a meta-learning manner, PASE consistently outperforms disambiguation-based PML-MD over all cases with varied noise levels. We speculate that learning a labeling confidence matrix that describes rather precise ranking relationships among class labels is much more difficult than learning just proper feature corrections, especially when the rate of label noise is high.

To understand the importance of each designing element that distinguishes PASE from existing PML approaches on large-scale data sets, a two-step procedure is devised to degrade PASE into a naive baseline. Since PASE learns from PML data by exerting label-specific corrections in the feature space, the first step

Table 6 Predictive performance of each comparing approach (mean±std. deviation) on large-scale data sets. ↑ (↓) indicates the larger (smaller) the value, the better the performance. Mean average precision (mAP) [57] is the dominant evaluation metric on these large-scale multi-label data sets. The best results are shown in bold.

Data set	Noise level	mAP ↑					
		PML-MD	UPML-HL	UPML-RL	PASE	PASE-nc	PASE-sc
MS-COCO	L	35.79±2.82	47.10±0.62	33.67±2.95	**54.02±0.28**	26.16±2.25	53.35±0.68
	H	16.88±1.54	36.50±2.55	32.18±1.40	**41.35±0.41**	9.49±0.50	40.56±1.22
VG-200	L	25.93±0.74	31.88±0.46	25.10±0.84	**35.08±0.07**	26.60±2.40	34.36±0.66
	H	14.00±0.03	28.05±0.27	23.52±0.75	**29.67±0.13**	12.16±0.58	29.43±0.02
Data set	Noise level	Average precision ↑					
		PML-MD	UPML-HL	UPML-RL	PASE	PASE-nc	PASE-sc
MS-COCO	L	0.583±0.015	0.696±0.004	0.585±0.028	**0.735±0.004**	0.452±0.038	0.702±0.008
	H	0.410±0.011	0.624±0.021	0.579±0.014	**0.642±0.004**	0.322±0.005	0.639±0.007
VG-200	L	0.439±0.007	0.477±0.004	0.362±0.010	**0.496±0.002**	0.366±0.018	0.445±0.012
	H	0.323±0.006	**0.451±0.003**	0.351±0.008	0.443±0.002	0.245±0.004	0.446±0.003
Data set	Noise level	Hamming loss ↓					
		PML-MD	UPML-HL	UPML-RL	PASE	PASE-nc	PASE-sc
MS-COCO	L	0.031±0.001	0.031±0.001	0.035±0.004	**0.023±0.001**	0.042±0.008	0.024±0.002
	H	0.033±0.001	0.030±0.001	0.034±0.003	**0.025±0.001**	0.035±0.002	0.026±0.001
VG-200	L	0.051±0.001	0.050±0.001	0.073±0.004	**0.047±0.001**	0.060±0.005	0.049±0.001
	H	0.054±0.001	0.050±0.001	0.073±0.004	**0.048±0.001**	0.058±0.001	0.050±0.001

is to remove the consideration of the distinct characteristics of each class label. We name such a degraded version PASE-sc, which learns from PML data by exerting a feature correction shared among all class labels. As shown in Table 6, PASE-sc is inferior to PASE and performs comparably with the second best approach UPML-HL. This indicates that label-specific features, which is a fascinating strategy in supervised learning, can still help with weakly-supervised learning and is the key element in PASE that allows it to outperform other PML approaches. Based on PASE-sc, the second step is to further remove the feature correction procedure. This degraded version named PASE-nc is actually a naive baseline approach that simply treats all candidate labels as relevant ones. As shown in Table 6, a significant performance drop is witnessed in PASE-np, which demonstrates feature correction is a basic component to make PASE competitive.

5 Conclusion

To tackle the PML problem, existing approaches mainly focus on manipulating the label space with disambiguation strategies, whereas the manipulation in the feature space is rarely investigated. In this paper, we demonstrate that it is possible to learn from PML data by correcting instance features in a label-specific manner, which complements the currently dominant disambiguation strategy and suggests a promising future direction for the PML problem. One by-product of the feature correction strategy is the variation of the underlying ratio of positive and negative instances, as all the positively annotated instances are treated as positive ones during the learning process. Thus, a natural direction for future work is to inspect this interesting phenomenon and explore whether it is possible to incorporate rebalancing methods for further improvement.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62225602) and Big Data Computing Center of Southeast University. The authors wish to thank the associate editor and anonymous reviewers for their helpful comments and suggestions.

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Xie M K, Huang S J. Partial multi-label learning. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, 2018. 4302–4309
2 Zhang M L, Fang J P. Partial multi-label learning via credible label elicitation. IEEE Trans Pattern Anal Mach Intell, 2021, 43: 3587–3599
3 Lyu G, Feng S, Li Y. Partial multi-label learning via probabilistic graph matching mechanism. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020. 105–113

4 Wang H, Liu W, Zhao Y, et al. Discriminative and correlative partial multi-label learning. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, 2019. 3691–3697
5 Xu N, Liu Y P, Geng X. Partial multi-label learning with label distribution. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, 2020. 6510–6517
6 Cao N, Zhang T, Jin H. Partial multi-label optimal margin distribution machine. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence, Montreal, 2021. 2198–2204
7 Sun L, Feng S, Wang T, et al. Partial multi-label learning by low-rank and sparse decomposition. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, 2019. 5016–5023
8 Xie M K, Huang S J. Partial multi-label learning with noisy label identification. IEEE Trans Pattern Anal Mach Intell, 2021, 44: 3676–3687
9 Hospedales T M, Antoniou A, Micaelli P, et al. Meta-learning in neural networks: a survey. IEEE Trans Pattern Anal Mach Intell, 2021, 44: 5149–5169
10 Xu M, Guo L-Z. Learning from group supervision: the impact of supervision deficiency on multi-label learning. Sci China Inf Sci, 2021, 64: 130101
11 Yang W J, Li C Q, Jiang L X. Learning from crowds with robust support vector machines. Sci China Inf Sci, 2023, 66: 132103
12 Cour T, Sapp B, Taskar B. Learning from partial labels. J Mach Learn Res, 2011, 12: 1501–1536
13 Zhang M L, Zhou Z H. A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng, 2014, 26: 1819–1837
14 Liu W, Wang H, Shen X, et al. The emerging trends of multi-label learning. IEEE Trans Pattern Anal Mach Intell, 2021, 44: 7955–7974
15 Feng L, An B. Partial label learning with self-guided retraining. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, 2019. 3542–3549
16 Yan Y, Guo Y. Partial label learning with batch label correction. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, 2020. 6575–6582
17 Wang H, Xiao R, Li Y, et al. PiCO: contrastive label disambiguation for partial label learning. In: Proceedings of the 10th International Conference on Learning Representations, 2022
18 Lyu G, Wu Y, Feng S. Deep graph matching for partial label learning. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, Vienna, 2022. 3306–3312
19 Qiao C, Xu N, Geng X. Decompositional generation process for instance-dependent partial label learning. In: Proceedings of the 11th International Conference on Learning Representations, Kigali, 2023
20 Tang C Z, Zhang M L. Confidence-rated discriminative partial label learning. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, 2017. 2611–2617
21 Gong C, Liu T, Tang Y, et al. A regularization approach for instance-based superset label learning. IEEE Trans Cybern, 2018, 48: 967–978
22 Zhang M L, Yu F, Tang C Z. Disambiguation-free partial label learning. IEEE Trans Knowl Data Eng, 2017, 29: 2155–2167
23 Wu X, Zhang M L. Towards enabling binary decomposition for partial label learning. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, 2018. 2868–2874
24 Lv J, Xu M, Feng L, et al. Progressive identification of true labels for partial-label learning. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 6500–6510
25 Feng L, Lv J, Han B, et al. Provably consistent partial-label learning. In: Proceedings of Advances in Neural Information Processing Systems 33, 2020. 10948–10960
26 Boutell M R, Luo J, Shen X, et al. Learning multi-label scene classification. Pattern Recogn, 2004, 37: 1757–1771
27 Zhang M L, Zhou Z H. ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn, 2007, 40: 2038–2048
28 Elisseeff A, Weston J. A kernel method for multi-labelled classification. In: Proceedings of Advances in Neural Information Processing Systems 14, 2001. 681–687
29 Zhu Y, Kwok J T, Zhou Z H. Multi-label learning with global and local label correlation. IEEE Trans Knowl Data Eng, 2018, 30: 1081–1094
30 Wehrmann J, Cerri R, Barros R C. Hierarchical multi-label classification networks. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 5225–5234
31 Xu N, Shu J, Liu Y, et al. Variational label enhancement. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 10597–10606
32 Zhang M L, Wu L. LIFT: multi-label learning with label-specific features. IEEE Trans Pattern Anal Mach Intell, 2015, 37: 107–120
33 Guo Y, Chung F, Li G, et al. Leveraging label-specific discriminant mapping features for multi-label learning. ACM Trans Knowl Discov Data, 2019, 13: 1–23
34 Lin Y, Hu Q, Liu J, et al. MULFE: multi-label learning via label-specific feature space ensemble. ACM Trans Knowl Discov Data, 2022, 16: 1–24
35 Hang J Y, Zhang M L, Feng Y, et al. End-to-end probabilistic label-specific feature learning for multi-label classification. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022. 6847–6855
36 Huang J, Li G, Huang Q, et al. Learning label-specific features and class-dependent labels for multi-label classification. IEEE Trans Knowl Data Eng, 2016, 28: 3309–3323
37 Huang J, Li G, Huang Q, et al. Joint feature selection and classification for multilabel learning. IEEE Trans Cybern, 2018, 48: 876–889
38 Hang J Y, Zhang M L. Collaborative learning of label semantics and deep label-specific features for multi-label classification. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 9860–9871
39 Yu Z B, Zhang M L. Multi-label classification with label-specific feature generation: a wrapped approach. IEEE Trans Pattern Anal Mach Intell, 2021, 44: 5199–5210
40 Yu G, Chen X, Domeniconi C, et al. Feature-induced partial multi-label learning. In: Proceedings of the IEEE International Conference on Data Mining, Singapore, 2018. 1398–1403
41 Yan Y, Guo Y. Adversarial partial multi-label learning with label disambiguation. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021. 10568–10576
42 Xie M K, Sun F, Huang S J. Partial multi-label learning with meta disambiguation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2021. 1904–1912
43 Sun L, Feng S, Liu J, et al. Global-local label correlation for partial multi-label learning. IEEE Trans Multimedia, 2022, 24: 581–593
44 Li Z, Lyu G, Feng S. Partial multi-label learning via multi-subspace representation. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, 2020. 2612–2618
45 Lyu G, Feng S, Jin Y, et al. Prior knowledge regularized self-representation model for partial multilabel learning. IEEE Trans Cybern, 2023, 53: 1618–1628
46 Li F, Shi S, Wang H. Partial multi-label learning via specific label disambiguation. Knowledge-Based Syst, 2022, 250: 109093
47 Xu T, Xu Y, Yang S, et al. Learning accurate label-specific features from partially multilabeled data. IEEE Trans Neural Netw Learn Syst, 2024, 35: 10436–10450

48 Gong X, Yuan D, Bao W. Partial multi-label learning via large margin nearest neighbour embeddings. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022. 6729–6736
49 Lyu G, Feng S, Li Y. Noisy label tolerance: a new perspective of partial multi-label learning. Inf Sci, 2021, 543: 454–466
50 Ren M, Zeng W, Yang B, et al. Learning to reweight examples for robust deep learning. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 4331–4340
51 Shu J, Xie Q, Yi L, et al. Meta-Weight-Net: learning an explicit mapping for sample weighting. In: Proceedings of Advances in Neural Information Processing Systems 32, 2019. 1917–1928
52 Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, 2017. 1126–1135
53 Xie M K, Huang S J. CCMN: a general framework for learning with class-conditional multi-label noise. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 154–166
54 Wilcoxon F. Individual Comparisons by Ranking Methods. Berlin: Springer, 1992. 196–202
55 Lin T Y, Maire M, Belongie S J, et al. Microsoft COCO: common objects in context. In: Proceedings of the 13th European Conference on Computer Vision, Zurich, 2014. 740–755
56 Krishna R, Zhu Y, Groth O, et al. Visual genome: connecting language and vision using crowdsourced dense image annotations. Int J Comput Vis, 2017, 123: 32–73
57 Chen T, Pu T, Wu H, et al. Structured semantic transfer for multi-label recognition with partial labels. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022. 339–346
58 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016. 770–778
59 Loshchilov I, Hutter F. Decoupled weight decay regularization. In: Proceedings of the 7th International Conference on Learning Representations, New Orleans, 2019
60 Smith L N. A disciplined approach to neural network hyper-parameters: part 1 – learning rate, batch size, momentum, and weight decay. 2018. ArXiv:1803.09820