This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Hybrid Noise-Oriented Multilabel Learning

Changqing Zhang, *Member, IEEE*, Ziwei Yu, Huazhu Fu, *Senior Member, IEEE*, Pengfei Zhu, *Member, IEEE*, Lei Chen, and Qinghua Hu, *Senior Member, IEEE*

***Abstract*—For real-world applications, multilabel learning usually suffers from unsatisfactory training data. Typically, features may be corrupted or class labels may be noisy or both. Ignoring noise in the learning process tends to result in an unreasonable model and, thus, inaccurate prediction. Most existing methods only consider either feature noise or label noise in multilabel learning. In this paper, we propose a unified robust multilabel learning framework for data with hybrid noise, that is, both feature noise and label noise. The proposed method, hybrid noise-oriented multilabel learning (HNOML), is simple but rather robust for noisy data. HNOML simultaneously addresses feature and label noise by bi-sparsity regularization bridged with label enrichment. Specifically, the label enrichment matrix explores the underlying correlation among different classes which improves the noisy labeling. Bridged with the enriching label matrix, the structured sparsity is imposed to jointly handle the corrupted features and noisy labeling. We utilize the alternating direction method (ADM) to efficiently solve our problem. Experimental results on several benchmark datasets demonstrate the advantages of our method over the state-of-the-art ones.**

***Index Terms*—Bi-sparsity, hybrid noise, label enrichment, multilabel learning.**

I. Introduction

MULTILABEL learning deals with the problem of assigning one instance with multiple labels simultaneously. For example, a document may belong to multiple different topics, while an image usually contains more than one type of object and, one music can be annotated with more than one tag reflecting different styles. Due to its importance in real-world applications, a number of methods [1]–[6] on multilabel classification have been proposed, which have been successfully used in many applications. Generally, compared with binary and multiclass classification, multilabel learning is more challenging due to the underlying complex correlation among multiple labels. Although many multilabel classification methods have been developed and found useful in diverse applications, multilabel learning is still rather challenging, especially when the training data contain complex noise [7]–[9].

In real-world applications, data may contain noise which is defined as anything that obscures the relationship between the features of an instance and classes [10], [11]. On the one hand, some recent methods [12]–[15] have been proposed for label noise. The representative methods usually focus on addressing an incomplete label. Some works [9], [16] consider *weak label* cases with a semisupervised manner, and aim to complete the missing labels with transductive learning. The work in [17] addresses multilabel learning with *incomplete* class assignment by taking rank strategy and group lasso technique. The method proposed in [7] tries to address large-scale training under the missing label case. On the other hand, since observed values of features usually tend to be affected, features themselves are usually noisy [18], [19]. For example, images may be corrupted and features of text may be affected by the dull words. Some methods [20]–[24] have been proposed for feature noise. However, in real-world data, noise is usually hybrid, that is, mixed with both label noise and feature noise (as shown in Fig. 1), which makes multilabel learning much more challenging.

Although different types of noise have been separately considered in existing works, noise contained in real-world data are usually relatively complex or hybrid due to the complexity of data generation. Unfortunately, most existing multilabel learning methods just consider either feature noise or label noise. To address ubiquitous complex noise, we jointly consider different types of noise, that is, feature noise, label noise, and hybrid noise, and accordingly, propose a novel robust multilabel learning method called hybrid noise-oriented multilabel learning (HNOML). First, based on the original label vector, ideal labeling for each sample is learned by simultaneously exploring label correlation and the locality of data. Specifically, we explore the correlation among labels by learning a label-enrichment projection, which contains the intrinsic relationship among labels. At the same time, graph embedding is introduced to enforce the smoothness over the enrichment label space according to feature space. Second, based on the ideal enrichment label vectors, structured sparsity is employed to alleviate the sample-specific noise and labeling noise simultaneously.

Manuscript received November 26, 2018; accepted January 9, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61602337, Grant 61732011, Grant 61432011, Grant 61872190, Grant 61602345, Grant 61702358, and Grant 61876127. This paper was recommended by Associate Editor S. Ventura. *(Changqing Zhang and Ziwei Yu contributed equally to this work.) (Corresponding author: Qinghua Hu.)*

C. Zhang, Z. Yu, P. Zhu, and Q. Hu are with the College of Intelligence and Computing, Tianjin Key Laboratory of Machine Learning, Tianjin University, Tianjin 300072, China (e-mail: zhangchangqing@tju.edu.cn; yuziwei@tju.edu.cn; zhupengfei@tju.edu.cn; huqinghua@tju.edu.cn).

H. Fu is with the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE (e-mail: huazhufu@gmail.com).

L. Chen is with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: chenlei@njupt.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2894985

Our main contributions are summarized as follows. We propose a unified robust multilabel learning framework to address the data with hybrid noise, that is, joint feature and label noise. The propose method, HNOML, simultaneously addresses feature and label noise with *bi-sparsity regularization bridged with label enrichment*, where the label enrichment explores the intrinsic correlation among different classes, and the structured sparsity jointly imposed on prediction loss and label matrix reconstruction error provides the robustness for both corrupted features and noisy labeling. Since there are multiple blocks of variables involved in our problem, it is hard to optimize by updating all the variables simultaneously. Therefore, we employ alternating direction method (ADM) [25] for our problem. Extensive experiments are conducted on diverse benchmark datasets, validating the effectiveness of the proposed method over state-of-the-art multilabel learning approaches.

II. Related Work

According to the handling manner for label correlations, existing multilabel learning methods could be categorized into the following three types [2]. The first-order strategy addresses the problem in a label-by-label manner, that is, transforming the multilabel problem into multiple binary classification tasks or its variants [26]–[28]. Obviously, this strategy ignores correlation among labels, which is usually critical for the success of multilabel learning. The methods belonging to the second-order strategy usually take the label correlations into consideration by constructing pairwise relations among labels [29]–[31]. Although promising performances achieved, real correlations may be more complex than second-order one. Hence, the high-order strategy builds more complex relationships among labels for multilabel learning [32]–[34], however, they are usually computationally expensive. Recent researches regard the above strategies as crisp manner, and advocate that categorical labeling information is actually a simplification of the rich semantics encoded by multilabel training examples [35]–[37].

Recently, multilabel learning with noisy data [7], [16], [17], [20], [21], [33], [38]–[40] has received increasing attention because of its practical application background.There are two lines of robust multilabel learning methods. The first line of methods focus on learning with missing labels [7], [16], [17], [38]–[40]. The method in [17] maximizes the rank margin by exploring the group lasso regularizer which estimates the error in ranking the assigned classes against the unassigned ones. By using graph regularization according to the similarity matrix of instances, the method in [16] enforces the classification boundary for each label to go across low density regions. The methods [38], [40] try to recover a complete label matrix by taking label correlations into consideration. The second line of multilabel learning methods concentrate on addressing the low-quantity features. Toward feature noise, multilabel dimensionality reduction [20], [41]–[43] or feature selection methods [21], [44]–[48] have been proposed, which pursue the low-dimensional spaces to maximize the dependence between the mapped or selected features and the associated class labels.

III. Problem Statement

A. Preliminaries

Let $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{Y} = \{-1, +1\}^C$ denote the feature space and label space, where D and C are the dimensionality of feature space and number of classes, respectively. Given training data with input–output pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, accordingly, the input feature matrix can be represented as $\mathbf{X} \in \mathbb{R}^{D\times N}$ and, the label matrix is represented as $\mathbf{Y} \in \mathbb{R}^{C\times N}$, where N is the number of samples. Based on training data, the goal is to learn a prediction function $f : \mathcal{X} \to \mathcal{Y}$, which can accurately predict a label vector for a new coming instance. Considering the linear model, it aims at training a prediction model $\mathbf{W} \in \mathbb{R}^{C\times D}$ as follows:

$$\mathbf{y} = \mathbf{W}\mathbf{x}_i + \mathbf{e}_i \tag{1}$$

where $\mathbf{e}_i$ is the regression error corresponding to $\mathbf{x}_i$. To obtain a prediction model, the objective function often has the following form:

$$\min_{\mathbf{W}} \sum_{i=1}^{N} \mathcal{L}(\mathbf{y}_i, \mathbf{W}\mathbf{x}_i) + \lambda\mathcal{R}(\mathbf{W}) \tag{2}$$

where $\mathcal{L}(\cdot,\cdot)$ and $\mathcal{R}(\cdot)$ are the loss function and regularization term for the learned model $\mathbf{W}$, respectively. Most existing works [49], [50] usually focus on designing a reasonable regularizer on $\mathbf{W}$ under different assumptions.

B. HNOML: Our Multilabel Learning Model

In this paper, we focus on robust multilabel learning with training data containing hybrid noise. To this end, we address this problem by using bi-sparsity regularization bridged with label enrichment in a unified framework. Specifically, we explore the correlation among different class labels with label enrichment, in which an ideal enriched label matrix corresponding to the feature matrix is obtained. In this way, the labeling is improved by substituting the original label matrix with the enriched label matrix for regression. Based on the enriched label matrix, we impose structured sparsity on both prediction loss and label matrix reconstruction error to simultaneously address feature and label noise and, thus, induce our HNOML model.

To obtain the enriched label matrix, we introduce an explicit mapping to explore the correlation among labels and, thus, label noise could be alleviated. With self-representation manner, the mapping $\mathbf{B} \in \mathbb{R}^{C\times C}$ is obtained which captures the correlation among C different classes. For example, if "car" and "road" are labeled simultaneously for most of training samples, then the correlation will be strong and implied in the enriching projection $\mathbf{B}$. Then, we obtain a general form of objective as

$$\min_{\mathbf{B},\mathbf{W}} \sum_{i=1}^{N} \mathcal{L}\big(\mathbf{B}\mathbf{y}_i, \mathbf{W}\mathbf{x}_i\big) + \mathcal{R}(\mathbf{W}). \tag{3}$$

Based on the enriched labels, the prediction model $\mathbf{W}$ will be more reasonable since more accurate relationship between labels and features are embedded.

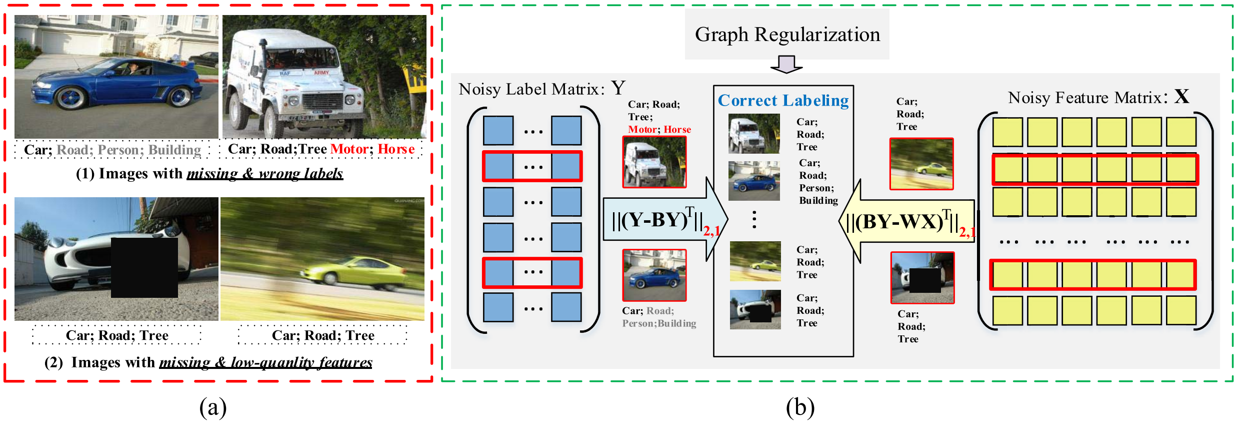


Fig. 1. Training data with (a) hybrid noise and (b) our model.

For learning the projection $\mathbf{B}$ from noisy labeling data, we should guarantee the reasonability of the learned projection $\mathbf{B}$. Therefore, we constrain the enriched label vectors to satisfy the following criteria.

1) The relationships between samples for enriched label vectors and original label vectors should be basically consistent.
2) To take advantages of the locality of data, that is, the similar pair of instances should have the similar enriched label vectors, graph embedding technique is introduced.
3) There may exist label noise for a few samples, hence sample-specific reconstruction error should be taken into consideration in label space.

According to the above analysis, to ensure the consistence between the enriched label vectors and the original label vectors, we define the following equation to measure the inconsistence between the enriched label matrix and original label matrix as

$$\Lambda(\mathbf{BY}, \mathbf{Y}) = ||\mathbf{Y} - \mathbf{BY}||_{2,1} \tag{4}$$

where the product $\mathbf{By}$ is a set of learned affine measurements of the original label vector $\mathbf{y}$, which captures salient features of the labels used to model their dependencies [51].

The structured sparsity, that is, $\ell_{2,1}$-norm for a matrix $\mathbf{A} \in \mathbb{R}^{P\times Q}$, is defined as

$$\|\mathbf{A}\|_{2,1} = \sum_{i=1}^{P} \sqrt{\sum_{j=1}^{Q} a_{ij}^2}. \tag{5}$$

The structure sparsity loss can deal with the sample-specific noise due to its row-wise sparsity property [52], [53].

Recall that our method tries to obtain enriched label vectors in accordance with the locality of data lying in feature space, accordingly, we employ a nearest neighbor graph on a scatter of data points to model the geometric structure of data and enforce the consistence between feature vectors and enriched label vectors. Specifically, the affinity matrix is constructed through the nearest neighbor graph as

$$s_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) & \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\mathcal{N}_k(\mathbf{x})$ is the set of k-nearest neighbors of the sample $\mathbf{x}$. The distance of label vectors $\mathbf{By}_i$ and $\mathbf{By}_j$ is defined as

$$d(\mathbf{By}_i, \mathbf{By}_j) = ||\mathbf{By}_i - \mathbf{By}_j||^2 \tag{7}$$

which is used to measure the "dissimilarity" between the enriched label vectors of two data points with respect to the learned projection $\mathbf{B}$. With the above defined affinity matrix $\mathbf{S}$, the consistence between enriched label vectors and feature vectors is measured as

$$\begin{aligned}\Omega(\mathbf{X}, \mathbf{BY}) &= \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{N} s_{ij}||\mathbf{By}_i - \mathbf{By}_j||^2 s_{ij} \\ &= \mathrm{Tr}\left(\mathbf{BYLY}^T\mathbf{B}^T\right)\end{aligned} \tag{8}$$

where $\mathrm{Tr}(\cdot)$ denote the trace of a matrix. $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is a Laplacian matrix, in which $\mathbf{D}$ is a diagonal degree matrix with $d_{ii} = \sum_{j=1}^{N} s_{ij}$. Based on the latent enriched label vectors, we aim to learn a reasonable prediction model $\mathbf{W}$. Therefore, we have the objective function as

$$\begin{aligned}\min_{\mathbf{W},\mathbf{B}} &\sum_{i=1}^{N} \|\mathbf{By}_i - \mathbf{Wx}_i\|^2 + \alpha \sum_{i=1}^{N} \|\mathbf{By}_i - \mathbf{By}_j\|^2 s_{ij} \\ &+ \beta \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{By}_i\|^2 + \gamma \|\mathbf{W}\|_F^2.\end{aligned} \tag{9}$$

In this objective function, we learns the final prediction model $\mathbf{W}$ and projection $\mathbf{B}$ jointly. Considering the sample-specific error over both features and labels, we can rewrite the above objective function into a more compact matrix form

$$\begin{aligned}\min_{\mathbf{W},\mathbf{B}} &\underbrace{\|(\mathbf{BY} - \mathbf{WX})^T\|_{2,1}}_{\text{Structured Loss}} + \underbrace{\alpha \mathrm{Tr}\left(\mathbf{BYLY}^T\mathbf{B}^T\right)}_{\text{Label Embedding}} \\ &+ \underbrace{\beta\|(\mathbf{Y} - \mathbf{BY})^T\|_{2,1}}_{\text{Label Enriching}} + \underbrace{\gamma\|\mathbf{W}\|_F^2}_{\text{Model Regularization}}\end{aligned} \tag{10}$$

where the structured sparsity is introduced. It is noteworthy that, beyond label enriching to alleviate label noise, the structured sparsity also provides robustness for the model. Specifically, the structured sparsity imposed on the first term addresses the sample-specific feature noise, while the structured sparsity on third term is used to resolve the sample-specific label noise.

For the first term in our model, since we consider the enriched label matrix $\mathbf{BY}$ as the ideal labeling, the structured sparsity loss is employed to introduce the robustness for the sample-specific outliers instead of feature-specific error [54], [55]. The second term explores the manifold of data, that is, the distance of a pair of enriched label vectors will be small if the pair of samples are similar in the feature space. The third term enforces the consistence between the enriched label vectors and the original label vectors, simultaneously constrained with structured sparsity to handle possible sample-specific label noise. Therefore, our objective function simultaneously explores the correlations among classes, addresses noisy labeling, and enhances robustness for corrupted features in a unified framework.

IV. Optimization

There are two blocks of variables in our objective function in (10). To optimize the problem in (10), we adopt alternating direction minimizing strategy and divide the objective function into two subproblems, that is, $\mathbf{W}$-subproblem and $\mathbf{B}$-subproblem. The optimization for them are as follows.

$\mathbf{W}$-Subproblem: To update $\mathbf{W}$, we fix $\mathbf{B}$ and should solve the subproblem with respect to $\mathbf{W}$ as follows:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \|(\mathbf{BY} - \mathbf{WX})^T\|_{2,1} + \gamma\|\mathbf{W}\|_F^2.$$

Setting the derivative of the above function with respect to $\mathbf{W}$ to zero, we have

$$\frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} = 2\mathbf{WXDX}^T - 2\mathbf{BYDX}^T + 2\gamma\mathbf{W} = 0$$

where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = (1/[2\|(\mathbf{BY} - \mathbf{WX})_i^T\|])$. $\mathbf{W}$ is updated by the following rule:

$$\mathbf{W}^* = \mathbf{BYDX}^T\left(\mathbf{XDX}^T + \gamma\mathbf{I}\right)^{-1}. \tag{11}$$

$\mathbf{B}$-Subproblem: With $\mathbf{W}$ being fixed, we should solve the subproblem with respect to $\mathbf{B}$ and have the following optimization problem:

$$\begin{aligned}\mathbf{B}^* = \arg\min_{\mathbf{B}} &\|(\mathbf{BY} - \mathbf{WX})^T\|_{2,1} + \alpha\mathrm{Tr}\left(\mathbf{BYLY}^T\mathbf{B}^T\right)\\ &+ \beta\|(\mathbf{Y} - \mathbf{BY})^T\|_{2,1}.\end{aligned}$$

It is easy to show the following equations:

$$\begin{aligned}\mathscr{L}(\mathbf{B}) &= \|(\mathbf{BY} - \mathbf{WX})^T\|_{2,1} + \alpha\mathrm{Tr}\left(\mathbf{BYLY}^T\mathbf{B}^T\right)\\ &\quad+ \beta\|(\mathbf{Y} - \mathbf{BY})^T\|_{2,1}\\ &= \mathrm{Tr}\left((\mathbf{BY} - \mathbf{WX})\mathbf{D}_1(\mathbf{BY} - \mathbf{WX})^T\right)\\ &\quad+ \alpha\mathrm{Tr}\left(\mathbf{BYLY}^T\mathbf{B}^T\right)\\ &\quad+ \beta\mathrm{Tr}\left((\mathbf{Y} - \mathbf{BY})\mathbf{D}_2(\mathbf{Y} - \mathbf{BY})^T\right).\end{aligned} \tag{12}$$

Setting the derivative of the above function with respect to $\mathbf{B}$ to zero, the following equation is obtained:

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{B})}{\partial \mathbf{B}} &= 2\mathbf{BYD}_1\mathbf{Y}^T - 2\mathbf{WXD}_1\mathbf{Y}^T\\ &\quad+ 2\alpha\mathbf{BYLY}^T - 2\beta\mathbf{YD}_2\mathbf{Y}^T + 2\beta\mathbf{BYD}_2\mathbf{Y}^T = 0\end{aligned}$$

Algorithm 1: Optimization Algorithm of HNOML

Input: Training data: $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, and α, β, and γ.
Initialize: $\mathbf{B} = \mathbf{I}$.
while *not converged* **do**
  Fix $\mathbf{B}$ update $\mathbf{W} \leftarrow$ Eq. (11);
  Fix $\mathbf{W}$ update $\mathbf{B} \leftarrow$ Eq. (13);
  Check the convergence conditions;
end
Output: $\mathbf{W}$, $\mathbf{B}$.

---

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are the diagonal matrices with $D_{1,ii} = (1/[2\|(\mathbf{BY} - \mathbf{WX})_i\|])$ and $D_{2,ii} = (1/[2\|(\mathbf{Y} - \mathbf{BY})_i^T\|])$. Then, we can update $\mathbf{B}$ by the following rule:

$$\begin{aligned}\mathbf{B}^* &= \left(\mathbf{WXD}_1\mathbf{Y}^T + \beta\mathbf{YD}_2\mathbf{Y}^T\right)\\ &\quad\times\left(\mathbf{YD}_1\mathbf{Y}^T + \alpha\mathbf{YLY}^T + \beta\mathbf{YD}_2\mathbf{Y}^T\right)^{-1}.\end{aligned} \tag{13}$$

The alternating optimization method is carried out until convergence or the maximum iteration number reached. Since alternating minimization may get stuck in a local minimum, a sensible initialization is usually necessary for a promising result. Since random initialization is risky, we initialize $\mathbf{B}$ with $\mathbf{B} = \mathbf{I}$ which equals the sparsest correlation among labels. The procedure for optimization HNOML is summarized as Algorithm 1.

A. Convergence Analysis

Theorem 1: The objective function in (12) is guaranteed to convergence with alternating direction method.

Proof: For convenience of description, we rewrite the objective function which we should minimize as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \mathbf{B}) &= \|(\mathbf{BY} - \mathbf{WX})^T\|_{2,1} + \alpha\mathrm{Tr}\left(\mathbf{BYLY}^T\mathbf{B}^T\right)\\ &\quad+ \beta\|(\mathbf{Y} - \mathbf{BY})^T\|_{2,1} + \gamma\|\mathbf{W}\|_F^2.\end{aligned} \tag{14}$$

Given $\mathbf{B}$ after the t-th iteration, that is, $\mathbf{B}^t$, we have the following inferences:

$$\begin{aligned}\mathbf{W}^{t+1} &= \arg\min_{\mathbf{W}} ||\left(\mathbf{B}^t\mathbf{Y} - \mathbf{W}^t\mathbf{X}\right)^T||_{2,1} + \gamma||\mathbf{W}^t||_F^2\\ &\Rightarrow \mathrm{Tr}\left(\left(\mathcal{B}^t - \mathcal{W}^{t+1}\right)\mathbf{D}^t\left(\mathcal{B}^t - \mathcal{W}^{t+1}\right)\right) + \gamma||\mathbf{W}^{t+1}||_F^2\\ &\leq \mathrm{Tr}\left(\left(\mathcal{B}^t - \mathcal{W}^t\right)\mathbf{D}^t\left(\mathcal{B}^t - \mathcal{W}^t\right)^T\right) + \gamma||\mathbf{W}^t||_F^2\end{aligned} \tag{15}$$

where $\mathbf{D}^t$ is a diagonal matrix with $D_{ii}^t = (1/[2\|(\mathbf{B}^t\mathbf{Y} - \mathbf{W}^t\mathbf{X})_i^T\|])$. We define $\mathcal{B}^t = (\mathbf{B}^t\mathbf{Y})^T$ and $\mathcal{W}^t = (\mathbf{W}^t\mathbf{X})^T$ for simplicity. Then, it is easy to show that

$$\begin{aligned}&\sum_i \frac{||\left(\mathcal{B}^t - \mathcal{W}^{t+1}\right)_i||_2^2}{2||(\mathcal{B}^t - \mathcal{W}^t)_i||_2} + \gamma||\mathbf{W}^{t+1}||_F^2\\ &\leq \sum_i \frac{||\left(\mathcal{B}^t - \mathcal{W}^t\right)_i||_2^2}{2||(\mathcal{B}^t - \mathcal{W}^t)_i||_2} + \gamma||\mathbf{W}^t||_F^2\\ &\Rightarrow ||\left(\mathcal{B}^t - \mathcal{W}^{t+1}\right)||_{2,1} + \gamma||\mathbf{W}^{t+1}||_F^2\\ &\quad- \left(||\left(\mathcal{B}^t - \mathcal{W}^{t+1}\right)||_{2,1} - \sum_i \frac{||\left(\mathcal{B}^t - \mathcal{W}^{t+1}\right)_i||_2^2}{2||(\mathcal{B}^t - \mathcal{W}^t)_i||_2}\right)\end{aligned}$$

$$\leq ||(\mathcal{B}^t - \mathcal{W}^t)||_{2,1} + \gamma ||\mathbf{W}^t||_F^2 - \left(||(\mathcal{B}^t - \mathcal{W}^t)||_{2,1} - \sum_i \frac{||(\mathcal{B}^t - \mathcal{W}^t)_i||_2^2}{2||(\mathcal{B}^t - \mathcal{W}^t)_i||_2} \right). \quad (16)$$

According to $\sqrt{a} - (a/2\sqrt{b}) \leq \sqrt{b} - (b/2\sqrt{a})$ [22], we have

$$||\left(\mathcal{B}^t - \mathcal{W}^{t+1}\right)||_{2,1} - \sum_i \frac{||(\mathcal{B}^t - \mathcal{W}^{t+1})_i||_2^2}{2||(\mathcal{B}^t - \mathcal{W}^t)_i||_2} \leq ||(\mathcal{B}^t - \mathcal{W}^t)||_{2,1} - \sum_i \frac{||(\mathcal{B}^t - \mathcal{W}^t)_i||_2^2}{2||(\mathcal{B}^t - \mathcal{W}^t)_i||_2}. \quad (17)$$

Therefore, according to (16) and (17), it is not difficult to show

$$||\left(\mathcal{B}^t - \mathcal{W}^{t+1}\right)||_{2,1} + \gamma ||\mathbf{W}^{t+1}||_F^2 \leq ||(\mathcal{B}^t - \mathcal{W}^t)||_{2,1} + \gamma ||\mathbf{W}^t||_F^2.$$

Hence, we have

$$\mathcal{L}\left(\mathbf{W}^{t+1}, \mathbf{B}^t\right) \leq \mathcal{L}\left(\mathbf{W}^t, \mathbf{B}^t\right). \quad (18)$$

Similarly, we have

$$\mathcal{L}\left(\mathbf{W}^{t+1}, \mathbf{B}^{t+1}\right) \leq \mathcal{L}\left(\mathbf{W}^{t+1}, \mathbf{B}^t\right). \quad (19)$$

Based on the inequations (18) and (19), we obtain

$$\mathcal{L}\left(\mathbf{W}^{t+1}, \mathbf{B}^{t+1}\right) \leq \mathcal{L}\left(\mathbf{W}^{t+1}, \mathbf{B}^t\right) \leq \mathcal{L}\left(\mathbf{W}^t, \mathbf{B}^t\right).$$

According to the above results, Algorithm 1 is guaranteed to converge to a local optimal solution. ■

B. Complexity Analysis

There are two subproblems in our optimization procedure, that is, **W**-subproblem and **B**-subproblem. For $\mathbf{W} \in \mathbb{R}^{C\times D}$ and $\mathbf{B} \in \mathbb{R}^{C\times C}$, the complexity of these subproblems are $O(CND + C^2N + CN^2 + \mathrm{CD}^2 + D^3)$ and $O(CND + CN^2 + C^2N + C^3)$, respectively.

V. Experiment

A. Experiment Settings

We conduct our experiments on 20 benchmark datasets of diverse applications. All these datasets are from Mulan[1] and LEAR websites.[2] Specifically, the description of features could be found in [29] (Yeast), [4] (TMC), [5] (Emotions), [56] (CAL500), [57] (Medical), [26] (Scene), [58] (Genbase), [59] (Bibtex), [60] (Birds), and [61] (Corel16k001). For Arts, Computers, Education, Entertainment, Health, Recreation, and Reference, the detailed information could be found in [62]. For Corel5k, Pascal, and Espgame, we use DenseHue for these image datasets. The detailed statistics information of these datasets are shown in Table I. Some examples from these datasets are shown in Fig. 2. Note that, label cardinality (LCard) is a standard measure of "multilabeled-ness" [1], which indicates the average number of labels relevant to each instance. We randomly select 2/3 of the total samples from each dataset as training data with the remaining as test set. Due to randomness involved, the average results with standard deviation are reported for 30 runs.

[1]http://mulan.sourceforge.net/datasets-mlc.html

[2]lear.inrialpes.fr/people/guillaumin/data.php

(a) (b) (c)

Fig. 2. Example images used in our experiments. (a) Corel5k. (b) Pascal. (c) Espgame.

TABLE I
STATISTICS OF DATASETS

Dataset	Domain	#Instance	#Feature	#Label	LCard
Yeast	BIOLOGY	2417	103	14	4.2
TMC	TEXT	7077	500	22	2.2
Emotions	MUSIC	593	72	6	1.9
CAL500	MUSIC	502	68	174	26.0
Corel5k	IMAGE	5000	499	374	3.5
Pascal	IMAGE	10000	100	20	1.5
Espgame	IMAGE	20770	100	260	4.7
Medical	TEXT	978	1499	45	1.3
Scene	IMAGE	2407	294	6	1.1
Genbase	BIOLOGY	662	1186	27	1.3
Arts	TEXT	5000	462	26	1.6
Bibtex	TEXT	7395	1836	159	2.4
Birds	AUDIO	645	260	19	1.0
Corel16k001	IMAGE	13766	500	153	2.9
Computers	TEXT	5000	300	33	1.5
Education	TEXT	5000	300	33	1.5
Entertainment	TEXT	5000	300	21	1.4
Health	TEXT	5000	300	32	1.6
Recreation	TEXT	5000	300	22	1.4
Reference	TEXT	5000	300	33	1.2

Similarly to existing works [28], [37], five diverse metrics are employed for evaluation, and these metrics favor different properties for multilabel classification. Accordingly, we report results in terms of these diverse measures to perform a comprehensive evaluation. For *Hamming loss*, *Ranking loss*, *One-error*, and *Coverage*, smaller value indicates better classification performance, while larger value of *Average precision* indicates better performance. Please refer the work in [63] for the details of these evaluation metrics. We compare our method with a number of state-of-the-art multilabel classification methods, including binary relevance (BR) [1], label powerset (LP), pruned sets (PS) [64], and classifier chain (CC) [33] as the baselines, the lazy multilabel methods based on k-nearest neighbors (ML-kNN) [28], three ensemble methods: 1) random k-labelsets (RAkEL) [32]; 2) ensemble of PS (EPS) [64]; and 3) ensembles of CCs (ECC) [33]. We also compare ours with the fast image tagging

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
RESULTS (MEAN ± STD.) OF MULTILABEL LEARNING ALGORITHMS. ↓ (↑) INDICATES THE SMALLER (LARGER), THE BETTER. THE VALUE IN RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCES, RESPECTIVELY

Method	Metrics	BR	LP	PS	CC	ML-kNN	EPS	Ours
Yeast	Hamming Loss↓	.253±.004	.282±.005	.278±.005	.273±.007	.196±.004	.214±.005	.202±.004
	Ranking Loss ↓	.322±.011	.408±.008	.329±.009	.336±.015	.169±.005	.205±.003	.179±.007
	One-error ↓	.408±.040	.520±.019	.352±.012	.356±.018	.233±.010	.266±.009	.229±.013
	Coverage ↓	.670±.015	.680±.008	.633±.011	.648±.017	.451±.008	.482±.008	.469±.006
	Average Precision ↑	.614±.008	.566±.008	.637±.007	.620±.017	.761±.006	.728±.003	.757±.011
TMC	Hamming Loss↓	.075±.001	.092±.001	.073±.001	.061±.000	.072±.001	.067±.001	.061±.001
	Ranking Loss ↓	.171±.007	.419±.008	.135±.002	.124±.002	.081±.002	.089±.002	.053±.002
	One-error ↓	.336±.010	.590±.013	.281±.014	.268±.004	.278±.009	.251±.004	.196±.003
	Coverage ↓	.333±.010	.562±.010	.258±.003	.259±.004	.179±.004	.201±.002	.143±.005
	Average Precision ↑	.684±.008	.435±.009	.733±.002	.756±.001	.751±.004	.767±.003	.822±.006
Emotions	Hamming Loss↓	.265±.015	.277±.010	.272±.024	.273±.022	.201±.009	.208±.010	.200±.008
	Ranking Loss ↓	.309±.021	.345±.022	.303±.028	.310±.030	.173±.015	.183±.014	.172±.013
	One-error ↓	.414±.031	.461±.032	.428±.038	.429±.035	.269±.033	.311±.037	.279±.019
	Coverage ↓	.438±.029	.451±.024	.421±.023	.431±.036	.312±.011	.316±.012	.315±.014
	Average Precision ↑	.687±.017	.661±.018	.686±.023	.685±.020	.794±.016	.780±.017	.796±.009
CAL500	Hamming Loss↓	.166±.003	.200±.003	—	.179±.003	.139±.002	.141±.004	.136±.003
	Ranking Loss↓	.324±.013	.658±.007	—	.373±.005	.184±.004	.198±.031	.183±.014
	One-error ↓	.739±.037	.988±.009	—	.722±.042	.122±.017	.190±.102	.111±.037
	Coverage ↓	.973±.005	.982±.002	—	.978±.002	.749±.009	.765±.018	.741±.018
	Average Precision↑	.339±.009	.116±.002	—	.311±.005	.490±.007	.475±.056	.506±.016
Corel5k	Hamming Loss↓	.010±.000	.017±.000	.013±.000	.010±.000	.009±.000	.011±.000	.009±.000
	Ranking Loss↓	.149±.003	.748±.007	.405±.020	.190±.003	.136±.003	.487±.019	.138±.003
	One-error ↓	.707±.013	.985±.004	.820±.014	.722±.002	.733±.010	.803±.017	.670±.007
	Coverage ↓	.343±.006	.929±.003	.686±.019	.443±.004	.310±.005	.770±.017	.325±.007
	Average Precision ↑	.243±.004	.021±.003	.167±.009	.227±.004	.244±.005	.167±.007	.286±.003
Pascal	Hamming Loss↓	.086±.001	.110±.009	.109±.001	.099±.001	.070±.001	.072±.000	.070±.001
	Ranking Loss ↓	.381±.009	.431±.004	.377±.007	.367±.001	.246±.003	.277±.005	.227±.003
	One-error ↓	.692±.012	.793±.006	.712±.010	.710±.001	.584±.009	.588±.006	.587±.006
	Coverage ↓	.455±.008	.483±.004	.437±.006	.438±.003	.309±.003	.341±.005	.283±.005
	Average Precision ↑	.379±.008	.288±.005	.359±.008	.383±.003	.469±.005	.463±.007	.478±.002
Espgame	Hamming Loss↓	.018±.000	.030±.000	.019±.000	.018±.000	.017±.000	.017±.000	.017±.000
	Ranking Loss ↓	.266±.003	.499±.003	.393±.002	.244±.002	.190±.001	.380±.001	.183±.001
	One-error ↓	.658±.004	.931±.003	.678±.006	.651±.011	.618±.004	.604±.011	.590±.001
	Coverage ↓	.608±.008	.802±.003	.726±.004	.563±.004	.453±.002	.722±.003	.440±.002
	Average Precision ↑	.221±.001	.057±.001	.170±.002	.219±.005	.260±.001	.200±.003	.272±.000
Medical	Hamming Loss↓	.011±.001	.014±.001	.013±.001	.011±.001	.016±.001	.012±.001	.011±.001
	Ranking Loss ↓	.071±.013	.138±.013	.079±.013	.080±.012	.045±.007	.060±.009	.024±.006
	One-error ↓	.189±.025	.239±.016	.195±.019	.191±.020	.258±.018	.202±.020	.162±.027
	Coverage ↓	.096±.017	.169±.013	.104±.016	.104±.013	.065±.010	.083±.011	.036±.023
	Average Precision ↑	.832±.018	.746±.013	.814±.017	.827±.014	.796±.017	.827±.012	.877±.017
Scene	Hamming Loss↓	.136±.004	.149±.006	.149±.004	.143±.007	.093±.004	.101±.005	.110±.003
	Ranking Loss ↓	.236±.017	.219±.010	.211±.008	.251±.027	.085±.005	.106±.006	.103±.005
	One-error ↓	.412±.014	.414±.017	.410±.007	.388±.022	.246±.010	.282±.010	.272±.013
	Coverage ↓	.214±.016	.198±.008	.191±.006	.227±.023	.085±.005	.103±.006	.101±.006
	Average Precision ↑	.715±.011	.722±.010	.727±.005	.716±.017	.854±.006	.828±.006	.832±.008
Genbase	Hamming Loss↓	.002±.000	.002±.001	.004±.001	.002±.000	.006±.001	.004±.001	.001±.000
	Ranking Loss ↓	.005±.003	.011±.004	.016±.008	.005±.003	.008±.004	.012±.006	.001±.002
	One-error ↓	.005±.005	.012±.008	.014±.009	.005±.004	.012±.007	.014±.008	.001±.002
	Coverage ↓	.016±.004	.024±.004	.032±.010	.016±.004	.024±.007	.027±.007	.012±.003
	Average Precision ↑	.989±.005	.981±.007	.973±.010	.989±.005	.985±.006	.977±.008	.996±.003
Arts	Hamming Loss↓	.069±.001	.090±.000	.083±.000	.077±.001	.061±.000	.059±.000	.054±.001
	Ranking Loss ↓	.259±.013	.406±.007	.310±.003	.239±.012	.155±.004	.203±.003	.151±.004
	One-error ↓	.628±.013	.714±.012	.659±.001	.639±.010	.654±.008	.562±.009	.473±.009
	Coverage ↓	.341±.010	.486±.007	.385±.004	.322±.011	.213±.006	.275±.003	.222±.006
	Average Precision ↑	.472±.011	.354±.008	.424±.003	.477±.006	.498±.007	.528±.006	.611±.008
Bibtex	Hamming Loss↓	.015±.000	.021±.001	—	.015±.000	.014±.000	—	.012±.000
	Ranking Loss ↓	.164±.004	.422±.007	—	.175±.010	.205±.004	—	.111±.003
	One-error ↓	.506±.011	.784±.006	—	.512±.008	.586±.008	—	.372±.008
	Coverage ↓	.305±.009	.579±.009	—	.324±.018	.336±.005	—	.207±.006
	Average Precision ↑	.469±.008	.213±.009	—	.459±.008	.349±.006	—	.564±.004
Birds	Hamming Loss↓	.060±.003	.078±.004	.066±.004	.062±.002	.059±.001	.056±.003	.051±.004
	Ranking Loss ↓	.221±.010	.271±.016	.203±.017	.219±.001	.121±.007	.155±.012	.111±.015
	One-error ↓	.422±.032	.471±.019	.391±.025	.403±.016	.378±.018	.318±.022	.293±.017
	Coverage ↓	.291±.013	.321±.019	.269±.014	.290±.007	.174±.005	.225±.015	.172±.017
	Average Precision ↑	.633±.022	.589±.018	.645±.017	.639±.005	.692±.010	.704±.005	.750±.017
Corel16k001	Hamming Loss↓	.020±.000	.032±.000	.027±.000	.020±.000	.019±.000	.020±.000	.019±.000
	Ranking Loss ↓	.187±.001	.468±.004	.354±.002	.214±.001	.173±.001	.301±.003	.152±.000
	One-error ↓	.709±.006	.969±.003	.787±.007	.755±.003	.751±.006	.708±.002	.651±.006
	Coverage ↓	.366±.003	.716±.003	.601±.005	.410±.001	.336±.002	.551±.005	.304±.001
	Average Precision ↑	.287±.003	.066±.003	.193±.007	.268±.002	.281±.002	.263±.001	.336±.001

(FastTag) method [65], ranking-based method (MLR-GL) [17], multilabel manifold learning [37], and multilabel classification via calibrated label ranking (CLR) [31]. CLR effectively produces an ensemble that combines the models learned by the conventional BR ranking and pairwise classification methods, and this ensemble technique is a well-know technique for

TABLE III
RESULTS (MEAN ± STD.) OF MULTILABEL LEARNING ALGORITHMS. ↓ (↑) INDICATES THE SMALLER (LARGER), THE BETTER. THE VALUE IN RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCES, RESPECTIVELY

Method	Metrics	BR	LP	PS	CC	ML-kNN	EPS	Ours
Computers	Hamming Loss↓	.054±.001	.061±.002	.058±.002	.056±.001	.037±.001	.037±.001	.042±.001
	Ranking Loss ↓	.350±.016	.508±.008	.391±.004	.327±.021	.083±.004	.169±.005	.095±.002
	One-error ↓	.594±.014	.665±.003	.601±.013	.578±.008	.406±.019	.420±.012	.391±.013
	Coverage ↓	.422±.017	.562±.007	.452±.004	.392±.022	.121±.004	.230±.004	.140±.001
	Average Precision ↑	.472±.012	.347±.007	.429±.011	.479±.012	.665±.012	.623±.008	.680±.008
Education	Hamming Loss↓	.060±.000	.065±.003	.061±.001	.061±.001	.040±.000	.042±.001	.042±.001
	Ranking Loss ↓	.412±.009	.543±.005	.433±.006	.374±.012	.086±.002	.179±.005	.098±.005
	One-error ↓	.691±.011	.753±.007	.698±.012	.670±.006	.559±.007	.580±.014	.553±.007
	Coverage ↓	.485±.008	.598±.002	.492±.008	.447±.013	.113±.002	.234±.007	.134±.005
	Average Precision ↑	.389±.003	.272±.005	.358±.001	.410±.008	.575±.007	.525±.010	.590±.004
Entertainment	Hamming Loss↓	.082±.001	.087±.002	.085±.002	.083±.001	.057±.001	.055±.002	.059±.000
	Ranking Loss ↓	.377±.011	.503±.015	.421±.007	.342±.008	.107±.001	.182±.005	.109±.003
	One-error ↓	.610±.009	.649±.019	.619±.005	.590±.009	.504±.008	.484±.005	.464±.006
	Coverage ↓	.431±.013	.538±.014	.462±.009	.395±.006	.142±.004	.231±.006	.147±.006
	Average Precision ↑	.471±.002	.381±.015	.439±.006	.498±.006	.625±.004	.607±.008	.661±.005
Health	Hamming Loss↓	.052±.001	.058±.001	.056±.000	.055±.001	.042±.001	.038±.001	.043±.001
	Ranking Loss ↓	.307±.016	.477±.011	.321±.005	.296±.008	.055±.002	.105±.003	.067±.005
	One-error ↓	.505±.014	.583±.012	.504±.007	.496±.012	.396±.015	.328±.006	.369±.006
	Coverage ↓	.413±.023	.572±.006	.417±.001	.398±.007	.094±.004	.175±.007	.119±.004
	Average Precision ↑	.535±.005	.408±.009	.516±.005	.545±.009	.702±.006	.719±.001	.733±.003
Recreation	Hamming Loss↓	.087±.001	.097±.001	.092±.001	.094±.002	.057±.000	.058±.000	.063±.001
	Ranking Loss ↓	.383±.004	.459±.004	.403±.003	.372±.002	.153±.003	.218±.005	.135±.002
	One-error ↓	.686±.006	.714±.008	.682±.004	.670±.008	.547±.010	.562±.006	.474±.014
	Coverage ↓	.440±.001	.515±.007	.460±.004	.425±.004	.195±.003	.270±.006	.179±.002
	Average Precision ↑	.413±.006	.355±.006	.397±.003	.426±.004	.571±.004	.538±.005	.628±.005
Reference	Hamming Loss↓	.040±.001	.045±.000	.043±.000	.041±.001	.029±.000	.028±.000	.035±.001
	Ranking Loss ↓	.328±.009	.475±.006	.393±.008	.292±.010	.078±.001	.178±.005	.094±.005
	One-error ↓	.589±.005	.629±.005	.591±.008	.563±.008	.448±.004	.459±.003	.465±.008
	Coverage ↓	.356±.009	.494±.008	.415±.007	.315±.012	.093±.002	.208±.005	.114±.006
	Average Precision ↑	.491±.007	.390±.004	.451±.007	.512±.010	.654±.003	.618±.004	.656±.003

TABLE IV
RESULTS (MEAN ± STD.) OF *ROBUST* MULTILABEL LEARNING ALGORITHMS. ↓ (↑) INDICATES THE SMALLER (LARGER), THE BETTER. THE VALUE IN RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCES, RESPECTIVELY

Method	Metrics	ECC	RAkEL	CLR	MLML	MLR-GL	FastTag	Ours
Computers	Hamming Loss↓	.037±.002	.041±.001	.041±.001	.046±.012	.525±.038	.047±.001	.042±.001
	Ranking Loss ↓	.098±.003	.207±.007	.092±.003	.351±.036	.320±.020	.107±.008	.095±.002
	One-error ↓	.404±.012	.427±.006	.413±.008	.469±.117	.428±.061	.462±.006	.391±.013
	Coverage ↓	.142±.002	.279±.008	.135±.004	.823±.026	.803±.020	.157±.009	.140±.001
	Average Precision ↑	.657±.009	.609±.005	.657±.003	.219±.029	.205±.016	.601±.009	.680±.008
Education	Hamming Loss↓	.041±.001	.043±.001	.046±.001	.049±.008	.378±.021	.045±.000	.042±.001
	Ranking Loss ↓	.094±.001	.249±.003	.106±.017	.384±.013	.329±.018	.128±.006	.098±.005
	One-error ↓	.545±.004	.555±.016	.606±.047	.535±.049	.522±.047	.720±.029	.553±.007
	Coverage ↓	.126±.001	.320±.003	.137±.018	.783±.043	.695±.031	.163±.008	.134±.005
	Average Precision ↑	.575±.002	.526±.007	.527±.047	.189±.012	.178±.015	.446±.018	.590±.004
Entertainment	Hamming Loss↓	.053±.001	.064±.001	.061±.001	.066±.022	.500±.033	.068±.001	.059±.000
	Ranking Loss ↓	.109±.003	.232±.010	.097±.002	.328±.040	.323±.012	.163±.007	.109±.003
	One-error ↓	.434±.002	.477±.012	.446±.003	.255±.094	.380±.052	.786±.034	.464±.006
	Coverage ↓	.147±.006	.289±.013	.132±.005	.854±.056	.852±.021	.196±.034	.147±.006
	Average Precision ↑	.661±.005	.600±.011	.662±.003	.301±.044	.258±.012	.414±.023	.661±.004
Health	Hamming Loss↓	.036±.001	.040±.001	.044±.000	.042±.015	.437±.024	.051±.001	.043±.001
	Ranking Loss ↓	.052±.001	.130±.005	.075±.023	.274±.032	.316±.022	.082±.005	.067±.005
	One-error ↓	.300±.001	.325±.008	.345±.042	.430±.072	.362±.065	.495±.011	.369±.006
	Coverage ↓	.094±.002	.211±.012	.130±.035	.713±.028	.711±.046	.126±.006	.119±.004
	Average Precision ↑	.756±.002	.707±.004	.681±.052	.296±.039	.303±.028	.615±.011	.733±.003
Recreation	Hamming Loss↓	.056±.001	.061±.001	.059±.000	.060±.008	.522±.031	.065±.001	.063±.001
	Ranking Loss ↓	.160±.005	.234±.008	.134±.005	.291±.020	.306±.009	.231±.023	.135±.002
	One-error ↓	.496±.015	.527±.009	.487±.008	.255±.025	.309±.029	.797±.043	.474±.014
	Coverage ↓	.206±.006	.296±.009	.178±.007	.933±.018	.936±.017	.273±.022	.179±.002
	Average Precision ↑	.599±.010	.555±.007	.610±.008	.302±.020	.280±.013	.355±.044	.628±.005
Reference	Hamming Loss↓	.028±.000	.029±.000	.030±.000	.032±.007	.455±.016	.036±.002	.035±.001
	Ranking Loss ↓	.082±.001	.218±.007	.096±.019	.355±.033	.321±.024	.130±.008	.094±.005
	One-error ↓	.431±.004	.444±.006	.455±.017	.525±.080	.566±.081	.527±.013	.465±.008
	Coverage ↓	.098±.002	.250±.008	.113±.021	.721±.036	.681±.035	.147±.009	.114±.006
	Average Precision ↑	.661±.005	.617±.007	.640±.024	.193±.010	.175±.014	.556±.016	.656±.003

robustness. We tune the parameters of our method on validation data from the set {0.001, 0.01, 0.1, 1, 10, 100, 1000}. For the number of neighbors k in (6), we empirically set it as 5 since it is observed that the performance varies little with different numbers. We try to tune the parameters of compared methods to the best performance as suggested ways.

TABLE V
RESULTS (MEAN ± STD.) OF *ROBUST* MULTILABEL LEARNING ALGORITHMS. ↓ (↑) INDICATES THE SMALLER (LARGER), THE BETTER. THE VALUE IN RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCES, RESPECTIVELY

Method	Metrics	ECC	RAkEL	CLR	MLML	MLR-GL	FastTag	Ours
	Hamming Loss ↓	.210±.004	.232±.005	.222±.005	.204±.005	.270±.001	.223±.019	.202±.004
	Ranking Loss ↓	.187±.006	.220±.006	.181±.004	.290±.007	.304±.013	.195±.008	.179±.007
Yeast	One-error ↓	.253±.012	.253±.012	.243±.008	.264±.089	.191±.041	.249±.013	.229±.013
	Coverage ↓	.474±.009	.474±.009	.482±.010	.944±.011	.957±.012	.495±.010	.468±.006
	Average Precision ↑	.743±.007	.712±.006	.743±.004	.503±.009	.473±.008	.737±.006	.757±.011
	Hamming Loss ↓	.067±.001	.070±.001	.057±.001	.060±.000	.293±.003	.070±.001	.061±.001
	Ranking Loss ↓	.055±.002	.080±.002	.037±.001	.082±.004	.115±.005	.056±.002	.053±.002
TMC	One-error ↓	.227±.006	.278±.009	.173±.003	.221±.003	.203±.073	.294±.015	.196±.003
	Coverage ↓	.142±.003	.179±.004	.113±.001	.787±.067	.727±.030	.139±.005	.143±.005
	Average Precision ↑	.800±.005	.752±.004	.847±.001	.631±.014	.517±.007	.774±.005	.822±.006
	Hamming Loss ↓	.201±.010	.219±.013	.255±.012	.197±.013	.275±.016	.307±.009	.200±.008
	Ranking Loss ↓	.166±.010	.189±.018	.180±.015	.154±.015	.162±.017	.186±.010	.172±.013
Emotions	One-error ↓	.273±.032	.308±.034	.296±.016	.133±.013	.083±.017	.330±.026	.279±.019
	Coverage ↓	.302±.008	.324±.019	.317±.017	.827±.013	.812±.044	.315±.010	.315±.014
	Average Precision ↑	.799±.017	.778±.017	.781±.012	.719±.018	.713±.029	.768±.011	.796±.009
	Hamming Loss ↓	.145±.002	.169±.002	.140±.023	.154±.003	.295±.003	.149±.005	.136±.003
	Ranking Loss ↓	.209±.004	.287±.005	.190±.005	.458±.006	.464±.004	.252±.011	.183±.014
CAL500	One-error ↓	.212±.023	.336±.042	.168±.086	.815±.028	.855±.006	.296±.040	.111±.037
	Coverage ↓	.791±.008	.949±.005	.764±.015	.862±.006	.856±.016	.882±.013	.741±.018
	Average Precision ↑	.466±.007	.397±.007	.487±.044	.197±.005	.188±.004	.434±.012	.506±.016
	Hamming Loss ↓	.009±.000	.010±.000	.010±.000	.010±.000	.200±.001	.010±.000	.009±.000
	Ranking Loss ↓	.139±.003	.666±.006	.147±.005	.348±.013	.332±.006	.273±.009	.138±.003
Corel5k	One-error ↓	.689±.013	.784±.013	.733±.013	.853±.005	.936±.012	.715±.010	.670±.007
	Coverage ↓	.317±.006	.903±.004	.322±.008	.650±.018	.608±.010	.566±.015	.325±.007
	Average Precision ↑	.264±.005	.099±.005	.228±.013	.069±.004	.047±.004	.217±.005	.286±.003
	Hamming Loss ↓	.070±.000	.075±.001	.072±.001	.070±.000	.410±.068	.073±.000	.070±.001
	Ranking Loss ↓	.236±.003	.312±.004	.198±.003	.266±.002	.292±.005	.233±.004	.227±.003
Pascal	One-error ↓	.585±.006	.615±.007	.570±.010	.600±.050	.800±.132	.593±.002	.587±.006
	Coverage ↓	.294±.003	.375±.004	.251±.004	.970±.007	.976±.007	.291±.005	.283±.005
	Average Precision ↑	.485±.004	.441±.004	.507±.006	.208±.003	.156±.004	.461±.006	.478±.002
	Hamming Loss ↓	.017±.000	.018±.000	.017±.000	.018±.000	.487±.035	.017±.000	.017±.000
	Ranking Loss ↓	.190±.001	.387±.001	.156±.001	.317±.000	.326±.002	.201±.003	.183±.001
Espgame	One-error ↓	.586±.008	.618±.009	.567±.005	.595±.006	.925±.003	.595±.008	.590±.001
	Coverage ↓	.459±.003	.729±.002	.388±.002	.962±.004	.942±.001	.474±.004	.440±.002
	Average Precision ↑	.282±.001	.201±.003	.305±.003	.086±.002	.045±.000	.268±.002	.272±.000
	Hamming Loss↓	.010±.000	.011±.001	.021±.002	.013±.000	.165±.001	.018±.006	.011±.001
	Ranking Loss ↓	.032±.006	.078±.006	.122±.017	.096±.000	.198±.006	.025±.015	.024±.006
Medical	One-error ↓	.137±.013	.183±.014	.599±.125	.380±.056	.576±.023	.338±.035	.162±.027
	Coverage ↓	.049±.009	.098±.007	.145±.020	.144±.028	.307±.008	.155±.023	.036±.023
	Average Precision ↑	.889±.007	.827±.013	.438±.074	.594±.048	.291±.017	.720±.032	.877±.017
	Hamming Loss↓	.101±.005	.106±.005	.138±.003	.078±.013	.228±.003	.178±.001	.110±.003
	Ranking Loss ↓	.106±.006	.106±.005	.106±.003	.050±.003	.105±.004	.096±.006	.103±.005
Scene	One-error ↓	.282±.010	.282±.012	.307±.010	.000±.000	.050±.081	.281±.015	.272±.013
	Coverage ↓	.103±.006	.103±.004	.104±.003	.684±.038	.800±.017	.093±.005	.101±.006
	Average Precision ↑	.828±.006	.829±.007	.817±.006	.862±.010	.678±.012	.832±.009	.832±.008
	Hamming Loss↓	.002±.000	.002±.000	.002±.001	.002±.000	.248±.009	.001±.000	.001±.000
	Ranking Loss ↓	.005±.003	.005±.003	.013±.005	.019±.026	.007±.005	.003±.003	.001±.002
Genbase	One-error ↓	.004±.004	.006±.005	.005±.005	.050±.041	.058±.050	.001±.002	.001±.002
	Coverage ↓	.016±.005	.016±.004	.030±.008	.065±.028	.067±.014	.014±.006	.012±.003
	Average Precision ↑	.991±.005	.989±.005	.985±.007	.952±.040	.932±.033	.995±.004	.996±.003
	Hamming Loss↓	.054±.000	.060±.000	.058±.001	.054±.000	.444±.005	.063±.000	.054±.001
	Ranking Loss ↓	.134±.003	.264±.004	.122±.002	.270±.015	.299±.015	.170±.005	.151±.004
Arts	One-error ↓	.500±.012	.561±.011	.517±.013	.322±.068	.438±.063	.597±.012	.473±.009
	Coverage ↓	.192±.003	.347±.003	.179±.000	.880±.029	.876±.027	.235±.007	.222±.006
	Average Precision ↑	.590±.006	.504±.006	.585±.004	.302±.021	.222±.007	.503±.009	.611±.008
	Hamming Loss↓	.013±.000	.014±.000	.013±.000	.013±.000	.292±.005	.014±.000	.012±.000
	Ranking Loss ↓	.089±.002	.270±.006	.066±.000	.086±.004	.317±.006	.098±.003	.111±.003
Bibtex	One-error ↓	.400±.009	.472±.011	.412±.006	.378±.037	.621±.051	.387±.011	.372±.008
	Coverage ↓	.170±.002	.431±.009	.123±.000	.592±.020	.862±.007	.175±.005	.207±.006
	Average Precision ↑	.553±.003	.423±.005	.554±.033	.408±.006	.116±.004	.559±.008	.564±.004
	Hamming Loss↓	.050±.001	.057±.002	.053±.001	.052±.002	.322±.006	.073±.003	.051±.004
	Ranking Loss ↓	.112±.001	.161±.003	.090±.006	.163±.019	.248±.023	.139±.015	.111±.015
Birds	One-error ↓	.288±.011	.326±.027	.288±.017	.350±.089	.459±.084	.445±.041	.293±.017
	Coverage ↓	.170±.005	.223±.006	.140±.002	.564±.057	.607±.038	.195±.018	.172±.017
	Average Precision ↑	.749±.006	.705±.012	.759±.009	.419±.029	.312±.026	.639±.030	.750±.017
	Hamming Loss↓	.019±.000	.020±.000	.019±.000	.019±.000	.318±.004	.019±.000	.019±.000
	Ranking Loss ↓	.168±.001	.400±.001	.140±.001	.275±.004	.339±.005	.217±.011	.152±.000
Corel16k001	One-error ↓	.698±.008	.767±.001	.672±.004	.735±.002	.867±.014	.683±.006	.651±.006
	Coverage ↓	.325±.002	.661±.002	.281±.002	.879±.009	.906±.007	.408±.018	.304±.001
	Average Precision ↑	.300±.005	.159±.001	.328±.001	.094±.002	.064±.002	.292±.007	.336±.001

B. Experimental Results

1) Quantitative Results: Tables II–V show the classification comparison of different methods on the benchmark datasets. Since each dataset is randomly divided into training and test parts, both the average performance and standard deviation are reported in terms of each evaluation measure. Based on the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

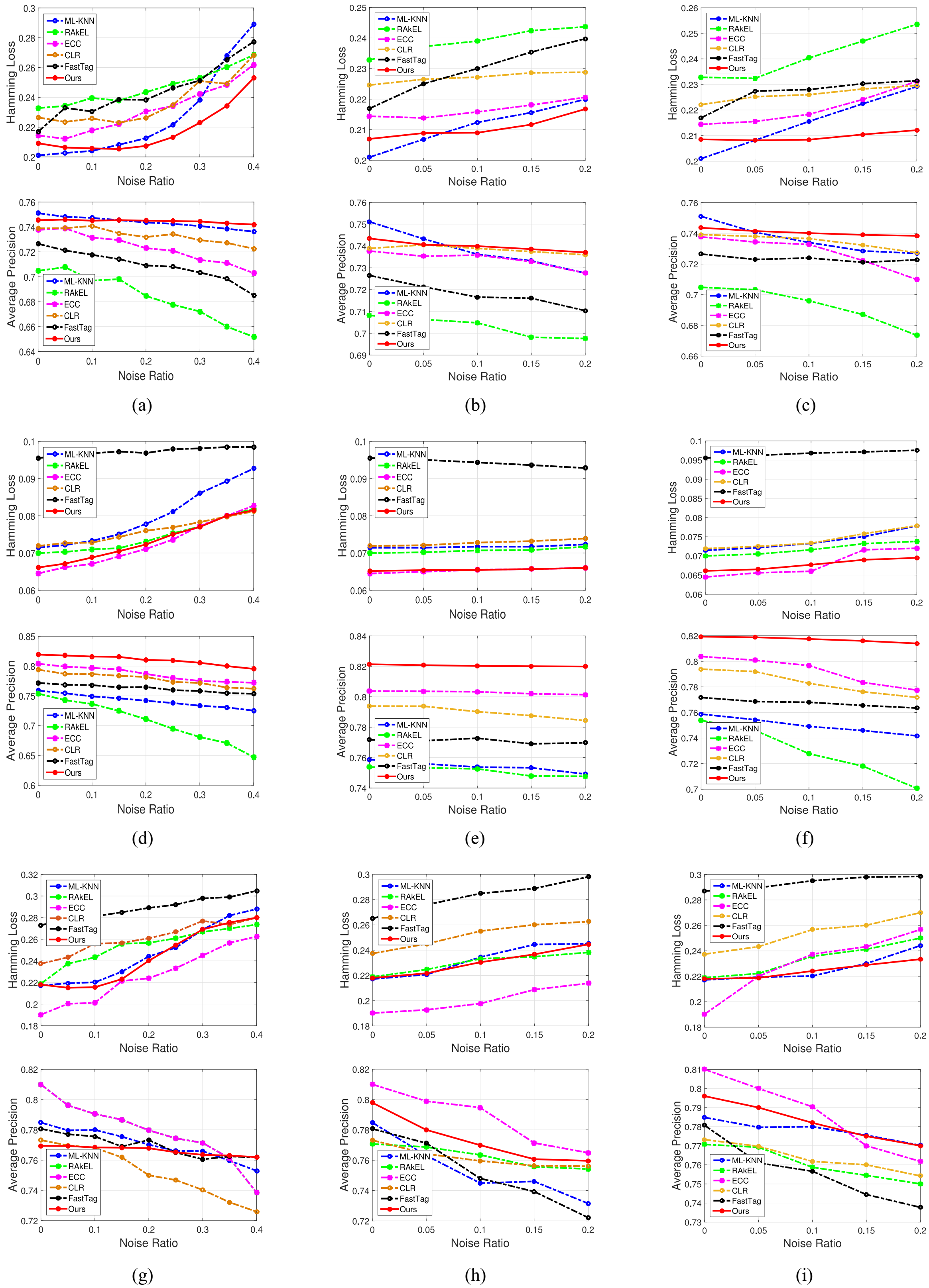


Fig. 3. Robustness experiments with different types of noise. The first to third columns correspond to label noise, feature noise, and hybrid noise, respectively. (a)–(c), (d)–(f), and (g)–(i) correspond to Yeast, TMC, and Emotions, respectively.

results in Tables II and III, several observations are obtained as follows.

1) Our method achieves the competitive performance on all datasets. For example, HNOML performs as the best one on CAL500 and Genbase in terms of all the five evaluation metrics.
2) The BR method, which is well known for multilabel classification, does not achieve promising performance.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fig. 4. Example classification results on Corel5k.

The possible reason is that directly decomposing multilabel task into independent binary problems neglects modeling interdependencies among labels.

3) Compared with BR, PS, and CC basically obtain much better performance since these methods take the label correlations into consideration.
4) Based on LP, RAkEL learns an ensemble of multiple LP classifiers, and the results in Tables II and V indicate that RAkEL improves substantially over LP with large margin.
5) It is observed that the nearest competitors are ML-kNN and CLR. Although their performances are slightly better than ours on a few datasets, our performances are more stable for different datasets. Specifically, HNOML outperforms ML-kNN and CLR on most datasets.

We also compare our algorithm with the algorithms which aim to handle label noise or feature noise. As shown in Tables IV and V, ECC improves BR by passing label correlation information along a chain of classifiers, which delivers a large improvement. Our method outperforms ECC and CLR on most datasets, although ECC adopts computationally expensive ensemble learning technique for robustness and CLR solves the problem by calibrating label ranking. Our hybrid noise-oriented algorithm also clearly outperforms the methods for label noise, that is, FastTag and MLR-GL, which validates the advantage of jointly addressing different types of noise. Note that MLR-GL is based on ranking which tends to correctly label the top-ranked classes, hence it usually performs well in terms of One-error.

2) Robustness Results: To evaluate the robustness of the proposed method for different types of noise, we demonstrate the performances of all methods with respect to different types of noise and degrees on Yeast, TMC, and Emotions as shown in Fig. 3. Specifically, three types of noise are used: 1) label noise; 2) feature noise; and 3) hybrid noise. To simulate label noise, we refer to the work [9] to randomly set positive labels (+1) to negative (−1) with the ratio of selected samples from 0% to 40% (0–0.4). For feature noise, similar to the work [66], we generate the error (noise) matrix **E** with a parameter ($\delta = 0.5$) to control the noise magnitude.

TABLE VI
COMPARING WITH METHODS EXPLICITLY DEALING WITH FEATURE NOISE, WHERE THE NOISE RATIO IS FROM 0.00 TO 0.20. THE PERFORMANCE IS EVALUATED IN TERMS OF HAMMING LOSS

Dataset	Method	0	0.05	0.10	0.15	0.20
Yeast	RMTL	.220	.223	.234	.240	.246
	RMTFL	.210	.215	.220	.228	.233
	Ours	**.206**	**.210**	**.212**	**.216**	**.225**
TMC	RMTL	.092	.096	.099	.100	.102
	RMTFL	.080	.082	.085	.090	.092
	Ours	**.066**	**.065**	**.065**	**.065**	**.064**
Emotions	RMTL	.250	.255	.261	.266	.270
	RMTFL	.230	.236	.242	.249	.256
	Ours	**.218**	**.221**	**.230**	**.236**	**.244**

Then, we add the generated error to the selected samples with the ratio from 0% to 20% (0–0.2). For hybrid noise, we directly combine the above two types of noise with the ratio from 0% to 20% (0–0.2). For different types of noise, although the performance of our method is slightly lower than ML-kNN at the beginning (low noise degree), much better performance is achieved for heavily noisy data. It is noteworthy that our method is rather stable even for the training data with hybrid noise, which empirically validates the robustness of our algorithm for complex noise.

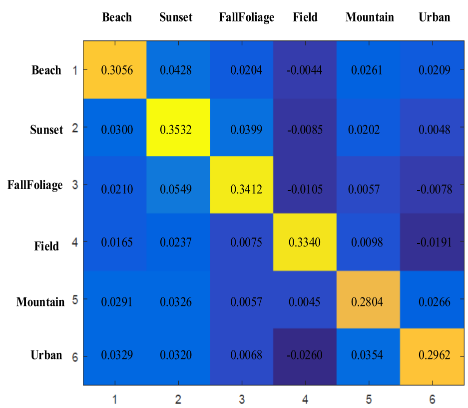
Furthermore, we also compare ours with robust multitask learning with least squares loss [24] and robust multitask feature learning [23], which explicitly deal with feature noise. We present the results with different degrees of feature noise on Tables VI and VII, which further validate the robustness of our method.

3) Results Visualization: As shown in Fig. 4, we present some example results by our algorithm on Corel5k, where the labels in green and in gray indicate the successfully and unsuccessfully predicted labels, respectively. For the failed examples, we find that the number of images containing the unsuccessfully predicted labels is usually very small. In addition, it is observed that these objects corresponding to unsuccessfully predicted labels are usually very small. We visualize the label enrichment matrix **B** on Scene and Emotions to investigate the discovered correlations among

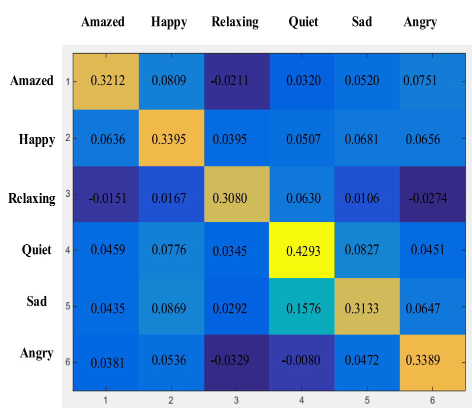
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VII
COMPARING WITH METHODS EXPLICITLY DEALING WITH FEATURE NOISE, WHERE THE NOISE RATIO IS FROM 0.00 TO 0.20. THE PERFORMANCE IS EVALUATED IN TERMS OF AVERAGE PRECISION

Dataset	Method	0	0.05	0.10	0.15	0.20
Yeast	RMTL	.752	.746	.741	.732	.728
	RMTFL	.758	.750	.746	.741	.740
	Ours	**.762**	**.756**	**.750**	**.747**	**.746**
TMC	RMTL	.786	.784	.779	.776	.774
	RMTFL	.798	.794	.790	.784	.782
	Ours	**.821**	**.819**	**.818**	**.816**	**.812**
Emotions	RMTL	.776	.772	.765	.762	.758
	RMTFL	.786	.782	.779	.774	.770
	Ours	**.797**	**.792**	**.789**	**.779**	**.772**



(a)

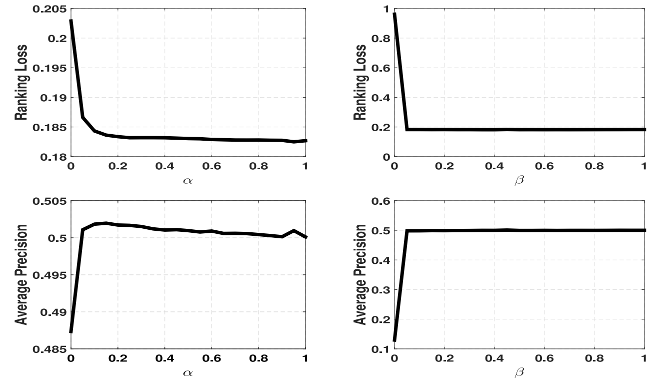


(b)

Fig. 5. Visualization of the label enrichment matrix **B**. (a) Scene. (b) Emotions.

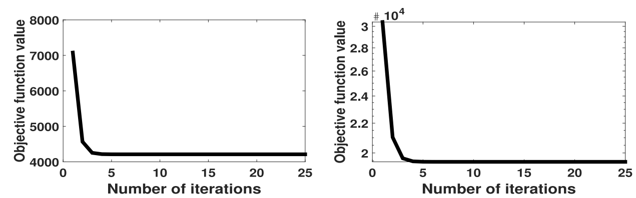
different labels. According to Fig. 5, it can be observed that the label enrichment matrix **B** reasonably encodes the correlations among different classes. For example, the label "filed" is positively correlated to "mountain" but not other labels, which is consistent with data. While for the Scene dataset, "quiet" and "sad" are highly positively correlated while "relaxing" is negatively correlated to "amazed" and "angry."

4) Parameter Tuning and Convergence Experiment: Fig. 6 shows the parameter tuning of the proposed algorithm. It is observed that the performance is relatively low with $\alpha = 0$ or $\beta = 0$, and the performance becomes much better and stable given relatively larger values. This implies the importance of



(a) (b)

Fig. 6. Parameter tuning. (a) Label embedding. (b) Label enrichment.



(a) (b)

Fig. 7. Convergence experiment. (a) Pascal. (b) Espgame.

TABLE VIII
SUMMARY OF THE FRIEDMAN STATISTICS F_F ($k = 13, M = 20$) AND THE CRITICAL VALUE IN TERMS OF EACH EVALUATION METRIC (k: NUMBER OF COMPARED ALGORITHMS; M: NUMBER OF DATASETS)

Evaluation metric	F_F	critical value ($\alpha = 0.05$)
Hamming Loss	36.1856	1.7948
Ranking Loss	40.7688	
One-error	17.7423	
Coverage	59.2508	
Average Precision	51.5588	

the preservation of locality in data and original label information. Fig. 7 gives the convergence experiment, where the results demonstrate that our method converges fast within a small number of iterations, which further empirically proves Theorem 1.

5) Statistical Comparisons of Multiple Classifiers: To compare multiple algorithms systematically, Friedman test [67] is employed in our experiments. Table VIII presents the Friedman statistics F_F and the corresponding critical values on each evaluation metric. According to the results in Table VIII, at the significance level $\alpha = 0.05$, the null hypothesis of "equal" performance among these algorithms over multiple datasets is obviously rejected in terms of each metric. Following the work [36], we also take the *post-hoc* test [67] to further evaluate the relative performance among these compared algorithms. Specifically, *Bonferroni–Dunn test* [67] is employed by treating our algorithm as the control one. The difference between the average ranks of our method and other compared algorithms is evaluated with the critical difference

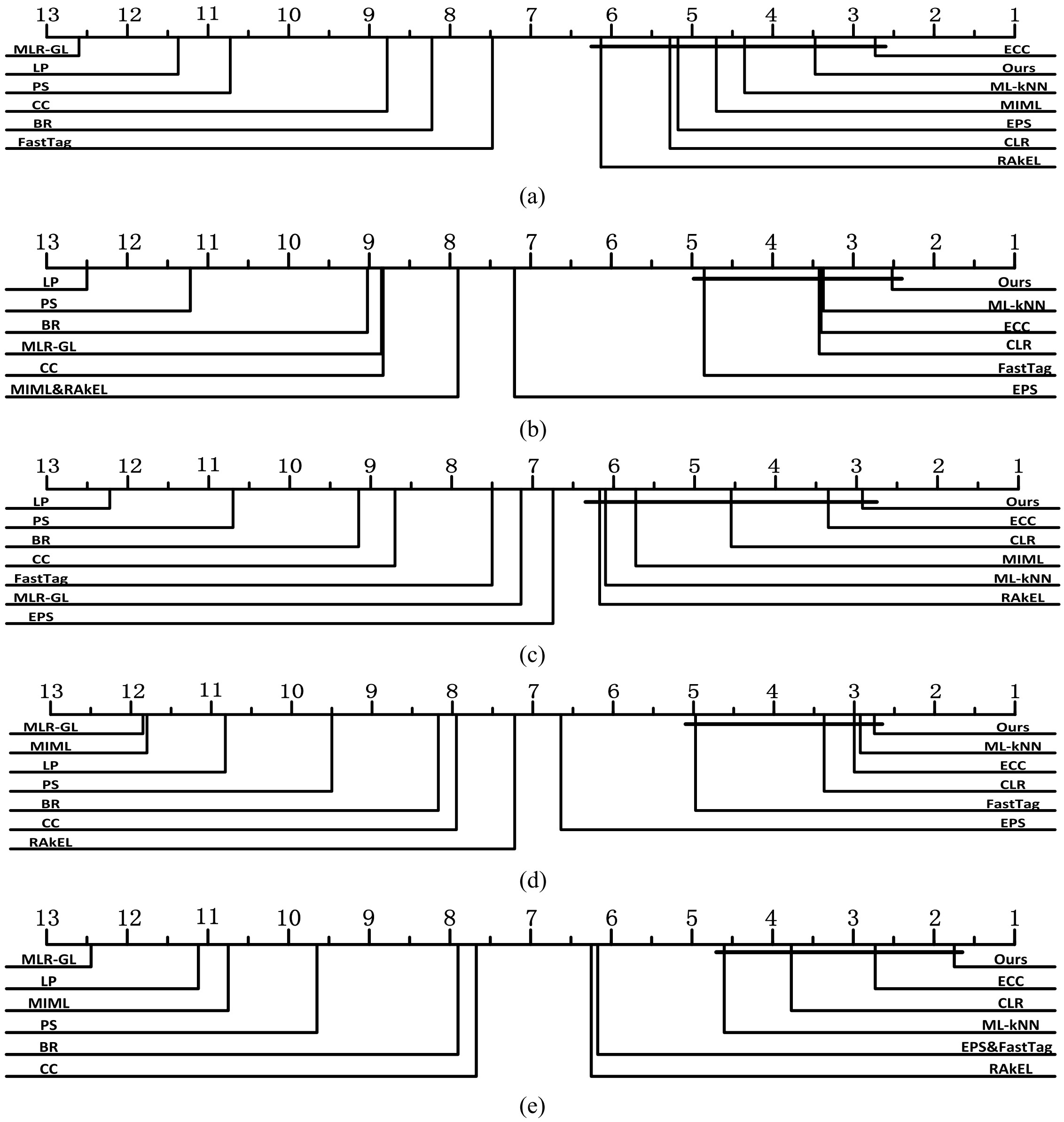


Fig. 8. Comparison of the proposed method (control algorithm) with other methods using the *Bonferroni–Dunn test*. (a) Hamming loss. (b) Ranking loss. (c) One-error. (d) Coverage. (e) Average precision.

(CD) defined as

$$\mathrm{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6M}} \tag{20}$$

where k and M are the number of compared algorithms and number of datasets, respectively. We have $q_\alpha = 2.865$ at the significance level $\alpha = 0.05$ and thus $\mathrm{CD} = 3.528$ ($k = 13$, $M = 20$). Accordingly, the performance between ours and other compared algorithms could be considered obviously different if their average ranks on all datasets differ by at least one CD.

For clarification, we illustrate the CD diagrams [67] on each evaluation metric in Fig. 8, where the average rank of each algorithm is marked on the axis. Basically, algorithms not connected with ours in the CD diagram are considered to have significantly different performance from the ours (control algorithm). Based on Fig. 8, our algorithm achieves significantly superior or at least comparable performance in terms of all evaluation metrics.

VI. Conclusion

In this paper, we consider multilabel classification under hybrid noisy data. To this end, we developed a robust multilabel learning model, called HNOML. Both the label noise and feature noise are addressed in a unified framework by jointly utilizing label enriching and structured sparsity. Our key idea lies in explicitly addressing feature noise and label noise (hybrid noise) in a unified framework, rather than only addressing missing labels as existing works. Empirical experiments clearly demonstrate that our method performs rather well with noisy training data, which validates the strong robustness of our method. In the future, more complex and more types of noise will be considered in our model.

Moreover, general relationships (e.g., nonlinearity) among labels will be explored.

REFERENCES

[1] G. Tsoumakas and I. Katakis, *Multi-Label Classification: An Overview*, Dept. Info., Aristotle Univ. Thessaloniki, Thessaloniki, Greece, 2006.

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

[3] E. Gibaja and S. Ventura, "Multi-label learning: A review of the state of the art and ongoing research," *Wiley Interdiscipl. Rev. Data Min. Knowl. Disc.*, vol. 4, no. 6, pp. 411–444, 2014.

[4] F. Herrera, F. Charte, A. J. Rivera, and M. J. D. Jesus, *Multilabel Classification: Problem Analysis, Metrics and Techniques*. New York, NY, USA: Springer, 2016.

[5] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Comput. Surveys*, vol. 47, no. 3, p. 52, 2015.

[6] C. Zhang *et al.*, "Latent semantic aware multi-view multi-label classification," in *Proc. AAAI*, 2018, pp. 4414–4421.

[7] H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale multi-label learning with missing labels," in *Proc. ICML*, 2014, pp. 593–601.

[8] Q. Wang, B. Shen, S. Wang, L. Li, and L. Si, "Binary codes embedding for fast image tagging with incomplete labels," in *Proc. ECCV*, 2014, pp. 425–439.

[9] B. Wu, S. Lyu, and B. Ghanem, "Constrained submodular minimization for missing labels and class imbalance in multi-label learning," in *Proc. AAAI*, 2016, pp. 2229–2236.

[10] R. J. Hickey, "Noise modelling and evaluating learning from examples," *Artif. Intell.*, vol. 82, nos. 1–2, pp. 157–179, 1996.

[11] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[12] Q. Wang, L. Si, and D. Zhang, "Learning to hash with partial tags: Exploring correlation between tags and hashing bits for large scale image retrieval," in *Proc. ECCV*, 2014, pp. 378–392.

[13] L. Jing, L. Yang, J. Yu, and M. K. Ng, "Semi-supervised low-rank mapping learning for multi-label classification," in *Proc. CVPR*, 2015, pp. 1483–1491.

[14] B. Wu, S. Lyu, B.-G. Hu, and Q. Ji, "Multi-label learning with missing labels for image annotation and facial action unit recognition," *Pattern Recognit.*, vol. 48, no. 7, pp. 2279–2289, 2015.

[15] A. Akbarnejad and M. S. Baghshah, "A probabilistic multi-label classifier with missing and noisy labels handling capability," *Pattern Recognit. Lett.*, vol. 89, pp. 18–24, Apr. 2017.

[16] Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou, "Multi-label learning with weak label," in *Proc. AAAI*, 2010, pp. 593–598.

[17] S. S. Bucak, R. Jin, and A. K. Jain, "Multi-label learning with incomplete class assignments," in *Proc. CVPR*, 2011, pp. 2801–2808.

[18] L. V. D. Maaten, M. Chen, S. Tyree, and K. Q. Weinberger, "Learning with marginalized corrupted features," in *Proc. ICML*, 2013, pp. 410–418.

[19] J. Zhuo, J. Zhu, and B. Zhang, "Adaptive dropout rates for learning with corrupted features," in *Proc. IJCAI*, 2015, pp. 4126–4132.

[20] Y. Zhang and Z.-H. Zhou, "Multilabel dimensionality reduction via dependence maximization," *ACM Trans. Knowl. Disc. Data*, vol. 4, no. 3, pp. 1–14, 2010.

[21] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive Bayes classification," *Inf. Sci.*, vol. 179, no. 19, pp. 3218–3229, 2009.

[22] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. NIPS*, 2010, pp. 1813–1821.

[23] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proc. KDD*, 2012, pp. 895–903.

[24] J. Chen, J. Zhou, and J. Ye, "Integrating low-rank and group-sparse structures for robust multi-task learning," in *Proc. ACM SIGKDD*, 2011, pp. 42–50.

[25] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM J. Optim.*, vol. 21, no. 1, pp. 57–81, 2011.

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

[27] A. Clare and R. D. King, *Knowledge Discovery in Multi-Label Phenotype Data* (LNCS-2168). Heidelberg, Germany: Springer, 2001, pp. 42–53.

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

[29] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. NIPS*, 2001, pp. 681–687.

[30] N. Ghamrawi and A. Mccallum, "Collective multi-label classification," in *Proc. CIKM*, 2005, pp. 195–200.

[31] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach. Learn.*, vol. 73, no. 2, pp. 133–153, 2008.

[32] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random *k*-labelsets for multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

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

[34] R. Yan, J. Tesic, and J. R. Smith, "Model-shared subspace boosting for multi-label classification," in *Proc. ACM SIGKDD*, 2007, pp. 834–843.

[35] Y.-K. Li, M.-L. Zhang, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," in *Proc. ICDM*, 2016, pp. 251–260.

[36] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, Jan. 2015.

[37] P. Hou, X. Geng, and M.-L. Zhang, "Multi-label manifold learning," in *Proc. AAAI*, 2016, pp. 1680–1686.

[38] B. Wu, Z. Liu, S. Wang, B.-G. Hu, and Q. Ji, "Multi-label learning with missing labels," in *Proc. ICPR*, 2014, pp. 1964–1968.

[39] B. Wu, S. Lyu, and B. Ghanem, "ML-MG: Multi-label learning with missing labels using a mixed graph," in *Proc. ICCV*, 2015, pp. 4157–4165.

[40] L. Xu, Z. Wang, Z. Shen, Y. Wang, and E. Chen, "Learning low-rank label correlations for multi-label classification with missing labels," in *Proc. ICDM*, 2014, pp. 1067–1072.

[41] S. Ji and J. Ye, "Linear dimensionality reduction for multi-label classification," in *Proc. IJCAI*, 2009, pp. 1077–1082.

[42] Y.-N. Chen and H.-T. Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Proc. NIPS*, 2012, pp. 1529–1537.

[43] S. Ji, L. Tang, S. Yu, and J. Ye, "Extracting shared subspace for multi-label classification," in *Proc. ACM SIGKDD*, 2008, pp. 381–389.

[44] X. Chang, F. Nie, Y. Yang, and H. Huang, "A convex formulation for semi-supervised multi-label feature selection," in *Proc. AAAI*, 2014, pp. 1171–1177.

[45] L. Jian, J. Li, K. Shu, and H. Liu, "Multi-label informed feature selection," in *Proc. IJCAI*, 2016, pp. 1627–1633.

[46] Q. Gu, Z. Li, and J. Han, "Correlated multi-label feature selection," in *Proc. CIKM*, 2011, pp. 1087–1096.

[47] O. Reyes, C. Morell, and S. Ventura, "Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context," *Neurocomputing*, vol. 161, pp. 168–182, Aug. 2015.

[48] O. Reyes, C. Morell, and S. Ventura, "Evolutionary feature weighting to improve the performance of multi-label lazy algorithms," *Integr. Comput.-Aided Eng.*, vol. 21, no. 4, pp. 339–354, 2014.

[49] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *Proc. ICML*, 2009, pp. 433–440.

[50] L. Meier, S. Van De Geer, and P. Bühlmann, "The group lasso for logistic regression," *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)*, vol. 70, no. 1, pp. 53–71, 2008.

[51] D. Belanger and A. Mccallum, "Structured prediction energy networks," in *Proc. ICML*, 2016, pp. 983–992.

[52] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2018.2877660.

[53] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. ICCV*, 2015, pp. 1582–1590.

[54] Y. Yang *et al.*, "Latent max-margin multitask learning with skelets for 3-D action recognition," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 439–448, Feb. 2017.

[55] C. Zhang *et al.*, "Infant brain development prediction with latent partial multi-view representation learning," *IEEE Trans. Med. Imag.*, to be published, doi: 10.1109/TMI.2018.2874964.

[56] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, “Semantic annotation and retrieval of music and sound effects,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 2, pp. 467–476, Feb. 2008.

[57] J. P. Pestian *et al.*, “A shared task involving multi-label classification of clinical free text,” in *Proc. Workshop BioNLP*, 2007, pp. 97–104.

[58] S. Diplaris, G. Tsoumakas, P. A. Mitkas, and I. Vlahavas, “Protein classification with multiple algorithms,” in *Proc. Panhellenic Conf. Inf.*, 2005, pp. 448–456.

[59] I. Katakis, G. Tsoumakas, and I. Vlahavas, “Multilabel text classification for automated tag suggestion,” in *Proc. ECMLPKDD Disc. Challenge*, 2008, pp. 1–9.

[60] F. Briggs *et al.*, “The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment,” in *Proc. IEEE Int. Workshop MLSP*, 2013, pp. 1–8.

[61] K. Barnard *et al.*, “Matching words and pictures,” *J. Mach. Learn. Res.*, vol. 3, pp. 1107–1135, Feb. 2003.

[62] N. Ueda and K. Saito, “Parametric mixture models for multi-labeled text,” in *Proc. NIPS*, 2002, pp. 737–744.

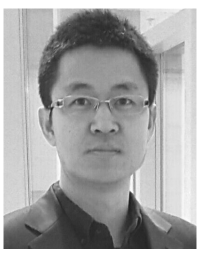
[63] R. E. Schapire and Y. Singer, “BoosTexter: A boosting-based system for text categorization,” *Mach. Learn.*, vol. 39, nos. 2–3, pp. 135–168, 2000.

[64] J. Read, B. Pfahringer, and G. Holmes, “Multi-label classification using ensembles of pruned sets,” in *Proc. ICDM*, 2008, pp. 995–1000.

[65] M. Chen, A. X. Zheng, and K. Q. Weinberger, “Fast image tagging,” in *Proc. ICML*, 2013, pp. 1274–1282.

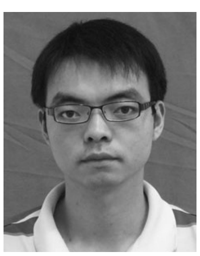
[66] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, “Latent multi-view subspace clustering,” in *Proc. CVPR*, 2017, pp. 4333–4341.

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



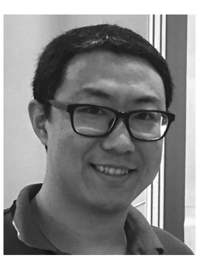
Changqing Zhang (M’18) received the B.S. and M.S. degrees in computer science from the College of Computer, Sichuan University, Chengdu, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2016.

He is an Assistant Professor with the College of Intelligence and Computing, Tianjin University. He has been a Postdoctoral Research Fellow with the Department of Radiology and BRIC, School of Medicine, University of North Carolina, Chapel Hill, NC, USA. His current research interests include machine learning, computer vision, and medical image analysis.



Ziwei Yu received the B.S. degree in computer science and technology from Northeastern University, Qinhuangdao, China, in 2015 and the M.Sc. degree in computer science and technology from Tianjin University, Tianjin, China, in 2018.

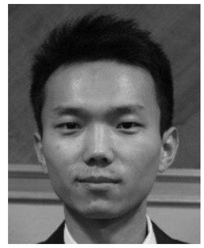
His current research interests include multilabel learning and multiview learning.



Huazhu Fu (SM’18) received the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2013.

After graduation, he spent about two years with Nanyang Technological University, Singapore, as a Research Fellow. From 2015 to 2018, he was a Research Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. He is currently a Senior Scientist with the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE. His current research interests include computer vision, image processing, and medical image analysis.

Dr. Fu is an Associate Editor of IEEE ACCESS and *BMC Medical Imaging*.



Pengfei Zhu (M’15) received the B.S. and M.S. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2009 and 2011, respectively, and the Ph.D. degree from Hong Kong Polytechnic University, Hong Kong, in 2015.

He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has published over 50 papers in ICCV, CVPR, ECCV, AAAI, IJCAI, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and the IEEE TRANSACTIONS ON IMAGE PROCESSING. His current research interests include machine learning and computer vision.

Dr. Zhu is the Local Arrangement Chair of IJCRS 2015, CCML 2017, and CCCV 2017.

Lei Chen received the master’s degree in computer software and theory from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2005 and the Ph.D. degree in communication and information systems from the Nanjing University of Posts and Telecommunications, Nanjing, in 2014.

He is currently an Associate Professor with the School of Computer Science, Nanjing University of Posts and Telecommunications. He was a Visiting Researcher with the University of North Carolina, Chapel Hill, NC, USA, from 2016 to 2017. His current research interests include machine learning, pattern recognition, and medical image analysis.

Qinghua Hu (SM’13) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively.

He was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently a Full Professor with Tianjin University, Tianjin, China. He has authored over 150 journal and conference papers in the areas of granular computing-based machine learning, reasoning with uncertainty, pattern recognition, and fault diagnosis. His current research interests include multimodality learning, metric learning, uncertainty modeling, and reasoning with fuzzy sets, rough sets, and probability theory.

Prof. Hu was the Program Committee Co-Chair of the International Conference on Rough Sets and Current Trends in Computing in 2010, the Chinese Rough Set and Soft Computing Society in 2012 and 2014, and the International Conference on Rough Sets and Knowledge Technology and the International Conference on Machine Learning and Cybernetics in 2014, and the General Co-Chair of IJCRS 2015.