Knowledge-Based Systems 250 (2022) 109093

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Partial multi-label learning via specific label disambiguation

Feng Li, Shengfei Shi [*], Hongzhi Wang

School of Computer Science and Technology, Harbin Institute of Technology, Heilongjiang, 150001, China

ARTICLE INFO

Article history:
Received 7 January 2022
Received in revised form 16 May 2022
Accepted 17 May 2022
Available online 26 May 2022

Keywords:
Partial multi-label learning
Label-specific feature extraction
Specific label propagation

ABSTRACT

Partial Multi-Label Learning (PML) aims to learn a robust multi-label classifier from training data, where each instance is associated with a set of candidate labels, among which only a subset of them is relevant. Some existing methods consider the noise in the feature space and have made some achievements. However, they ignored that each label might be only related to a subset of original features, and the other features might be noise in different label spaces. Such as, the similarity between two instances may be different in different label spaces, which is crucial in PML. To tackle the problem, we propose a novel framework named **P**artial multi-l**A**bel learning via **S**pecific l**A**bel **D**isambiguation(**PASAD**), which tries to extract the label-specific features for disambiguation. Specifically, we first adapt the Hilbert–Schmidt Independence Criterion (HSIC) to identify the projection matrix for each label by maximizing the dependence between feature space and each label space. With these matrices, the instances can be mapped into each label-specific feature space to reduce irrelevant information. Besides, label correlations are considered to enrich the label-specific features. Afterward, we propose a specific label propagation method to estimate the labeling confidence values, which adapts the label propagation in each label-specific feature space and considers the interactions between different label spaces. We combine the two stages in an iterative manner. Finally, any binary classifier can be applied to induce a classification model by each label's new specific features and credible labels. Tremendous experimental results demonstrate the effectiveness and superiority of our proposed method.

1. Introduction

Multi-label learning (MLL) seeks to learn a classifier from the data set where each instance is associated with multiple classes simultaneously [1]. It has been widely studied for many years in various domains [2–4]. A strong assumption is that each instance has been annotated with its relevant labels precisely. However, precise labeling is difficult and costly to obtain in many real-world applications. Instead, It is more common for samples to be roughly labeled. That is, each instance is associated with a set of labels, which contains noise or irrelevant labels that are unknown. For example, as shown in Fig. 1, the image is roughly labeled by noisy annotators, and *sun*, *people*, *bird* are irrelevant labels. Due to the misleading noise labels, it is hard to learn a robust classifier for predicting. To handle this problem, Xie and Huang [5] proposed a learning framework named partial multi-label learning (PML), which provided an effective solution and attracted considerable attention.

Existing PML methods can be roughly classified into two categories: two-stage strategy and end-to-end strategy [6]. For the former, the whole learning process is divided into two stages, including first identifying the ground-truth labels from the candidate label set and learning a classifier using state-of-art MLL methods with reliable labels. PARTICLE [7] identified high-confidence reliable labels with label propagation technique. DRAMA [8] used the feature manifold and candidate label set to calculate the confidence of the labels, then trained a predict model with gradient boosting decision tree using the confidence directly. Besides, considering the different relevance among candidate labels, Xu et al. [9,10] proposed a novel label enhancement method that tried to recover the label distribution of each instance instead of disambiguation. They considered more label correlations and the topological information of the feature space was used to assist the recovery. For the latter, optimizing candidate labels and training model are carried out simultaneously. PML-fp and PML-lc [5] constrained the relationship of pair-wise labels to estimate the confidence values and induced a predict model. PML-LRS [11] employed the low-rank and sparse decomposition scheme to separate ground-truth labels and noise labels. MUSER [12] considered the feature noise and tried to learn from the both feature and label subspace to reduce the bias. PML-MD [13] tried to disambiguate in a meta-learning fashion with a different background setting.

[*] Corresponding author.
E-mail addresses: 21S003049@stu.hit.edu.cn (F. Li), shengfei@hit.edu.cn (S. Shi), wangzh@hit.edu.cn (H. Wang).

https://doi.org/10.1016/j.knosys.2022.109093
0950-7051/© 2022 Elsevier B.V. All rights reserved.

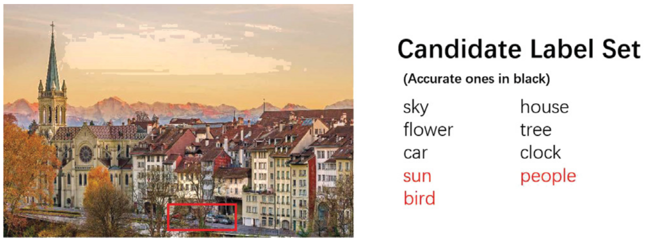


Fig. 1. An example of partial multi-label learning.

Although existing methods have their advantages, they ignored the different feature spaces associated with different labels, which is crucial for calculating the similarity or other information. That is, as shown in Fig. 1, when we judge whether *car* is a ground-truth label, the features related to *house* may be noise. So only using the same feature space to optimize candidate labels is not detailed enough, resulting in misleading information.

To solve this problem effectively and get a robust classifier, in this paper, we firstly adapt the Hilbert–Schmidt Independence Criterion (HSIC) to identify the projection matrix for each label space by maximizing the dependence between the projected feature information and label l_a space. In addition, considering there are much valuable information in label correlations and some labels may have no specific features, we use label correlations to enrich the specific feature space of each label. That is, for label l_a space, we select the r most relevant labels to construct its label kernel space for feature extraction. Secondly, we propose a specific label propagation for disambiguation. Specifically, we adapt the label propagation in each enriched label-specific feature space to estimate the labeling confidence values. Meanwhile, the interactions between different enriched label spaces are considered. That is, the current label space's r-nearest neighbor spaces will affect its label propagation according to the degree of correlation. Then high-confidence labels are selected to refine candidate labels. The above two stages are combined in an iterative manner in this paper. Finally, by using the specific features of each enriched label space and credible labels, we can apply any binary classifiers to induce the classification model. In the experimental part, we conduct extensive experiments on real-world and synthesized data sets to prove the superiority and effectiveness of our method.

The main contributions are as follows:

- We extract the enriched label-specific features for each label by adapting the Hilbert–Schmidt Independence Criterion (HSIC) to identify the projection matrices, considering using label correlations to enrich the label kernel spaces. In this way, the noise and negative impact of features associated with different labels can be removed.
- We propose a specific label propagation for disambiguation in each enriched label-specific feature space, where the interactions between different spaces are considered.
- With the enriched label-specific features for each label and credible labels, we can transform the PML into a set of binary classification problems, which has less noise of both feature space and label space. Then any binary classifiers can be applied to induce the classification model.
- We compare PASAD with several state-of-the-art MLL and PML methods on various PML data sets. Experimental results prove that PASAD is an effective strategy to solve the PML. Especially in real-world PML data sets, our method shows great superiority compared to the comparison methods.

The rest of this paper is organized as follows. First we review the related works on partial multi-label learning in Section 2. Then Section 3 introduces the details of the proposed model. In Section 4, the results of the comparison experiments and evaluation are reported. Finally, we conclude this paper in Section 5.

2. Releated work

Partial multi-label learning is a weakly supervised MLL framework, where each instance is roughly labeled by noisy annotator. It is developed from two popular learning frameworks: multi-label learning [1] and partial label learning [14].

2.1. Multi-Label Learning

Multi-Label Learning (MLL) deals with the data set where each instance is associated with a set of accurate labels. It has been studied for many years. Some works transformed this task into a set of binary classification problems [15,16]. However, they treated each label independently. So for better performance, many studies tried to exploit label correlations, including pairwise label correlations [17–20] and high-order correlations among all labels [21–24]. It is worth noting that MLL assumes each instance has been precisely labeled, which is almost impossible in the real world. In contrast, PML is a more realistic and weakly supervised MLL framework, where each instance is associated with a candidate label set and the ground-truth labels are not directly available.

2.2. Partial Label Learning

Partial Label Learning (PLL) is a single label task where each instance is associated with a candidate label set and only one is ground-truth [14,25,26]. To solve this problem, a direct strategy is disambiguation, which tries to recover the ground-truth label from the candidate label set. Some methods just made a prediction by averaging the outputs from all candidate labels [27,28], where each instance is treated equally. Another way is to regard the ground-truth label as a latent variable and then iteratively refine the model parameters [25,29–31]. Both PLL and PML learn from training data without precisely labeling, where each instance is assigned with a candidate label set. Nonetheless, PML is much more challenging as we need to learn a multi-label classifier instead of a multi-class one, and the number of ground-truth labels is unknown.

2.3. Feature extraction with Hilbert–Schmidt Independence Criterion

The Hilbert–Schmidt Independence Criterion (HSIC) [32] is an effective measure of dependence between two random variables by computing the Hilbert–Schmidt norm of the cross-covariance operator. Due to its simplicity and effectiveness, it has been adapted in many feature transforming work: Gangeh et al. [33] attempted to adapt HSIC to select informative genes from biological expression data. Bao et al. [34] and Zhu et al. [35] both tried to utilize HSIC to reduce the redundancy in feature space. However, Bao et al. [34] maximized the dependence between projected features and full label space while not considered the negative impacts of features associated with different labels. Zhu et al. [35] extracted features related to single label space while ignored the label correlations, which contains valuable information in MLL.

So in this paper, by combining the advantages of the above two methods, HSIC is adapted to identify a set of projection matrices that can map instances into each label-specific feature space, enriched by using label correlations. The effectiveness will be validated in Section 4.8.

2.4. Partial Multi-Label Learning

Partial Multi-Label Learning (PML) deals with the problem where each instance is associated with a set of candidate labels, in which some are ground-truth, and others are noise. Existing PML methods can be roughly divided into two categories [6]: (1) Two-stage strategy divides the learning process into two stages, including refining the candidate labels and learning a classifier with reliable labels. For example, PARTICLE [7] used the label propagation technique to get credible label sets. Wang et al. [8] tried to train a predict model with gradient boosting decision tree using the confidence values of labels, which were gained by the feature manifold. Xu et al. [9,10] used label enhancement method to recover the label distribution of each instance, which considered the topological information of the feature space and more label correlations. (2) End-to-end strategy is to optimize candidate labels and train model simultaneously. Xie and Huang [5] constrained the relationship of pair-wise labels and used a ranking model to induce a classifier. Sun et al. [11] employed low-rank and sparse matrix factorization to separate noise labels according to the low-rank assumption. Besides, MUSER [12] considered the noise in feature space and deposed both the label space and feature space into subspace to reduce the noise.

Despite the advances these methods have achieved, they did not consider that each label was only related to a subset of features, and the others would be noise during the disambiguation in this label space. Hence, in the next section, we propose a novel model named Partial Multi-Label Learning via Specific Label Disambiguation (PASAD). It aims to map the instances into each enriched label-specific feature space first and adapts a specific label propagation to identify the high-confidence labels. Then any binary classifiers can be applied to learn from the transformed training data.

Notations

$\mathcal{X}$	d-dimensional instance feature space
$\mathcal{D}$	Training set
$\mathcal{Q}$	Reproducing Kernel Hilbert Space mapped from $\mathcal{X}$
$\mathbf{X} \in \mathbb{R}^{d\times n}$	Instance matrix formed by the training set
$\mathbf{P} \in \mathbb{R}^{d\times d'}$	Projection matrix
$\mathbf{L} \in \mathbb{R}^{n\times n}$	Inner product matrix of labels in enriched label space
$\mathbf{R} \in \mathbb{R}^{q\times q}$	Label correlation matrix
$\hat{\mathbf{F}} \in \mathbb{R}^{n\times q}$	Confidence value matrix directly updated by the weighted sum of labeling confidence of its k-nearest neighbors in each enriched label space
Γ	Label correlations
$\mathcal{Y}$	Label space with q labels
C	Candidate label set
$\mathcal{F}$	Reproducing Kernel Hilbert Space mapped from $\mathcal{Y}$
$\mathbf{Y} \in \{0, 1\}^{q\times n}$	Candidate label matrix
$\mathbf{K} \in \mathbb{R}^{n\times n}$	Inner product matrix of instances in $\mathcal{Q}$
$\mathbf{S} \in \mathbb{R}^{n\times n}$	Instance similarity matrix
$\mathbf{F} \in \mathbb{R}^{n\times q}$	Confidence value matrix
$\tilde{\mathbf{F}} \in \mathbb{R}^{n\times q}$	Confidence value matrix after interacting between different enriched label spaces
Θ	Lagrange multipliers

3. The proposed method

In order to reduce the negative impacts of features associated with different labels, we first adapt HSIC to extract the features with each specific label in Section 3.1, where label correlations are used to enrich the label-specific feature spaces. In Section 3.2, we propose a specific label propagation method using the enriched features for disambiguation of each label, which considers the interactions between different label spaces. Then we combine the two stages into an iterative manner for better performance and give an efficient training Algorithm in Section 3.3. Finally we give the optimization solution in Section 3.4 and complexity analysis in Section 3.5.

For the notations, we denote $\mathcal{X} = \mathbb{R}^d$ as the d-dimensional instance feature space, and denote $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ as the label space with q class labels. Given a PML training set $\mathcal{D} = \{(\boldsymbol{x}_i, C_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})^{\mathrm{T}}$ and $C_i \subset \mathcal{Y}$ denotes the candidate label set, in which only some labels are ground-truth. Furthermore, Let $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ be the instance matrix formed by the training set and $\mathbf{Y} = [Y_{j'i}]_{q\times n}$ as the candidate label matrix, where $Y_{j'i} = 1$ means the j'th label is a candidate label of the ith instance.

3.1. Specific feature extraction based on partial label dependence maximization

Generally speaking, in PML, a main idea is to refine the candidate labels using the information of feature space based on the assumption that similar samples in feature space have similar labels. However, each label may only be associated with a subset of original features. That is, the features have different importance for different labels, which should not be treated equally. The irrelevant features may have negative influences on calculating some information, such as the similarities between instances, which are crucial for label disambiguation in PML. Hence, we attempt to find a set of projection matrices $\{\mathbf{P}_a\}_{a=1}^q$, and $\mathbf{P}_a = [\boldsymbol{p}_a^1, \boldsymbol{p}_a^2, \ldots, \boldsymbol{p}_a^{d'}] \in \mathbb{R}^{d\times d'}$, which can map the original training instances into the label l_a-specific feature space, d' is the dimension of the new feature space, $\mathbf{X}_a = \mathbf{P}_a^{\mathrm{T}}\mathbf{X}$. As shown in Fig. 2, the similarities of two instances may vary from different label-specific feature spaces. After transforming, we can get more credible correlations of instances related to different labels. Specifically, in order to obtain the projection matrix $\mathbf{P}_a$, we try to maximize the dependence between the new feature space and enriched label l_a space, which can be measured effectively by the Hilbert–Schmidt Independence Criterion (HSIC) [32]. Then we adapt the specific label propagation to estimate the confidence of label l_a using the specific features $\mathbf{X}_a$ for disambiguation, which can remove noise features and negative impacts of features associated with different labels, and it will be introduced in Section 3.2.

Here in order to maximize the dependence between the ath projected specific feature space and the label l_a information ($1 \leqslant a \leqslant q$), we adapt HSIC to capture the relevant features associated with label l_a, and an empirical estimate of HSIC is:

$$\mathrm{HSIC}(\mathcal{Q}, \mathcal{F}, \mathcal{D}) = (n-1)^{-2} tr(\mathbf{HK}^{(a)}\mathbf{HL}^{(a)}) \tag{1}$$

where $\mathrm{tr}(\cdot)$ is the trace of a matrix, $\mathbf{H} = [H_{ij}]_{n\times n}$, where $H_{ij} = \delta_{ij} - \frac{1}{n}$ and $\delta_{ij} = 1$ when $i = j$, $\delta_{ij} = 0$ otherwise. $\mathcal{Q}$ and $\mathcal{F}$ are the reproducing kernel Hilbert space (RKHS) mapped from $\mathcal{X}$ and $\mathcal{Y}$. $\mathbf{K}^{(a)}$ and $\mathbf{L}^{(a)}$ are two kernel matrices which need to be defined. Hereafter in this subsection we omit the suffix of label. That is, i.e. we omit a and use $\mathbf{P}$ instead of $\mathbf{P}_a$ to denote the projection matrix of l_a label space to increase the readability. In this paper,

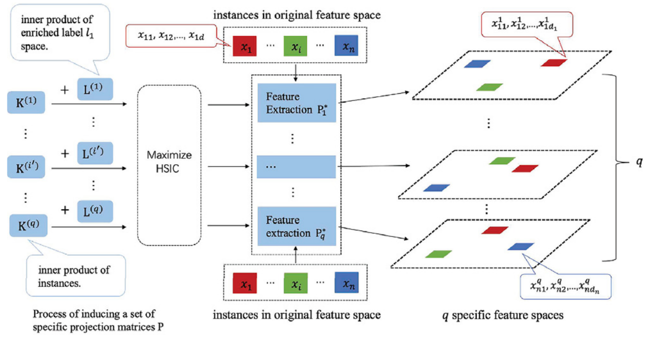

inner product of enriched label l_1 space.
instances in original feature space
$x_{11}, x_{12}, \dots, x_{1d}$
x_1 ··· x_i ··· x_n
$x^1_{11}, x^1_{12}, \dots, x^1_{1d_1}$
$K^{(1)}$ + $L^{(1)}$
$K^{(i')}$ + $L^{(i')}$
$K^{(q)}$ + $L^{(q)}$
Maximize HSIC
Feature Extraction P_1^*
···
Feature extraction P_q^*
q
inner product of instances.
x_1 ··· x_i ··· x_n
$x^q_{n1}, x^q_{n2}, \dots, x^q_{nd_n}$
Process of inducing a set of specific projection matrices P
instances in original feature space
q specific feature spaces

Fig. 2. The process of specific feature extraction. $\mathbf{K}^{(a)}$ and $\mathbf{L}^{(a)}$ are constructed for each label l_a to induce the specific feature projection matrix $\mathbf{P}_a$, which can map the instances into l_a-specific feature space. The instances may have different similarities in different label-specific feature spaces, which are crucial for disambiguation.

we define $\mathbf{K} = [K_{ij}]_{n\times n}$ as the matrix of inner product of instances in $\mathcal{Q}$, and K_{ij} is calculated as follows:

$$K_{ij} = \left\langle \mathbf{P}^{\mathrm{T}}\boldsymbol{x}_i, \mathbf{P}^{\mathrm{T}}\boldsymbol{x}_j \right\rangle \tag{2}$$

Similarly, we define $\mathbf{L} = [L_{ij}]_{n\times n}$ as the inner product of label l_a space, $\mathbf{L} = \langle \mathbf{Y}_{a\cdot}, \mathbf{Y}_{a\cdot} \rangle$. Substituting $\mathbf{K} = \mathbf{X}^{\mathrm{T}}\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{X}$ into Eq. (1), we can get the following objective function:

$$\mathbf{P}^* = \arg\max_{\mathbf{P}} \operatorname{tr}(\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{X}\mathbf{H}\mathbf{L}) \tag{3}$$

To tackle the scaling problem, we limit the projection bases to be orthonormal. Besides, in order to further eliminate the redundant information due to the correlated vectors in the new feature space [36], following [34], we constrain the new feature vectors are also to be orthonormal. The objective function is updated as follows:

$$\begin{aligned} &\max_{\mathbf{P}} \operatorname{tr}(\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{X}\mathbf{H}\mathbf{L}) \\ &s.t. (\boldsymbol{p}^i)^{\mathrm{T}}(\mu\mathbf{X}\mathbf{X}^{\mathrm{T}} + (1-\mu)\mathbf{I})\boldsymbol{p}^j = \delta_{ij} \end{aligned} \tag{4}$$

where $\mu \in [0, 1]$ is used to balance the importance of the two constraints. The solution of Problem (4) will be introduced in Section 3.4.

However, till now, we have only used the information of a single label to extract label-specific features, and the rich information contained in label correlations is ignored. Besides, some labels may have no unique features and need to be classified by other labels. Hence, this paper attempts to use label correlations to enrich each label-specific feature space for better performance. Firstly, we use the label co-occurrence in training data to measure the correlation [37]:

$$\Gamma_{i',j'} = \frac{1}{n}\sum_{t=1}^{n} \mathrm{I}\left(Y_{i't} = 1, Y_{j't} = 1\right) \tag{5}$$

where $\mathrm{I}(\cdot)$ is the indicator function, $\mathrm{I}(\cdot) = 1$ if $\cdot$ is true, $\mathrm{I}(\cdot) = 0$ otherwise.

For each label $l_a \in \mathcal{Y}$, we attempt to find the top r labels $\mathcal{N}_r(l_a)$, which are most related to label l_a (including itself). As different label spaces may have different neighbor spaces, r changes as label space changes. In this paper, we use a threshold ξ to get each r, i.e. $l_b \in \mathcal{N}_r(l_a)$, if $\Gamma_{a,b} \geq \xi$. We use π to represent the indices of the r labels:

$$\Gamma_{a,a} \geq \Gamma_{a,\pi[2]} \geq \cdots \geq \Gamma_{a,\pi[r]} \tag{6}$$

Then, we denote the instance i's ath enriched label space as $\boldsymbol{y}_i^a = (Y_{ai}, Y_{\pi[2]i}, \dots, Y_{\pi[r]i})^{\mathrm{T}}$, and for label l_a, we update its enriched kernel space $\mathbf{L}$ as:

$$L_{ij} = \left\langle \boldsymbol{y}_i^a, \boldsymbol{y}_j^a \right\rangle \tag{7}$$

In this way, we get a set of projection matrices $\{\mathbf{P}_a\}_{a=1}^q$, which can not only reduce the noise and negative impacts between each other but also enrich specific features associated with label l_a space. The process of specific feature extraction is shown as Fig. 2.

3.2. Refine candidate labels with specific label propagation

After the feature extraction, we aim to refine candidate labels using the specific features of each enriched label space. Label propagation is an effective method that has been successfully adapted in many MLL tasks [7,38], which uses the weighted sum of labeling confidence values of neighbors in feature space to update its own for each instance. However, previous works usually estimated the confidence using the original features or the same sub-features for different labels to calculate the weight matrix. It may contain much noise and get a misleading propagation bias, especially in PML. Such as the left of Fig. 3(b), when we use original features to find the 10-nearest neighbors, the label distribution may lead the ground-truth label to be updated false and lose valuable information. So in this paper, we use specific label propagation instead. That is, when we get the projection matrix $\mathbf{P}_a$ for label l_a, we can map the instances into the enriched label l_a-specific feature space. Then the label propagation can be adapted in the new feature space to eliminate the negative influence of irrelevant features, which is more reliable as shown in the right of Fig. 3(b). Besides, as label correlations contain much valuable information, we also consider the interactions between different label spaces for better performance. That is, if label $l_{i'}$ is a neighbor space of $l_{j'}$, then the result of label $l_{i'}$ generated by enriched label $l_{j'}$ space will affect the propagation in enriched label $l_{i'}$ space according to the degree of correlations.

Specifically, Let $\mathcal{D}_a$ be the transformed training set with the ath specific feature space. For each instance $(\mathbf{P}_a^{\mathrm{T}}\boldsymbol{x}_i, Y_{ai}) \in \mathcal{D}_a$, let $\mathcal{N}_k(\boldsymbol{x}_i)$ be the set of $\boldsymbol{x}_i$'s k-nearest neighbors in $\mathcal{D}_a$, and the similarity matrix in label l_a-specific feature space is defined as $\mathbf{S}_a = [S_{ij}^a]_{n\times n}$, and S_{ij}^a is:

$$\forall 1 \leqslant i \leqslant n : S_{ij}^a = \begin{cases} exp(-\dfrac{\left\|\mathbf{P}_a^{\mathrm{T}}\boldsymbol{x}_i - \mathbf{P}_a^{\mathrm{T}}\boldsymbol{x}_j\right\|_2^2}{2\sigma^2}), & \boldsymbol{x}_j \in \mathcal{N}_k(\boldsymbol{x}_i), \\ 0, & \text{otherwise}. \end{cases} \tag{8}$$

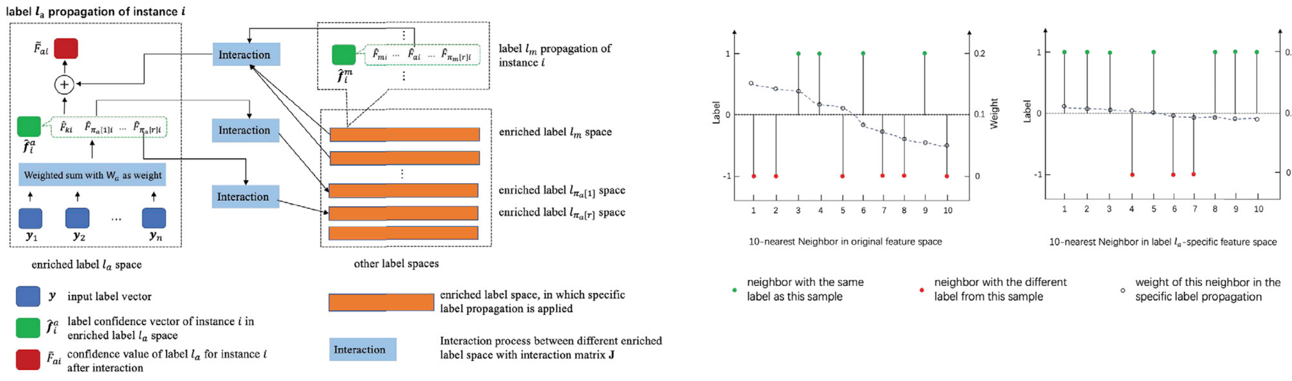
Furthermore, following [7,39], let $\mathbf{W}_a = \mathbf{S}_a\mathbf{D}_a^{-1}$ be the propagation matrix by normalizing the columns of $\mathbf{S}_a$, where $\mathbf{D}_a = \mathrm{diag}[D_1^a, D_2^a, \dots, D_n^a]$ is the diagonal matrix with $D_j^a = \sum_{i=1}^n S_{ij}^a$. Denote the labeling confidence matrix as a non-negative matrix $\mathbf{F} = [F_{i,c}]_{n\times q}$, where $F_{i,c} \geq 0$ represents the confidence of l_c being a ground-truth label for instance $\boldsymbol{x}_i$. Accordingly, the initial labeling confidence matrix $\mathbf{F}^{(0)}$ is configured as:

$$\forall 1 \leqslant i \leqslant n : \quad F_{i,c}^{(0)} = \begin{cases} \dfrac{1}{|C_i|}, & l_c \in C_i, \\ 0, & \text{otherwise}. \end{cases} \tag{9}$$

Thereafter, $\mathbf{F}$ is updated by iterative specific label propagation until convergence. The basic propagation process is as follows:

$$\forall 1 \leqslant a \leqslant q : \quad \hat{\mathbf{F}}_a^{(t)} = \alpha \cdot \mathbf{W}_a\mathbf{F}^{(t-1)} + (1-\alpha)\cdot\mathbf{F}^{(0)} \tag{10}$$

Here, $\alpha \in [0, 1]$ is the balancing parameter which adjusts the relative relationship between the iterative propagation label information and the initial confidence of labels. $\hat{\mathbf{F}}_a = [\hat{F}_{i,c}]_{n\times q}$ is the confidence value matrix directly updated by the weighted sum of labeling confidence of its k-nearest neighbors in enriched l_a space.


label l_a propagation of instance i
$\tilde{F}_{ai}$
$\hat{\boldsymbol{f}}_i^a$
$\hat{F}_{ki}$ $\hat{F}_{\pi_a[1]i}$ ··· $\hat{F}_{\pi_a[r]i}$
Weighted sum with W_a as weight
y_1 y_2 ··· y_n
enriched label l_a space
Interaction
Interaction
Interaction
$\hat{F}_{mi}$ ··· $\hat{F}_{ai}$ ··· $\hat{F}_{\pi_m[r]i}$
$\hat{\boldsymbol{f}}_i^m$
label l_m propagation of instance i
enriched label l_m space
enriched label $l_{\pi_a[1]}$ space
enriched label $l_{\pi_a[r]}$ space
other label spaces
y input label vector
$\hat{\boldsymbol{f}}_i^a$ label confidence vector of instance i in enriched label l_a space
$\tilde{F}_{ai}$ confidence value of label l_a for instance i after interaction
enriched label space, in which specific label propagation is applied
Interaction
Interaction process between different enriched label space with interaction matrix J
Label
Weight
1
0
-1
0.2
0.1
0
1 2 3 4 5 6 7 8 9 10
10-nearest Neighbor in original feature space
10-nearest Neighbor in label l_a-specific feature space
neighbor with the same label as this sample
neighbor with the different label from this sample
weight of this neighbor in the specific label propagation

(a) An example of specific label propagation: instance i's label l_a propagation.

(b) Label distributions of one instance's 10-nearest neighbors.

Fig. 3. Fig. 3(a) is an iteration of the whole specific label l_a propagation of instance i. $\hat{\boldsymbol{f}}_i^a$ is a label confidence vector of instance i in enriched label l_a space, which is obtained by weighted sum of its k-neighbors in l_a-specific feature space. After interacting with other enriched label spaces according to label correlations, the confidence value of l_a for instance i $\tilde{F}_{ai}$ can be obtained; Fig. 3(b) is the label distribution of 10-nearest neighbors of fifth instance in music_emotion data set, where the left is in origin feature space, and the right is in the enriched label-specific feature space of label l_3.

Till now, we only use the enriched label-specific features to do label propagation on each label space while not considering the interactions of label confidence in different label spaces. It is worth noting that in enriched label $l_{i'}$ space, the propagation will update r labeling confidence values, which contains the neighbor labels of $l_{i'}$, and the results generated by different enriched label spaces may be different, which contain valuable information. So for better performance, we add an interactive process for each label during iterations. First, we denote the label correlation matrix $\mathbf{R} = [R_{i'j'}]_{q\times q}$ as follows:

$$\forall 1 \leqslant i' \leqslant q: \quad R_{i'j'} = \begin{cases} \Gamma_{i',j'}, & l_{j'} \in \mathcal{N}_r(l_{i'}), \\ 0, & \text{otherwise}. \end{cases} \tag{11}$$

Similarly, we normalize the label correlation matrix $\mathbf{R}$, which may be asymmetric, by column to yield the interaction matrix $\mathbf{J} = \mathbf{R}\mathbf{D}_r^{-1}$ for different enriched label spaces, where $\mathbf{D}_r = \operatorname{diag}[D_1, D_2, \ldots, D_q]$ and $D_{j'} = \sum_{i'=1}^{q} R_{i'j'}$. Hoping the spaces will interact with each other according to the degree of correlations, the propagation process is updated as follows:

$$\begin{aligned} \forall 1 \leqslant a \leqslant q: \quad & \hat{\mathbf{F}}_a^{(t)} = \alpha \cdot \mathbf{W}_a \mathbf{F}^{(t-1)} + (1-\alpha) \cdot \mathbf{F}^{(0)} \\ \forall 1 \leqslant a \leqslant q: \quad & \tilde{\mathbf{F}}^{(t)} = \tilde{\mathbf{F}}^{(t)} + \mathbf{J}_{a\cdot} \cdot \hat{\mathbf{F}}_a^{(t)} \end{aligned} \tag{12}$$

where $\mathbf{J}_{a\cdot}$ is the interaction weights of enriched label l_a space with other spaces, and $\tilde{\mathbf{F}} = [\tilde{F}_{i,c}]_{n\times q}$ is an intermediate variable whose initial values are all 0. In this way, the label propagation of neighbor space can influence itself, and the whole specific label propagation process is shown as Fig. 3(a). Then the confidence matrix $\tilde{\mathbf{F}}^{(t)}$ will be re-scaled into $\mathbf{F}^{(t)}$ by normalizing each row as follows:

$$\forall 1 \leqslant i \leqslant n: \quad F_{i,c}^{(t)} = \begin{cases} \dfrac{\tilde{F}_{i,c}}{\sum_{l_l \in C_i} \tilde{F}_{i,l}}, & l_c \in C_i, \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

Finally the high-confidence reliable labels can be identified to form the credible label set as follows:

$$C_i' = \left\{ l_c | \mathbf{F}_{ic}^{(t)} \geq thr1, l_c \in C_i \right\} \cup \{ l_c^* | l_c^* = \arg\max_{l_c \in C_i} \mathbf{F}_{ic}^{(t)} \} \tag{14}$$

3.3. Iterative optimization algorithm of PASAD

Due to the noise labels, label correlations in PML are challenging to be explored, leading to the projected features not accurate.

```
Algorithm 1 Training Algorithm of our-method.
Input: the training data $\mathbf{X} \in \mathbb{R}^{d\times n}$
       the training candidate labels $\mathbf{Y} \in \{0,1\}^{q\times n}$
       the regularization parameters $\xi, k, \alpha, \mu, thr1, thr2$
Output: $\mathbf{F}^{(t)}$ and $\{\mathbf{P}_a\}_{a=1}^{q}$
Initialize the $n \times q$ confidence matrix $\mathbf{F}^{(0)}$ according to Eq. (9);
set $t = 0$;
repeat
    Calculate $\mathbf{H} = 1 - \frac{1}{n}\mathbf{e}\mathbf{e}^{\mathrm{T}}$
    for $a = 1$ to $q$ do
        Calculate the kernel matrix for label-specific feature space $\mathbf{K}^{(a)}$ according to Eq. (2) and enriched label space $\mathbf{L}^{(a)}$ according to Eq. (7);
        Maximize HSIC according to Eq. (4) and get $d$ eigenvalues.
        Calculate $d'$ according to Eq. (20)
        Generate the projection matrix $\mathbf{P}_a$ by concatenating the $d'$ eigenvectors corresponding to the top $d'$ eigenvalues;
        Calculate the similarity matrix in label $l_a$ space according to Eq. (8);
        Update $\tilde{\mathbf{F}}^{(t)}$ according to Eq. (12);
    end for
    Re-scale $\tilde{\mathbf{F}}^{(t)}$ into confidence matrix $\mathbf{F}^{(t)}$ according to Eq. (13)
    Refine the candidate label set according to Eq. (14);
    $t = t + 1$;
until convergence
return $\mathbf{F}^{(t)}$ and $\{\mathbf{P}_a\}_{a=1}^{q}$
```

So for better performance, we try to combine the specific feature extraction and refining candidate labels with specific label propagation into an iterative procedure, which can mutually remove the irrelevant features and noise labels.

However, simple combining may significantly reduce time efficiency because the cost of identifying the projection matrices will increase as the dimension of $\mathbf{K}^{(a)}$ increases. To tackle this problem, we use an adaptive feature extraction process without degrading the model's performance instead. When the credible label matrix changes reach a threshold, we identify the projection matrices again and continue to the next iteration. The entire training algorithm of our method is outlined in Algorithm 1.

Let $\mathbf{Y}' = [Y_{ji}']_{q\times n}$ be the label matrix constructed by the refined candidate label set C'. After the training procedure as Algorithm

1, for each class label l_a, we can get the transformed training set with the most relevant features and reliable label as:

$$\mathcal{D}'_a = \{(\mathbf{P}_a^{\mathrm{T}}\boldsymbol{x}_i, Y'_{ai}) | 1 \leq i \leq n\} \tag{15}$$

Based on $\mathcal{D}'_a$, classification model $f_a : \mathbf{X}_a \rightarrow \mathbb{R}$ can be induced by any binary classifier. Moreover, for an unseen instance, we can predict its relevant labels as:

$$C^* = \{l_a | f_a(\mathbf{P}_a^{\mathrm{T}}\boldsymbol{x}^*) > 0, 1 \leq a \leq q\} \tag{16}$$

3.4. Optimization

The Problem (4) can be solved by Lagrange Model effectively. First we have the following deformation:

$$\begin{aligned}
&\mathrm{tr}(\mathbf{HX}^{\mathrm{T}}\mathbf{P}_a\mathbf{P}_a^{\mathrm{T}}\mathbf{XHL}) = \mathrm{tr}(\sum_{i=1}^{d'}\mathbf{HX}^{\mathrm{T}}\boldsymbol{p}_a^i(\boldsymbol{p}_a^i)^{\mathrm{T}}\mathbf{XHL})\\
&= \sum_{i=1}^{d'}\mathrm{tr}(\mathbf{HX}^{\mathrm{T}}\boldsymbol{p}_a^i(\boldsymbol{p}_a^i)^{\mathrm{T}}\mathbf{XHL}) = \sum_{i=1}^{d'}(\boldsymbol{p}_a^i)^{\mathrm{T}}(\mathbf{XHLHX}^{\mathrm{T}})\boldsymbol{p}_a^i \\
&= \mathrm{tr}(\mathbf{P}_a^{\mathrm{T}}\mathbf{XHLHX}^{\mathrm{T}}\mathbf{P}_a)
\end{aligned} \tag{17}$$

Then to solve the Problem (4), the following Lagrange function can be induced:

$$\mathcal{L}(\mathbf{P}_a) = \mathrm{tr}(\mathbf{P}_a^{\mathrm{T}}\mathbf{XHLHX}^{\mathrm{T}}\mathbf{P}_a) + \mathrm{tr}(\Theta(\mathbf{I} - \mathbf{P}_a^{\mathrm{T}}(\mu\mathbf{XX}^{\mathrm{T}} + (1-\mu)\mathbf{I})\mathbf{P}_a)) \tag{18}$$

where the entries of matrix $\Theta = diag(\theta_1, \theta_2, \dots, \theta_{d'})$ are the Lagrange multipliers. By setting the derivative of Problem (18) to 0, we can get the function as follows:

$$\mathbf{XHLHX}^{\mathrm{T}}\boldsymbol{p}_a^i = \lambda_i(\mu\mathbf{XX}^{\mathrm{T}} + (1-\mu)\mathbf{I})\boldsymbol{p}_a^i \tag{19}$$

Then the projection matrix $\mathbf{P}_a = [\boldsymbol{p}_a^1, \boldsymbol{p}_a^2, \dots, \boldsymbol{p}_a^{d'}]$ can be induced where $\boldsymbol{p}_a^i$ is the eigenvector associated the ith eigenvalue of generalized eigenvalue problem (19). As $\mathbf{XHLHX}^{\mathrm{T}}$ is symmetric, its eigenvalues are all real and the eigenvectors are orthogonal to each other. Assuming the eigenvalues are sorted as $\lambda_1 \geqslant \lambda_2 \geqslant \dots \geqslant \lambda_d$, as eigenvalues reflect the importance of corresponding projected features, we choose the eigenvectors associated with the largest d' eigenvalues to form the projection matrix $\mathbf{P}_a$, where d' is calculated by the threshold $thr2$ as follows:

$$\sum_{i=1}^{d'}\lambda_i \geqslant thr2 \times (\sum_{i=1}^{d}\lambda_i) \tag{20}$$

3.5. Complexity analysis

In this part, we analyze the complexity of the proposed method. First, for the specific feature extraction part, the main costs are used to solve the eigenvalues and eigenvectors of the symmetric matrix induced by HSIC. Specifically, the time cost of initializing $\mathbf{H}$ is $O(n^2)$. The time cost of calculating specific label space $\mathbf{L}^{(a)}$ is $O(nq^2 + q^2)$. The time cost of solving the eigenvalues of the symmetric matrix according to Eq. (19) is $O(n^2d + d^3)$. Second, for the specific label propagation part, the time cost of finding the k-nearest neighbors in each feature space is $O(kn^{1-\frac{1}{k}} + ndd' + nkd')$, where KD-Tree is used to optimize the process. The time cost of initializing the confidence matrix is $O(nq)$. The time cost of specific label propagation for each label space is $O(n^2q + nq^2 + nq)$ according to Eq. (12). As we have q label spaces, the complexity of one iteration is $O(n^2q^2 + nq^3 + nq^2)$. The time cost of re-scaling is $O(nq)$. Finally, we combine the specific feature extraction and refine candidate labels with specific label propagation into an iterative procedure. As $n \gg q$, $d \gg k$ and $d \gg d'$, the complexity of the whole disambiguation process is $O(n^2 + t_1 * (nq^2 + n^2d + d^3 + kn^{1-\frac{1}{k}} + ndd') + t_2 * n^2q^2)$, where t_1 is the specific times of re-extracting the specific features, and t_2 is the iteration number of specific label propagation.

Table 1
Characteristics of the experimental data sets.

Datasets	#Instances	#Labels	#Features	Domain	Cardinality
music_emotion	6833	11	98	Music	2.42
music_style	6839	10	98	Music	1.44
mirflickr	10 433	7	100	Image	1.77
emotions	593	6	72	Music	1.869
birds	645	19	260	Audio	1.014
medical	978	45	1449	Text	1.245
enron	1702	53	1001	Text	3.378
scene	2407	6	294	Image	1.07
corel5k	5000	374	499	Image	3.522
bibtex	7395	159	1836	Text	2.402

4. Experiments

4.1. Experimental setting

The experiments are carried out based on Python3.7, intel corel i7-7500U, 8g memory, win10 OS, which are independent of machine configuration. We perform experiments on seven synthetic data sets and three real-world PML data sets in various domains, including music_emotion, music_style, emotions, and birds for music categorization, medical, enron, and bibtex for text classification, mirflickr, scene, and corel5k for image categorization. There are some pre-processing as in [5,40] that rare labels are filtered out of data sets with too many class labels to keep under 15 labels. Then instances with no labels are filtered out. The synthetic data sets can be generated from widely-used MLL data sets by adding $\lfloor \gamma \times |U_i| \rfloor$ noise labels, where $\gamma \in \{50\%, 100\%, 150\%, 200\%\}$ is the percentage of noise, U_i is ground-truth labels of instance i, and $\lfloor \cdot \rfloor$ represents the round down symbol. We perform ten-fold cross validation on the ten data sets. The mean metric value and standard deviation are recorded for comparison with other methods. Table 1 shows the characteristics of the experimental data sets.

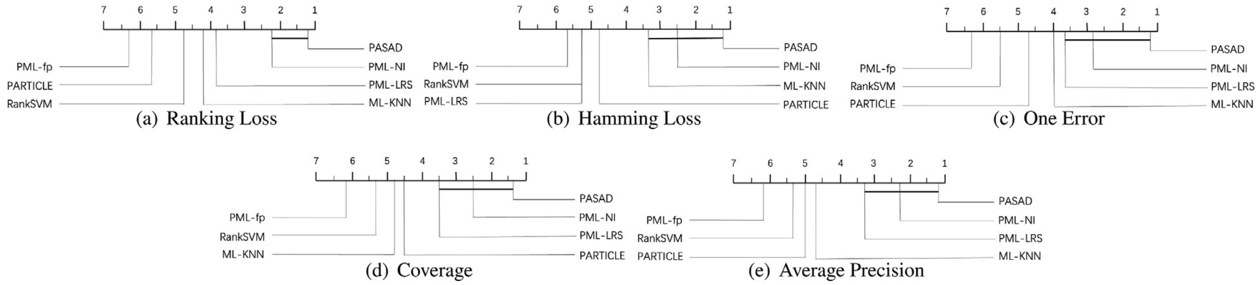
4.2. Evaluation metrics

The performance of multi-label classifier needs to be evaluated from multiple perspectives, not just using accuracy. In order to evaluate the performance of our method, we employ five widely used multi-label metrics, including *ranking loss*, *hamming loss*, *one error*, *coverage* and *average precision*. For *average precision*, the larger the value, the better the performance, and the smaller the value of other metrics, the better the performance. The details of the five criteria can be found in [1].

4.3. Comparison methods

To show the effectiveness of the proposed method, we choose six state-of-art methods for comparison, including four PML methods and two classical MLL methods as follows:

- ML-kNN [16]. The main idea is to use the maximum posterior probability to classify the test sample based on the statistical information of the nearest neighbor. It has been widely used because of its simplicity.
- RankSVM [41]. The basic idea is the maximum margin strategy. A set of linear classifiers can be learned by optimizing the empirical ranking loss. At the same time, non-linear classification problems can be dealt with through kernel tricks.
- PML-fp [5]. It evaluates the confidence value of each label by constraining the relationship of pair-wise labels. Meanwhile, the optimal solution is obtained by minimizing the ranking loss.



(a) Ranking Loss (b) Hamming Loss (c) One Error (d) Coverage (e) Average Precision

Fig. 4. Comparison of ourmethod (control algorithm) against six comparing algorithms with the Bonferroni–Dunn test. Algorithms not connected with ourmethod in the CD diagram are considered to have a significantly different performance from the control algorithm (CD = 2.5486 at 0.05 significance level).

- PML-LRS [11]. It separates ground-truth labels and noise labels using a low-rank and sparse decomposition scheme and simultaneously trains a predict model.
- PARTICLE [7]. It uses the label propagation technique to identify high-confidence labels by aggregating the k-nearest neighbors' information. Then an MLC classifier can be trained by exploiting pair-wise label ranking.
- PML-NI [40]. It constrains the multi-label classifier to be low-rank and the noisy label identifier to be sparse to recover the ground-truth labels and identifies the noisy labels simultaneously.

For the comparing methods, parameters are set as suggested in the original papers, i.e., PML-LRS: $\gamma = 0.01$, $\beta = 0.1$ and $\eta = 1$; PARTICLE: balancing parameter $\alpha = 0.95$ and credible label elicitation threshold $thr = 0.9$; PML-NI: $\lambda = 1$, $\beta = 1$, $\gamma = 0.5$. And according to experimental experience, the balancing parameters of PASAD are set as $\xi = 0.4$, $k = 12$, $\alpha = 0.75$, $\mu = 0.75$, $thr1 = 0.1$ and $thr2 = 0.99$ in this paper and they are fine tuned on some data sets, which will be analyzed in Section 4.8.

4.4. Comparison results

The detailed experimental results are shown as Tables 3 and 4. We can observe that our algorithm shows significant superiority and achieves the best performance in most cases. Especially on the real-world PML data sets, our method achieves the best results in almost all cases except for the data set music emotion where PML-NI outperforms in terms of *hammingloss*.

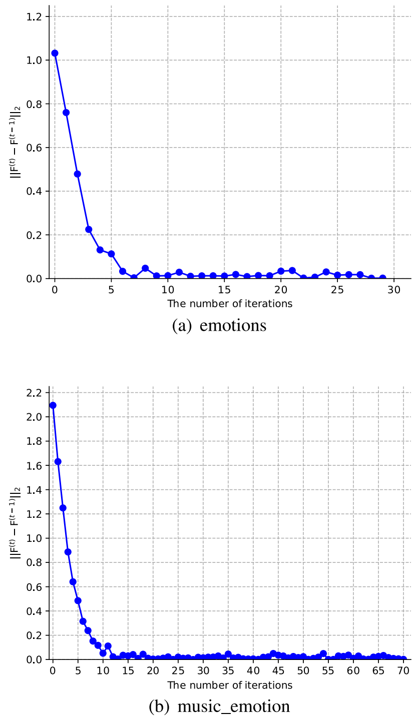
Meanwhile, we employ Friedman test [11,40,42,43] as the statistical test to compare the relative performance of the contrasting approaches. The Friedman statistics F_F and the related critical values are summarized in Table 2. For each evaluation metric, the null hypothesis of indistinguishable performance among the comparing algorithm is rejected at 0.05 significance level. Thereafter, the post-hoc Bonferroni–Dunn test [11,40,42,43] is used to illustrate the relative performance among comparing approaches. The CD (CD = 2.5486 in our experiment: #comparing algorithms $k = 7$, #data sets $N = 10$) diagrams [42] on each evaluation metric are shown in Fig. 4. The average rank of each comparing algorithm is marked along the axis. Algorithms not interconnected with thick lines indicate that their average rank is outside one CD, which means a significant difference.

Overall, based on the above, we can observe the following information:

- Our method ranks 1st in 85.7% cases on the seven synthetic data sets (Table 3) and 93.3% cases on the three real-world PML data sets (Table 4). The latter can better show the effectiveness of PASAD in the real-world scenarios.
- Our method achieves the lowest average rank in terms of all evaluation metrics and no algorithm significantly outperforms our method across all evaluation metrics.
- Our method significantly outperforms RankSVM, PML-fp, PARTICLE in terms of all evaluation metrics. PML-NI also performs well in our experiments and outperforms us on ***bibtex*** in terms of *ranking loss*, *hamming loss* and *coverage*, and ***music_emotion*** in terms of *hamming loss*.

Table 2
Friedman statistics F_F in terms of each evaluation metric and the critical value at 0.05 significance level (#comparing algorithms $k = 7$, #data sets $N = 10$).

Evaluation metric	F_F	Critical value
Ranking Loss	24.7802	2.2720
Hamming loss	12.3559	
One error	16.3012	
Coverage	13.7027	
Average precision	25.6154	



(a) emotions

(b) music_emotion

Fig. 5. The convergence curves of PASAD on emotions and music_emotion data sets with increasing number of iterations.

These experimental results convincingly validate the significance of the superiority for the proposed PASAD approach.

4.5. Convergence analysis

In this subsection, We conduct the convergence analysis of Algorithm 1 on emotions with 100% label noise and music_emotion

Table 3
Comparison of ourmethod with state-of-the-art MLL and PML methods on five evaluation metrics on synthetic PML datasets, where the best performances are shown in bold.

Datasets	ML-KNN	RankSVM	PML-fp	PML-LRS	PML-NI	PARTICLE	PASAD
Ranking loss (the smaller, the better)							
emotions	.183 ± .030	.222 ± .048	.308 ± .032	.185 ± .039	.216 ± .011	.261 ± .031	**.162 ± .020**
birds	.206 ± .028	.225 ± .027	.390 ± .034	.226 ± .034	.187 ± .033	.306 ± .031	**.171 ± .028**
medical	.071 ± .004	.050 ± .014	.115 ± .027	.027 ± .009	.023 ± .002	.106 ± .022	**.017 ± .003**
enron	.183 ± .004	.279 ± .024	.328 ± .013	.279 ± .011	.177 ± .017	.201 ± .047	**.118 ± .004**
scene	.121 ± .018	.314 ± .261	.197 ± .013	.165 ± .019	.129 ± .007	.190 ± .047	**.102 ± .009**
corel5k	.287 ± .007	.327 ± .032	.362 ± .009	.285 ± .007	.240 ± .009	.327 ± .040	**.238 ± .011**
bibtex	.159 ± .015	.119 ± .024	.195 ± .010	.128 ± .005	**.055 ± .004**	.133 ± .017	.057 ± .005
Hamming loss (the smaller, the better)							
emotions	.217 ± .015	.289 ± .017	.318 ± .024	.219 ± .017	.236 ± .008	.279 ± .030	**.208 ± .011**
birds	.089 ± .011	.143 ± .012	.168 ± .026	.143 ± .011	.098 ± .009	.149 ± .010	**.087 ± .005**
medical	.040 ± .003	.073 ± .009	.107 ± .005	.047 ± .012	.028 ± .002	.076 ± .008	**.021 ± .001**
enron	.151 ± .003	.218 ± .010	.256 ± .011	.278 ± .009	.162 ± .005	.174 ± .035	**.122 ± .005**
scene	.120 ± .005	.243 ± .114	.256 ± .011	.168 ± .010	.126 ± .007	.146 ± .055	**.115 ± .004**
corel5k	.114 ± .002	.184 ± .003	.225 ± .008	.126 ± .003	.117 ± .003	.189 ± .011	**.113 ± .002**
bibtex	.056 ± .002	.096 ± .006	.129 ± .005	.135 ± .004	**.038 ± .002**	.087 ± .008	.040 ± .002
One error (the smaller, the better)							
emotions	.281 ± .066	.375 ± .068	.378 ± .016	.287 ± .092	.346 ± .044	.383 ± .059	**.261 ± .060**
birds	.522 ± .079	.572 ± .128	.789 ± .025	.462 ± .093	.436 ± .045	.673 ± .029	**.380 ± .068**
medical	.233 ± .017	.250 ± .047	.254 ± .035	.201 ± .042	.133 ± .028	.192 ± .014	**.101 ± .019**
enron	.309 ± .023	.562 ± .103	.569 ± .021	.468 ± .025	.317 ± .032	.371 ± .108	**.226 ± .019**
scene	**.260 ± .025**	.565 ± .324	.450 ± .021	.379 ± .020	.324 ± .016	.344 ± .015	.279 ± .021
corel5k	.712 ± .013	.732 ± .032	.744 ± .011	.643 ± .016	.623 ± .016	.646 ± .031	**.607 ± .010**
bibtex	.393 ± .032	.383 ± .100	.491 ± .013	.383 ± .014	.179 ± .014	.444 ± .031	**.159 ± .013**
Coverage (the smaller, the better)							
emotions	.323 ± .020	.353 ± .041	.337 ± .018	.306 ± .022	.345 ± .024	.342 ± .014	**.299 ± .007**
birds	.309 ± .049	.324 ± .023	.415 ± .054	**.280 ± .009**	.284 ± .039	.341 ± .005	.306 ± .038
medical	.156 ± .005	.132 ± .015	.177 ± .022	.106 ± .008	.101 ± .005	.126 ± .024	**.098 ± .006**
enron	.411 ± .006	.481 ± .021	.527 ± .025	.498 ± .022	.391 ± .029	.362 ± .098	**.325 ± .014**
scene	.117 ± .015	.276 ± .215	.180 ± .025	.154 ± .016	.124 ± .006	.176 ± .037	**.101 ± .007**
corel5k	.378 ± .010	.415 ± .031	.447 ± .016	.379 ± .008	.327 ± .013	.499 ± .009	**.322 ± .014**
bibtex	.248 ± .015	.214 ± .025	.289 ± .006	.222 ± .005	**.144 ± .003**	.162 ± .006	.152 ± .003
Average precision (the higher, the better)							
emotions	.782 ± .032	.736 ± .046	.696 ± .023	.788 ± .047	.753 ± .023	.707 ± .032	**.804 ± .029**
birds	.537 ± .041	.523 ± .025	.320 ± .047	.582 ± .059	.602 ± .031	.411 ± .038	**.610 ± .042**
medical	.828 ± .013	.831 ± .037	.826 ± .025	.912 ± .028	.919 ± .014	.849 ± .013	**.939 ± .009**
enron	.679 ± .014	.535 ± .040	.492 ± .017	.566 ± .010	.682 ± .026	.632 ± .052	**.769 ± .014**
scene	.812 ± .020	.609 ± .260	.715 ± .017	.757 ± .016	.797 ± .009	.761 ± .066	**.829 ± .014**
corel5k	.433 ± .009	.409 ± .028	.388 ± .021	.471 ± .007	.503 ± .010	.447 ± .016	**.506 ± .015**
bibtex	.681 ± .026	.713 ± .058	.684 ± .008	.708 ± .008	.859 ± .007	.689 ± .030	**.870 ± .006**

Table 4
Comparison of ourmethod with state-of-the-art MLL and PML methods on real-world PML datasets, where the best performances are shown in bold.

Datasets	ML-KNN	RankSVM	PML-fp	PML-LRS	PML-NI	PARTICLE	PASAD
Ranking loss (the smaller, the better)							
music_emotion	.304 ± .009	.261 ± .009	.370 ± .012	.256 ± .002	.253 ± .005	.324 ± .013	**.250 ± .006**
music_style	.198 ± .005	.224 ± .007	.174 ± .011	.148 ± .006	.141 ± .007	.222 ± .009	**.131 ± .005**
mirflickr	.184 ± .011	.129 ± .011	.143 ± .028	.127 ± .004	.122 ± .003	.144 ± .137	**.095 ± .016**
Hamming loss (the smaller, the better)							
music_emotion	.358 ± .012	.391 ± .006	.226 ± .005	.328 ± .004	**.212 ± .003**	2̇37 ± .007	.217 ± .005
music_style	.847 ± .002	.637 ± .006	.162 ± .006	.694 ± .006	.115 ± .002	.125 ± .002	**.115 ± .001**
mirflickr	.220 ± .004	.218 ± .002	.202 ± .057	.223 ± .003	.166 ± .004	.210 ± .093	**.165 ± .002**
One error (the smaller, the better)							
music_emotion	.545 ± .008	.516 ± .024	.593 ± .010	.481 ± .006	.499 ± .021	.563 ± .005	**.453 ± .009**
music_style	.380 ± .009	.573 ± .019	.437 ± .012	.345 ± .018	.354 ± .013	.405 ± .005	**.324 ± .012**
mirflickr	.447 ± .131	.345 ± .058	.298 ± .121	.305 ± .019	.299 ± .016	.247 ± .289	**.198 ± .073**
Coverage (the smaller, the better)							
music_emotion	.468 ± .008	.423 ± .008	.420 ± .007	.417 ± .004	.413 ± .005	.413 ± .008	**.412 ± .005**
music_style	.261 ± .004	.274 ± .011	.298 ± .012	.205 ± .010	.199 ± .008	.205 ± .013	**.192 ± .007**
mirflickr	.271 ± .002	.232 ± .006	.233 ± .074	.231 ± .005	.228 ± .003	.250 ± .062	**.205 ± .010**
Average precision (the higher, the better)							
music_emotion	.555 ± .006	.582 ± .010	.562 ± .012	.589 ± .006	.597 ± .008	.509 ± .009	**.624 ± .017**
music_style	.686 ± .006	.575 ± .011	.665 ± .011	.714 ± .008	.731 ± .007	.656 ± .002	**.752 ± .009**
mirflickr	.693 ± .005	.776 ± .022	.758 ± .039	.786 ± .006	.790 ± .007	.786 ± .191	**.851 ± .028**

Table 5
Comparison results of ourmethod and its ablation variants in terms of ranking loss, hamming loss and average precision.

Datasets	PASAD	PASAD/S-E	PASAD/E-P	PASAD/E	PASAD/P	PASAD/all
Ranking loss (the smaller, the better)						
music_emotion	**.250 ± .006**	.283 ± .019	.261 ± .014	.260 ± .003	.276 ± .011	.271 ± .005
music_style	**.131 ± .005**	.173 ± .015	.187 ± .022	.155 ± .006	.211 ± .005	.202 ± .007
mirflickr	**.095 ± .016**	.121 ± .016	.127 ± .017	.129 ± .005	.128 ± .010	.141 ± .015
emotions	**.162 ± .020**	.187 ± .020	.196 ± .023	.190 ± .027	.177 ± .036	.195 ± .022
birds	**.171 ± .028**	.216 ± .011	.206 ± .025	.196 ± .028	.184 ± .022	.211 ± .025
medical	**.017 ± .003**	.038 ± .010	.068 ± .005	.023 ± .006	.027 ± .005	.069 ± .005
enron	**.118 ± .004**	.177 ± .009	.168 ± .018	.126 ± .015	.120 ± .008	.173 ± .006
scene	**.102 ± .009**	.112 ± .007	.140 ± .010	.109 ± .009	.115 ± .011	.154 ± .009
corel5k	**.238 ± .011**	.306 ± .012	.299 ± .012	.256 ± .015	.251 ± .008	.271 ± .011
bibtex	**.057 ± .005**	.106 ± .005	.119 ± .014	.064 ± .006	.058 ± .003	.121 ± .006
Hamming loss (the smaller, the better)						
music_emotion	**.217 ± .005**	.235 ± .003	.336 ± .007	.240 ± .004	.339 ± .005	.339 ± .008
music_style	**.115 ± .001**	.126 ± .003	.759 ± .038	.145 ± .005	.839 ± .008	.838 ± .006
mirflickr	**.165 ± .002**	.174 ± .003	.223 ± .002	.173 ± .004	.217 ± .002	.225 ± .002
emotions	**.208 ± .011**	.211 ± .014	.230 ± .019	.238 ± .024	.235 ± .014	.222 ± .018
birds	**.087 ± .005**	.088 ± .008	.093 ± .016	.090 ± .007	.089 ± .005	.093 ± .004
medical	**.021 ± .001**	.036 ± .003	.047 ± .010	.030 ± .002	.031 ± .028	.053 ± .003
enron	**.122 ± .005**	.163 ± .003	.145 ± .013	.130 ± .007	.137 ± .005	.151 ± .002
scene	**.115 ± .004**	.118 ± .003	.138 ± .007	.122 ± .002	.134 ± .007	.136 ± .004
corel5k	**.113 ± .002**	.115 ± .002	.123 ± .008	.119 ± .005	.120 ± .001	.121 ± .002
bibtex	**.040 ± .002**	.052 ± .001	.050 ± .003	.043 ± .001	.045 ± .002	.057 ± .002
Average precision (the higher, the better)						
music_emotion	**.624 ± .017**	.575 ± .016	.580 ± .014	.588 ± .006	.570 ± .009	.571 ± .007
music_style	**.752 ± .009**	.706 ± .010	.675 ± .015	.722 ± .010	.661 ± .005	.668 ± .011
mirflickr	**.851 ± .028**	.834 ± .008	.789 ± .002	.843 ± .008	.780 ± .005	.761 ± .007
emotions	**.804 ± .029**	.785 ± .033	.774 ± .018	.774 ± .034	.788 ± .029	.773 ± .029
birds	**.610 ± .042**	.570 ± .013	.568 ± .044	.597 ± .038	.598 ± .024	.560 ± .033
medical	**.939 ± .009**	.881 ± .031	.807 ± .013	.916 ± .015	.899 ± .010	.798 ± .009
enron	**.769 ± .014**	.678 ± .016	.722 ± .013	.754 ± .025	.751 ± .015	.684 ± .009
scene	**.829 ± .014**	.814 ± .009	.794 ± .011	.823 ± .008	.820 ± .013	.773 ± .005
corel5k	**.506 ± .015**	.430 ± .011	.466 ± .010	.495 ± .026	.496 ± .011	.469 ± .012
bibtex	**.870 ± .006**	.760 ± .012	.741 ± .021	.854 ± .007	.861 ± .011	.729 ± .014

data sets. We fix $\xi = 0.4$, $\alpha = 0.75$, $\mu = 0.75$, $thr1 = 0.1$, $thr2 = 0.99$ and fix the number of nearest neighbors k to 10. The results are shown in Fig. 5. It can be easily observed that each $\|\mathbf{F}^{(t)} - \mathbf{F}^{(t-1)}\|_2$ gradually decreases to 0 as the number of iterations t increases to about 30 and 70, which can validate the efficiency of our method.

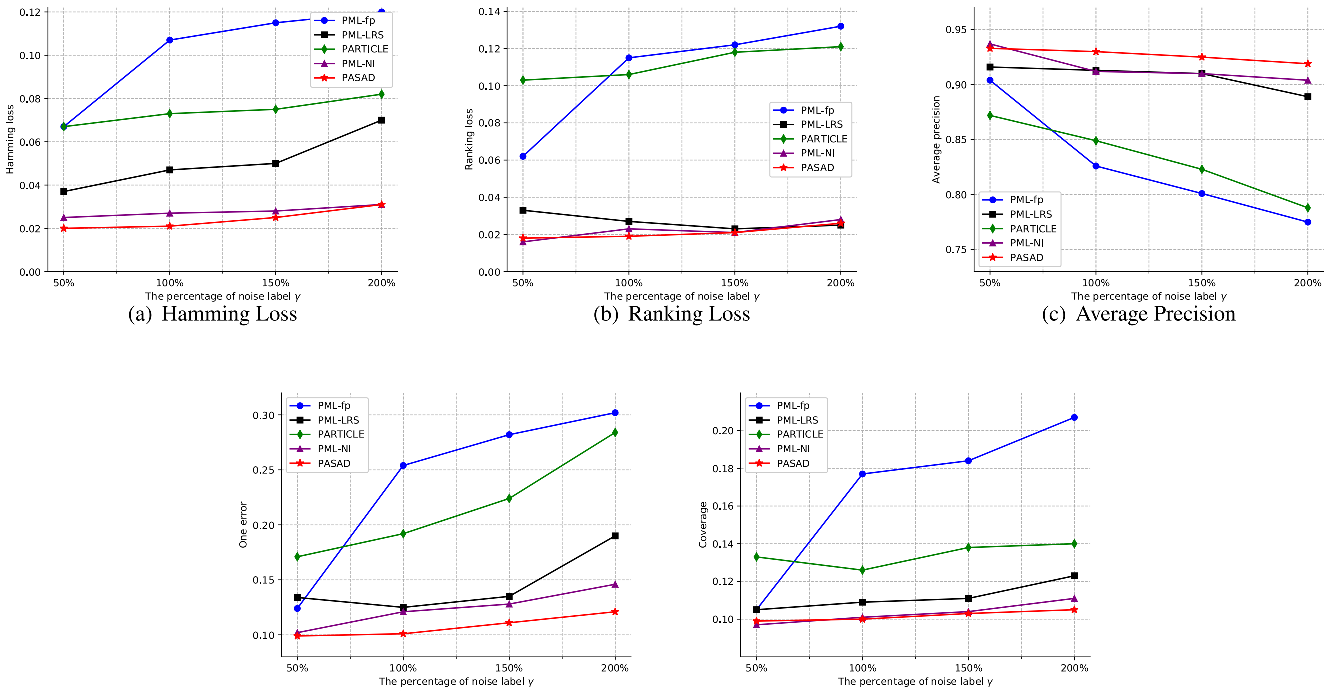
4.6. Ablation study

The proposed PASAD has three components which contribute to the learning, including specific feature extraction, enriching label-specific features and specific label propagation. To demonstrate the efficiency of each component, a careful ablation study is conducted. As enriching label-specific features is based on specific feature extraction, there are $2^3 - 2$ possible combinations, and the ablation variants are named as follows:

- **PASAD/S-E**: The PASAD model without the specific feature extraction as well as enriching the specific features.
- **PASAD/E-P**: The PASAD model without enriching the specific features as well as specific label propagation.
- **PASAD/E**: The PASAD model without enriching the specific features by using label correlations.
- **PASAD/P**: The PASAD model without specific label propagation.
- **PASAD/all**: The model with only using the binary classifier.

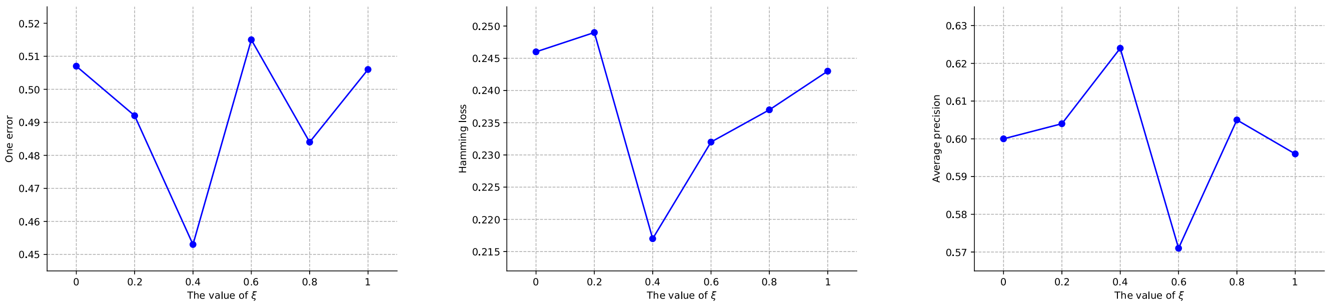
We compare the performance of PASAD/S-E, PASAD/E-P, PASAD/E, PASAD/P, PASAD/all and full model on all synthetic data sets with 100% label noise and three real-world PML data sets in terms of five evaluation metrics. The comparison results are shown in Tables 5 and 6. From the results, we can observe that: (1) The full model achieves the best results in all test data sets and the four variants produced inferior results. (2) For the variants, we can observe that the PASAD/S-E and PASAD/E produce a significant performance degradation. As without the specific feature extraction, the label propagation may be biased and get a worse result than the PASAD/all in some cases. Meanwhile, without considering the label correlations, the specific feature space of each label may ignore some crucial information, as some labels may have no specific features and need label correlation for classification. (3) The specific label propagation also affects the performance of the model, which is used for disambiguation. Due to the noise labels, specific feature extraction and label correlations may be inaccurate and get a worse result. Specific feature extraction is for specific label propagation to be more accurate and prevent deviation. The two are mutually constrained to obtain a reliable data set with less noise of both features and labels.

So in this subsection, we can get the following conclusions from the results: (1) The label-specific feature extraction considering label correlations is effective for PML, which is based on partial label dependence maximization. It can reduce the irrelevant features and negative impacts of features associated with different labels. (2) Specific label propagation can also improve performance by refining the candidate labels. That is, specific label propagation with enriched label-specific features can reduce the bias and get a more credible label set. With the new training data which has been disambiguated detailedly, PML can be transformed into a set of binary classification problems. In all, the ablation experiments confirm the rationality of our model, and each component helps improve the model performance.



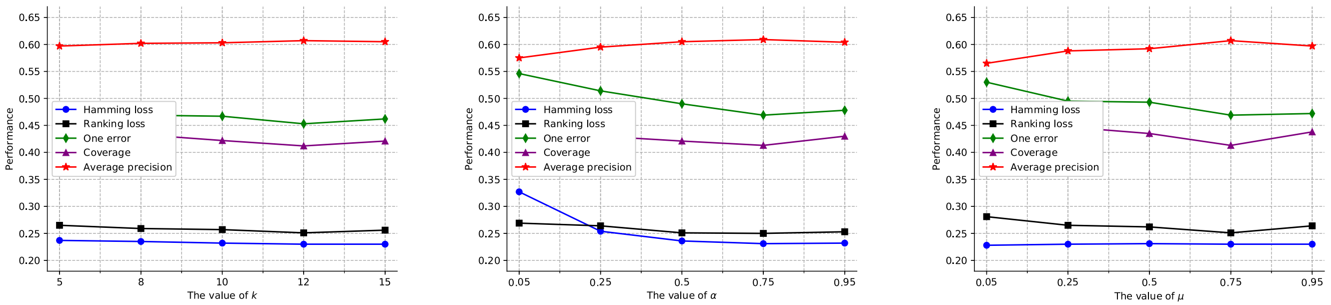
(a) Hamming Loss
(b) Ranking Loss
(c) Average Precision
(d) One Error
(e) Coverage

Fig. 6. Variations of OneError, Coverage, HammingLoss, RankingLoss, and Average-Precision with increasing the percentage of the noise label on medical data set.



(a) One Error
(b) Hamming Loss
(c) Average Precision

Fig. 7. Variations of OneError, HammingLoss, and AveragePrecision with increasing the value of ξ on music_emotion set.



(a) The performance of varying k
(b) The performance of varying α
(c) The performance of varying μ

Fig. 8. The performance of our method changes as each parameter increases with other parameters fixed on music_emotion data set.

Table 6
Comparison results of ourmethod and its ablation variants in terms of one error, coverage.

Datasets	PASAD	PASAD/S-E	PASAD/E-P	PASAD/E	PASAD/P	PASAD/all
One error (the smaller, the better)						
music_emotion	**.453 ± .009**	.504 ± .018	.534 ± .031	.507 ± .025	.555 ± .025	.552 ± .010
music_style	**.324 ± .012**	.371 ± .005	.388 ± .025	.368 ± .016	.405 ± .009	.405 ± .022
mirflickr	**.198 ± .073**	.256 ± .015	.290 ± .048	.248 ± .012	.305 ± .080	.334 ± .073
emotions	**.261 ± .060**	.269 ± .072	.287 ± .050	.297 ± .058	.286 ± .037	.303 ± .051
birds	**.380 ± .068**	.441 ± .028	.486 ± .064	.425 ± .057	.434 ± .035	.487 ± .050
medical	**.101 ± .019**	.168 ± .048	.261 ± .025	.130 ± .021	.154 ± .018	.286 ± .020
enron	**.226 ± .019**	.320 ± .032	.284 ± .027	.235 ± .038	.261 ± .027	.307 ± .009
scene	**.279 ± .021**	.303 ± .012	.323 ± .015	.290 ± .011	.289 ± .019	.355 ± .013
corel5k	**.607 ± .010**	.694 ± .011	.656 ± .017	.626 ± .035	.620 ± .013	.653 ± .020
bibtex	**.159 ± .013**	.285 ± .015	.324 ± .027	.175 ± .011	.171 ± .015	.339 ± .017
Coverage (the smaller, the better)						
music_emotion	**.412 ± .005**	.575 ± .029	.524 ± .013	.527 ± .008	.525 ± .009	.521 ± .004
music_style	**.192 ± .007**	.339 ± .017	.377 ± .041	.327 ± .004	.374 ± .005	.364 ± .006
mirflickr	**.205 ± .010**	.371 ± .005	.370 ± .004	.335 ± .002	.369 ± .002	.381 ± .004
emotions	**.299 ± .007**	.332 ± .016	.324 ± .018	.326 ± .032	.314 ± .030	.329 ± .012
birds	**.306 ± .038**	.391 ± .019	.344 ± .040	.358 ± .038	.337 ± .037	.363 ± .025
medical	**.098 ± .006**	.123 ± .011	.147 ± .005	.105 ± .005	.108 ± .007	.151 ± .007
enron	**.325 ± .014**	.465 ± .013	.440 ± .026	.396 ± .023	.379 ± .010	.460 ± .008
scene	**.101 ± .007**	.275 ± .006	.293 ± .010	.272 ± .008	.278 ± .010	.309 ± .006
corel5k	**.322 ± .014**	.403 ± .011	.342 ± .013	.346 ± .010	.338 ± .005	.364 ± .012
bibtex	**.152 ± .003**	.202 ± .006	.199 ± .006	.158 ± .008	.156 ± .006	.214 ± .008

4.7. Impact of false positive labels

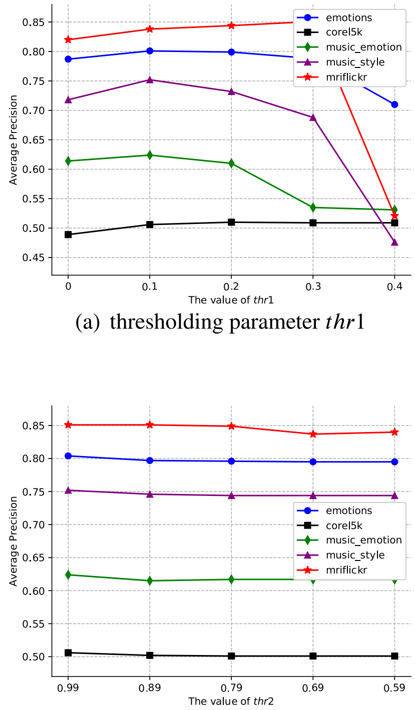
In this subsection, we conduct an experiment on medical data set to evaluate the performance of our method with different percentage of noise labels in terms of five evaluation metrics. The percentage of noise labels is tuned from {50%, 100%, 150%, 200%}. We compare PASAD with other PML methods and the comparison results are shown as Fig. 6. From Fig. 6, we can see that, with increasing the percentage of false positive labels, the performance of all the comparison methods degrades or fluctuates greatly. However, ourmethod still shows superior performance compared with other methods and is robust against false labels.

4.8. Parameter sensitivity

In this subsection, we study the influence of four trade-off hyper-parameters ξ, k, α, μ, $thr1$, $thr2$ on the model's performance. We conduct experiments by varying one parameter while keeping the other three parameters fixed. The experimental results are shown as Figs. 7–9. From Fig. 7, we can observe ξ will influence the performance on the music_emotion data sets. As ξ increases, the label-specific feature space will be enriched. However, the performance will drop when ξ is too large, as there may be more irrelevant features, which can also validate the effectiveness of specific feature extraction. Note that, the curve oscillation is because that we use the threshold ξ to control the number r_a of neighbor space related to label l_a, which is more flexible. From Fig. 8, we can observe that performance is not sensitive to the parameter k, and the impact of α is roughly the same as the impact of μ. From Fig. 9, we can observe that the performance of PASAD becomes relatively stable as $thr1$ increase to 0.1, which is the value used in this paper. PASAD is not sensitive to $thr2$ and we choose $thr2 = 0.99$ in this paper. Obviously, we can notice better performance is gained when $\xi = 0.4$, $k = 12$, $\alpha = 0.75$, $\mu = 0.75$, $thr1 = 0.1$ and $thr2 = 0.99$ on the music_emotion data set.

4.9. Time consumption analysis

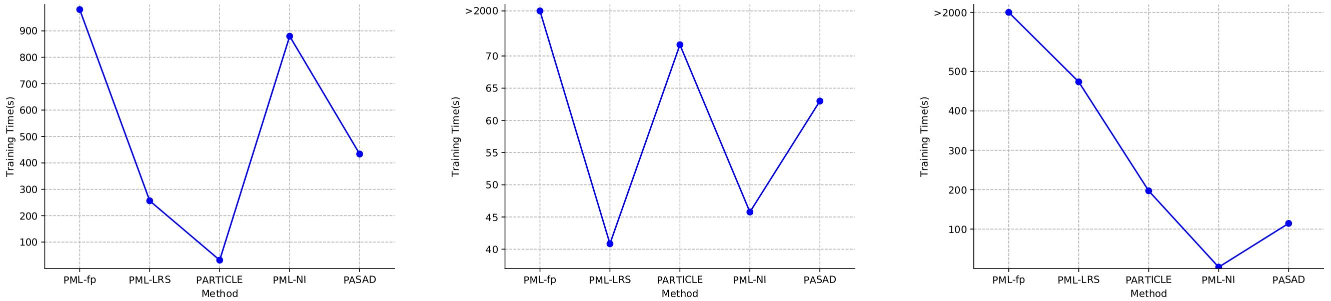
In this subsection, we conduct an experiment on medical with higher feature dimension, corel5k with more instances and labels and real-world data set music_style to compare the time consumption of our method with other methods'. The experimental results are shown as Fig. 10. From Fig. 10, we can observe that the time consumption of our method is in the acceptable range compared with other PML methods'. It is mainly related to n, d, q which has been analyzed in Section 3.5. This is a trade-off between time consumption and performance.



(a) thresholding parameter $thr1$

(b) thresholding parameter $thr2$

Fig. 9. The performance of our method changes as $thr1$, $thr2$ changes with other parameters fixed (in terms of average precision).



(a) medical (b) corel5k (c) music_style

Fig. 10. The time consumptions of PASAD and comparison methods on medical, corel5k and music_style data sets.

5. Conclusion

In this paper, we propose a novel method PASAD for partial multi-label learning. Specifically, for each label l_a space, the proposed PASAD tries to find a projection matrix by maximizing the dependence between the projected feature information and enriched label l_a space constructed using label correlations. Thereafter, we adapt specific label propagation using the label-specific features of each enriched label space for disambiguation, removing irrelevant features' biased influence for each label. We also consider the interactions of label confidence between different enriched label spaces during the propagation. Then, by combining the above two stages into an iterative process, label-specific feature extraction and specific label propagation can restrict each other for better performance in an iterative manner. Finally, PML can be transformed into a set of binary classification problems. Extensive experimental results on synthesized and real-world PML data sets demonstrate the effectiveness of the proposed method, especially in real-world PML data sets. We also conduct ablation experiments on the model to validate each component's effectiveness and demonstrate the model's rationality.

CRediT authorship contribution statement

Feng Li: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Shengfei Shi:** Writing – review & editing, Supervision. **Hongzhi Wang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Thanks Massive Data Computing Lab at School of Computer Science and Technology, Harbin Institute of Technology for providing the platform in the experiments.

Funding

This paper was partially supported by NSFC, China grant (No. U1866602) and Sichuan Science and Technology Program, China (No. 2021YFSY0014).

References

[1] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2013) 1819–1837.

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

[3] Z. He, J. Wu, P. Lv, Multi-label text classification based on the label correlation mixture model, Intell. Data Anal. 21 (6) (2017) 1371–1392.

[4] C. Sanden, J.Z. Zhang, Enhancing multi-label music genre classification through ensemble techniques, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2011, pp. 705–714.

[5] M.-K. Xie, S.-J. Huang, Partial multi-label learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[6] W. Liu, H. Wang, X. Shen, I. Tsang, The emerging trends of multi-label learning, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[7] J.-P. Fang, M.-L. Zhang, Partial multi-label learning via credible label elicitation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3518–3525.

[8] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, G. Chen, Discriminative and correlative partial multi-label learning, in: IJCAI, 2019, pp. 3691–3697.

[9] N. Xu, Y.-P. Liu, X. Geng, Partial multi-label learning with label distribution, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 6510–6517.

[10] N. Xu, Y.-P. Liu, Y. Zhang, X. Geng, Progressive enhancement of label distributions for partial multilabel learning, IEEE Trans. Neural Netw. Learn. Syst. (2021).

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

[12] Z. Li, G. Lyu, S. Feng, Partial multi-label learning via multi-subspace representation, in: Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence IJCAI-PRICAI-20, 2020, pp. 2612–2618.

[13] M.-K. Xie, F. Sun, S.-J. Huang, Partial multi-label learning with meta disambiguation, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1904–1912.

[14] T. Cour, B. Sapp, B. Taskar, Learning from partial labels, J. Mach. Learn. Res. 12 (2011) 1501–1536.

[15] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.

[16] M.-L. Zhang, Z.-H. Zhou, ML-KNN: A Lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

[17] J. Fürnkranz, E. Hüllermeier, E.L. Mencía, K. Brinker, Multilabel classification via calibrated label ranking, Mach. Learn. 73 (2) (2008) 133–153.

[18] S. He, L. Feng, L. Li, Estimating latent relative labeling importances for multi-label learning, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 1013–1018.

[19] Y. Li, Y. Song, J. Luo, Improving pairwise ranking for multi-label image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3617–3625.

[20] M.-L. Zhang, Z.-H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351.

[21] S. Burkhardt, S. Kramer, Online multi-label dependency topic models for text classification, Mach. Learn. 107 (5) (2018) 859–886.

[22] L. Jing, L. Yang, J. Yu, M.K. Ng, Semi-supervised low-rank mapping learning for multi-label classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1483–1491.

[23] J. Read, B. Pfahringer, G. Holmes, Multi-label classification using ensembles of pruned sets, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 995–1000.

[24] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (3) (2011) 333–359.

[25] Y. Grandvalet, Y. Bengio, et al., Learning from Partial Labels with Minimum Entropy, Technical Report, CIRANO, 2004.

[26] M.-L. Zhang, F. Yu, C.-Z. Tang, Disambiguation-free partial label learning, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2155–2167.

[27] E. Hüllermeier, J. Beringer, Learning from ambiguously labeled examples, Intell. Data Anal. 10 (5) (2006) 419–439.

[28] M.-L. Zhang, F. Yu, Solving the partial label learning problem: An instance-based approach, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[29] R. Jin, Z. Ghahramani, Learning with multiple labels, in: NIPS, Vol. 2, Citeseer, 2002, pp. 897–904.

[30] L. Liu, T.G. Dietterich, A conditional multinomial mixture model for superset label learning, in: Advances in Neural Information Processing Systems, Citeseer, 2012, pp. 548–556.

[31] F. Yu, M.-L. Zhang, Maximum margin partial label learning, in: Asian Conference on Machine Learning, PMLR, 2016, pp. 96–111.

[32] A. Gretton, O. Bousquet, A. Smola, B. Schölkopf, Measuring statistical dependence with Hilbert-Schmidt norms, in: International Conference on Algorithmic Learning Theory, Springer, 2005, pp. 63–77.

[33] M.J. Gangeh, H. Zarkoob, A. Ghodsi, Fast and scalable feature selection for gene expression data using hilbert-schmidt independence criterion, IEEE/ACM Trans. Comput. Biol. Bioinform. 14 (1) (2017) 167–181.

[34] W.-X. Bao, J.-Y. Hang, M.-L. Zhang, Partial label dimensionality reduction via confidence-based dependence maximization, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 46–54.

[35] W. Zhu, W. Li, X. Jia, Multi-label learning with local similarity of samples, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.

[36] J. Chen, S. Ji, B. Ceran, Q. Li, M. Wu, J. Ye, Learning subspace kernels for classification, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 106–114.

[37] G. Lyu, S. Feng, Y. Li, Partial multi-label learning via probabilistic graph matching mechanism, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 105–113.

[38] G. Lyu, S. Feng, W. Huang, G. Dai, H. Zhang, B. Chen, Partial label learning via low-rank representation and label propagation, Soft Comput. 24 (7) (2020) 5165–5176.

[39] W. Wang, M. Zhang, Semi-supervised partial label learning via confidence-rated margin maximization, in: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual, 2020, URL: https://proceedings.neurips.cc/paper/2020/hash/4dea382d82666332fb564f2e711cbc71-Abstract.html.

[40] M.-K. Xie, S.-J. Huang, Partial multi-label learning with noisy label identification, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[41] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, Adv. Neural Inf. Process. Syst. 14 (2001) 681–687.

[42] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[43] Q.-W. Zhang, Y. Zhong, M.-L. Zhang, Feature-induced labeling information enrichment for multi-label learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.