IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 25, 2020, accepted May 30, 2020, date of publication June 10, 2020, date of current version June 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001274

Tri-Structured-Sparsity Induced Joint Feature Selection and Classification for Hybrid Noise Resilient Multilabel Learning

LEI XU[1], CHUANCHENG SONG[1], (Student Member, IEEE), AND LEI CHEN[2,3], (Member, IEEE)

[1]Bell Honors School, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[2]School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[3]Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Corresponding author: Lei Chen (chenlei@njupt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61872190.

ABSTRACT Multilabel learning handles the problem that instances are associated with multiple labels. In practical applications, multilabel learning often suffers from imperfect training data. Typically, labels may be noisy or features may be corrupted, or both. Most existing multilabel learning models only consider either label noise or feature noise. Theoretically, ignoring any kind of noise in the learning process may lead to an unreasonable model, and thus affect the multilabel learning performance. In this paper, we propose a robust multilabel learning model, Tri-structured-Sparsity induced Joint Feature Selection and Classification (TriS-JFSC), to handle the data with hybrid noise. Specifically, the proposed TriS-JFSC model employs the tri-structured-sparsity regularization bridged with a label enhancement matrix to simultaneously smooth the feature and label noise, and embed a feature selection scheme that can simultaneously learn label-shared features and label-specific ones to boost the multilabel learning performance. Furthermore, by employing Alternating Direction Method of Multipliers (ADMM) method, a simple but efficient optimization algorithm is designed to solve the proposed TriS-JFSC model. Finally, the extensive experiments performed on several benchmark datasets demonstrate that our TriS-JFSC model outperforms other state-of-the-art learning methods.

INDEX TERMS Multilabel learning, hybrid noise, tri-structured-sparsity, label enhancement, feature selection.

I. INTRODUCTION

Multilabel learning deals with the problem that each instance is assigned with multiple labels. For example, a news document could cover multiple topics simultaneously, and different objects may appear in a picture at the same time. Due to its practical significance, multilabel learning has a wide range of applications in real world, such as bioinformatics [1], [2], clinical data analysis [3], image recognition [4], data mining [5], tag recommendation [6], information retrieval [7]. However, most of proposed multilabel learning methods lack consideration of data noise, which leads to degraded performance in practical applications when encountering unsatisfactory data. Thus, multilabel learning is still a challenging problem.

The fact that the data contains noise is very common in practical applications. Ignoring this problem will reduce training performance and lead to an unreasonable model. On the one hand, there may be incorrect or incomplete labels in data, and some methods have been proposed to address this problem [8]–[10]. On the other hand, since observed values tend to be affected, features may also contain noise. Given this consideration, some methods have been proposed to deal with feature noise [11]–[13].

Although different types of noise have been considered separately in existing methods, the real-world data is likely to contain both feature noise and label noise. To this end, Zhang *et al.* [14] proposed the hybrid noise-oriented multilabel learning (HNOML) model to address the hybrid noise

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

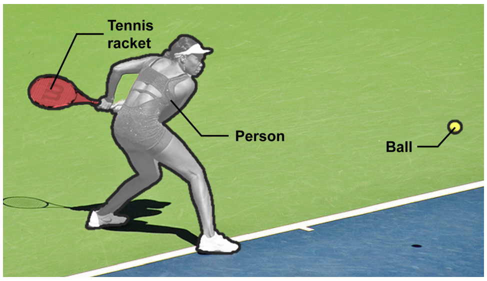


FIGURE 1. Each label is determined by a small subset of specific features.

problems for the imperfect training data. By employing the bi-structured-sparsity regularization and ℓ_2-norm induced graph Laplacian regularization, the HNOML model achieves competitive performance on many datasets compared with existing state-of-the-art methods. However, the HNOML model still can be improved in at least two aspects: Firstly, The ℓ_2-norm induced graph Laplacian regularization can well model the consistence between normal feature vector and enhanced label vector, but ignore the inconsistency between noisy feature vector and enhanced label vector. Secondly, the HNOML model indiscriminately uses all the features for multilabel learning. In fact, the features that determine each label for a sample are not the same, which means that some features are redundant. Taking image annotation as an example, Fig. 1[1] is tagged with three labels: Tennis racket, Person and Ball, which are dependent on the features in the red, grey, and yellow areas, respectively. Feature selection can help us find the most critical information of each label and boost the performance of classification. Nie *et al.* [15] designed a feature selection method to learn the feature subset shared by all labels. However, from Fig. 1, we can see that in addition to label-shared features, each label also depends on some specific features of its own, which is worthy of consideration.

To address these problems, we design a robust multilabel learning method called Tri-structured-Sparsity induced Joint Feature Selection and Classification (TriS-JFSC). Specifically, we employ a Structured-Sparsity induced Graph Trend Filtering (SS-GTF) regularization, together with Structured-Sparsity induced Label Fidelity Penalty (SS-LFP), to smooth the sample-specific label noise, and simultaneously learn a label enhancement matrix. Compared with ℓ_2-norm induced Laplacian regularization, SS-GTF not only utilizes the local correlation between samples, but also considers the inconsistency between the noisy feature vector and enhanced label vector. After that, we imposed the structured sparsity on the prediction loss to tolerate sample-specific noise in the feature matrix. Furthermore, we introduce an adaptive feature selection mechanism to boost the multilabel learning performance, which can extract the most discriminative features for each label. Different from Nie *et al.* [15], we employ the ℓ_1-norm regularization and structured sparsity regularization to simultaneously learn label-specific and label-shared features.

Our main contributions are summarized as follows:

- We propose a robust Tri-structured-Sparsity induced Joint Feature Selection and Classification (TriS-JFSC) model to address the multilabel learning problem on the imperfect training data (Section III. A) (Section III. D). Extensive experimental results on several benchmark datasets demonstrate that our proposed TriS-JFSC model can simultaneously tolerate the feature noise and label noise of the training samples (Section V).
- We design a structured-sparsity induced graph trend filtering (SS-GTF) regularization scheme, together with structured-sparsity induced label fidelity penalty (SS-LFP), to smooth the label noise, and thus obtain a label enhancement matrix to guide the learning of multilabel classifier (Section III. B).
- Different from existing multilabel learning methods, we embed a novel feature selection scheme into the proposed model to simultaneously select the label-shared features and label-specific features, and thus boost the model's multilabel learning performance (Section III. C).
- Based on the popular Alternating Direction Method of Multipliers (ADMM) method, we develop a simple but efficient optimization algorithm to solve the proposed TriS-JFSC model (Section IV).

II. RELATED WORK

A. MULTILABEL CLASSIFICATION

According to the manner of dealing with the label correlations, existing multilabel classification methods can be divided into three types, i.e., *first-order*, *second-order*, and *high-order* algorithms [16]. *First-order* methods address the multilabel problem in a label-by-label style, such as decomposing it into multiple independent binary classification problems. Typical methods are binary relevance(BR) [4], multi-label learning with label specific features (LIFT) [17], multi-label *k*-nearest neighbor (ML-kNN) [18] and sparse weighted instance-based multilabel learning(SWIM) [19]. Although this strategy is easy to implement, the ignorance of label correlations will result in its low generalization performance. *Second-order* methods cope with multilabel learning by exploiting the pairwise relationship between label pairs, such as ranking support vector machine (Rank-SVM) [20], learning label-specific features (LLSF) for multilabel classification [21], and backpropagation for multilabel learning [2]. Although these methods have achieved good performance, the real-world relationship may be more complex and has correlations beyond the second order. For this reason, *high-order* methods are proposed to establish more complex relationships, such as label powerset (LP) [22], random k labelsets (RAkEL) [23], ensembles of pruned sets (EPS) [24], and classifier chain (CC) [25]. However, their computational complexity is too high to deal with large-scale learning problems.

[1] Photo credits to http://cocodataset.org/

B. FEATURE SELECTION

Feature selection plays an important role in machine learning, as it can filter out irrelevant features for each learning task and boost the learning performance. Traditional feature selection methods can be roughly divided into three categories: *filter* methods, *wrapper* methods and *embedded* methods. *Filter* methods first select the features of the dataset and then train the classifier [26]–[28]. The feature selection process is independent of the training process. *Wrapper* methods use the heuristic search strategy to determine some subsets of features and then select features according to their corresponding performance on off-the-shelf classifiers [29]–[32]. *Embedded* methods integrate the feature selection into the classifier training [1], [33]–[35]. The aforementioned three types of methods can be directly applied to multilabel learning problems and some methods use them to consider the common features shared by all labels, such as robust feature selection (RFS) [15], subfeature uncovering with sparsity (SFUS) [36], and lasso [37]. However, each label may also depend on some specific features. Given this consideration, some methods are proposed to select label-specific features, such as LIFT [17], learning label-specific features for multilabel classification (LLSF) [21] and joint feature selection and classification (JFSC) [38].

III. PROPOSED MODEL

A. PRELIMINARIES

In this paper, **MATRICES** are written as boldface uppercase letters and **vectors** are written as boldface lowercase letters. For an arbitrary matrix $\mathbf{Z} \in \mathbb{R}^{m\times n}$, $\mathbf{z}_i$ denotes the i-th row of $\mathbf{Z}$ and $z_{i,j}$ denotes the element in the i-th row and j-th column of $\mathbf{Z}$. The Frobenius norm of $\mathbf{Z}$ is defined as $\|\mathbf{Z}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} z_{i,j}^2}$. And the $\ell_{2,1}$-norm, which is also called as the structured sparsity, is defined as $\|\mathbf{Z}\|_{2,1} = \sum_{i=1}^{m}\sqrt{\sum_{j=1}^{n} z_{i,j}^2}$.

Let $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{0, 1\}^c$ denote feature space and label space, where d and c are the number of features and number of labels (for each sample), respectively. Our goal is to learn a mapping $f : \mathcal{X} \to \mathcal{Y}$, which can accurately predict the label vector for each sample. We define the training feature matrix and label matrix as $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_n] \in \mathbb{R}^{n\times d}$ and $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_n] \in \{0, 1\}^{n\times c}$, where n is the number of samples. Assuming that feature space and label space are linearly related, and we denote the mapping f as a predictor matrix $\mathbf{Q} \in \mathbb{R}^{d\times c}$. Then for each training data pair $\{\mathbf{x}_i, \mathbf{y}_i\}$, the following equation should be satisfied:

$$\mathbf{y}_i = \mathbf{x}_i\mathbf{Q} + \mathbf{b} = \left[\mathbf{x}_i, 1\right]\begin{bmatrix}\mathbf{Q}\\ \mathbf{b}\end{bmatrix}, \quad i = 1, 2, \ldots, n \tag{1}$$

where $\mathbf{b} \in \mathbb{R}^{1\times c}$ is the bias. For simplicity, here we abuse $\mathbf{X} = [\mathbf{X}, \mathbf{1}]$ and $\mathbf{Q} = [\mathbf{Q}; \mathbf{b}]$ with $\mathbf{1} \in \mathbb{R}^{n\times 1}$ being the all-ones column vector, then we can present (1) in a more compact form as

$$\mathbf{Y} = \mathbf{X}\mathbf{Q}. \tag{2}$$

Based on $\mathbf{Q}$, we can predict the label vector $\tilde{\mathbf{y}}$ for each unseen instance $\tilde{\mathbf{x}}$ by calculating

$$\tilde{\mathbf{y}} = t(\left[\tilde{\mathbf{x}}, 1\right]\mathbf{Q}), \tag{3}$$

where $t(\cdot)$ is a thresholding function defined as

$$t(u) = \begin{cases} 0, & u < 0.5 \\ 1, & u \geq 0.5. \end{cases} \tag{4}$$

To obtain a reasonable $\mathbf{Q}$, the objective function is often designed as

$$\min_{\mathbf{Q}} \mathcal{L}(\mathbf{XQ}, \mathbf{Y}) + \lambda\mathcal{R}(\mathbf{Q}) \tag{5}$$

where λ is the tradeoff parameter, $\mathcal{L}(\cdot)$ is the loss function, and $\mathcal{R}(\cdot)$ is the regularization term which is used to design a more reasonable model $\mathbf{Q}$ under different assumptions.

In this paper, we focus on the multilabel learning problem that training data contain hybrid noise. To address this problem, we employ the tri-structured-sparsity regularization bridged with a label enhancement matrix to simultaneously smooth the feature and label noise. Specifically, we exploit the sample correlations with SS-GTF and utilize SS-LFP to smooth the label noise and obtain a label enhancement matrix $\widehat{\mathbf{Y}}$. In this way, the noisy labelling can be improved by substituting the origin labelling $\mathbf{Y}$ with the label enhancement matrix $\widehat{\mathbf{Y}}$. Benefiting from the label enhancement matrix, we further impose structured sparsity on the prediction loss to tolerate feature noise. Moreover, to improve the learning performance, we introduce an adaptive feature selection scheme to learn label-shared features and label-specific features, respectively. Therefore, we generalize our multilabel learning model as follows:

$$\begin{aligned}\min_{\mathbf{M},\mathbf{W},\mathbf{Q},\widehat{\mathbf{Y}}} \ & \mathcal{L}_1(\mathbf{XQ}, \widehat{\mathbf{Y}}) + \mu_1\mathcal{R}_1(\mathbf{W}) + \mu_2\mathcal{R}_2(\mathbf{M}) \\ & + \lambda_1\mathcal{L}_2(\widehat{\mathbf{Y}}) + \lambda_2\mathcal{S}(\widehat{\mathbf{Y}}) \quad s.t. \mathbf{Q} = \mathbf{M} + \mathbf{W}, \end{aligned} \tag{6}$$

where μ_1, μ_2, λ_1, and λ_2 are tradeoff parameters. $\mathcal{L}_1(\cdot)$ is the Structured-Sparsity induced prediction Loss function (SS-Loss). $\mathcal{R}_1(\cdot)$ and $\mathcal{R}_2(\cdot)$ are designed to select label-shared features and label-specific features, respectively. $\mathcal{L}_2(\cdot)$ is Structured-Sparsity induced Label Fidelity Penalty (SS-LFP). $\mathcal{S}(\cdot)$ is Structured-Sparsity induced Graph Trend Filtering (SS-GTF) which is utilized to exploit the local correlation between samples to learn the label enhancement matrix.

B. LABEL ENHANCEMENT MATRIX LEARNING

Recall that we try to reconstruct an ideal label matrix according to the local correlation lying in feature space, that is, closely related instances tend to share a common set of labels. To this end, we first employ the graph trend filtering (GTF) regularization to encourage highly correlated instances to have similar labels. Specifically, we construct a Graph $\mathcal{G}$ as a prior knowledge over the feature matrix to show the pairwise relationships between instances. Each node in $\mathcal{G}$ represents an instance, and the edge denotes the pairwise correlation between instances. We define $\mathbf{A} \in \mathbb{R}^{n\times n}$ as the similarity matrix and $a_{i,j}$ denotes the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Specifically, $a_{i,j}$ can be calculated by

$$a_{i,j} = \begin{cases} exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\xi^2}), & i \neq j \\ 0, & i = j \end{cases} \tag{7}$$

where ξ is the Gaussian kernel width. Taking $\mathbf{x}_i$ as an example, we define the top k largest values in $\mathbf{a}_i$ as the strong correlations with $\mathbf{x}_i$, then $\mathbf{x}_i$ will be connected to the corresponding nodes in $\mathcal{G}$.

Given the graph $\mathcal{G}$, it is reasonable to assume that if two instances are connected with an edge, they should have similar output. In view of this, we adopt the GTF regularization $\|\mathbf{SY}\|_1$ to keep this consistency between labels and features, where $\mathbf{S} \in \{-1, 0, 1\}^{\varepsilon \times n}$ is the incidence matrix of the graph G with ε being the number of edges. Specifically, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are linked by the p-th edge, then $\mathbf{S}$ has the p-th row:

$$\mathbf{s}_p = (0, \ldots, \underset{\substack{\uparrow \\ i}}{-1}, \ldots, \underset{\substack{\uparrow \\ j}}{1}, \ldots, 0). \tag{8}$$

The aforementioned assumption only considers normal data, whereas this assumption does not apply to feature noise. In view of this, we impose the structured sparsity on the GTF to filter this inconsistency. The structured sparsity has the property of row-wise sparsity, which enables GTF to alleviate the inconsistencies in feature noise. Therefore, the SS-GTF term is defined as

$$\mathcal{S}(\widehat{\mathbf{Y}}) = \|\mathbf{S}\widehat{\mathbf{Y}}\|_{2,1}. \tag{9}$$

Moreover, the obtained label enhancement matrix should be basically consistent with the original label matrix. Hence, we define the label fidelity penalty to measure the difference between $\widehat{\mathbf{Y}}$ and $\mathbf{Y}$. Considering the label noise in a few samples, we also impose the structured sparsity to address this problem. Consequently, SS-LFP is defined as

$$\mathcal{L}_2(\widehat{\mathbf{Y}}) = \|\mathbf{Y} - \widehat{\mathbf{Y}}\|_{2,1}. \tag{10}$$

After that, $\widehat{\mathbf{Y}}$ will be introduced in the prediction to substitute the original label matrix, so as to learn a more accurate classifier.

C. ROBUST FEATURE SELECTION

To address the feature noise in data, we constraint the prediction loss with structured sparsity and define SS-Loss as

$$\mathcal{L}_1(\mathbf{XQ}, \widehat{\mathbf{Y}}) = \|\mathbf{XQ} - \widehat{\mathbf{Y}}\|_{2,1}. \tag{11}$$

As discussed in the introduction, the labels may only be determined by a subsect of features. On the one hand, there are some features shared by all labels. On the other hand, each label may also depend on some specific features of its own. Hence, we decompose $\mathbf{Q}$ into $\mathbf{W}$ and $\mathbf{M}$. $\mathbf{W}$ is imposed with structured sparsity to learn label-shared features, and $\mathbf{M}$ is constrained by ℓ_1-norm to learn label-specific features. Then $\mathcal{R}_1(\mathbf{W})$ and $\mathcal{R}_2(\mathbf{M})$ are defined as

$$\mathcal{R}_1(\mathbf{W}) = \|\mathbf{W}\|_{2,1} \tag{12}$$

$$\mathcal{R}_2(\mathbf{M}) = \|\mathbf{M}\|_1. \tag{13}$$

D. PROPOSED TriS-JFSC MODEL

As a result, we propose our TriS-JFSC model as follows:

$$\min_{\mathbf{M},\mathbf{W},\mathbf{Q},\widehat{\mathbf{Y}}} \underbrace{\|\mathbf{XQ} - \widehat{\mathbf{Y}}\|_{2,1}}_{\text{SS-Loss}} + \underbrace{\mu_1\|\mathbf{W}\|_{2,1}}_{\substack{\text{Label-Shared} \\ \text{Feature Selection}}} + \underbrace{\mu_2\|\mathbf{M}\|_1}_{\substack{\text{Label-Specific} \\ \text{Feature Selection}}}$$
$$+ \underbrace{\lambda_1\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_{2,1}}_{\text{SS-LFP}} + \underbrace{\lambda_2\|\mathbf{S}\widehat{\mathbf{Y}}\|_{2,1}}_{\text{SS-GTF}} \; s.t.\; \mathbf{Q} = \mathbf{M} + \mathbf{W} \tag{14}$$

where μ_1, μ_2, λ_1, and λ_2 are tradeoff parameters. The first three items are utilized for feature-noise-robust feature selection which can simultaneously tolerate sample-specific feature noise and select label-specific and label-shared features. SS-LFP and SS-GTF are employed for label-noise-robust label enhancement matrix learning, and the obtained label enhancement matrix will guide the learning of multilabel classifier. The illustration of our model is presented in Fig. 2.

IV. OPTIMIZING TriS-JFSC VIA ADMM

A. OPTIMIZATION

Alternating Direction Method of Multipliers (ADMM) is a popular algorithm for solving convex optimization problems. In this paper, we employ ADMM alogrithm to solve the optimization problem in (14). To use ADMM, we first convert (14) into its augmented Lagrangian form:

$$\begin{aligned} &L(\mathbf{M}, \mathbf{Q}, \mathbf{W}, \widehat{\mathbf{Y}}, \mathbf{\Omega}) \\ &= \|\mathbf{XQ} - \widehat{\mathbf{Y}}\|_{2,1} + \mu_1\|\mathbf{W}\|_{2,1} + \mu_2\|\mathbf{M}\|_1 + \lambda_1\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_{2,1} \\ &\quad + \lambda_2\|\mathbf{S}\widehat{\mathbf{Y}}\|_{2,1} + \langle\mathbf{\Omega}, \mathbf{M}+\mathbf{W}-\mathbf{Q}\rangle + \frac{\rho}{2}\|\mathbf{M}+\mathbf{W}-\mathbf{Q}\|_F^2 \end{aligned} \tag{15}$$

where $\mathbf{\Omega} \in \mathbb{R}^{(d+1)\times c}$ is the Lagrange multiplier, $\rho > 0$ is the adaptive penalty parameter and $\langle\cdot,\cdot\rangle$ represents the Frobenius inner product. Let $\mathbf{U} = \mathbf{\Omega}/\rho$, then the optimization of (15) can be transformed into the problem of minimizing the following function:

$$\begin{aligned} &\hat{L}(\mathbf{M}, \mathbf{Q}, \mathbf{W}, \widehat{\mathbf{Y}}, \mathbf{U}) \\ &= \|\mathbf{XQ} - \widehat{\mathbf{Y}}\|_{2,1} + \mu_1\|\mathbf{W}\|_{2,1} + \mu_2\|\mathbf{M}\|_1 + \lambda_1\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_{2,1} \\ &\quad + \lambda_2\|\mathbf{S}\widehat{\mathbf{Y}}\|_{2,1} + \frac{\rho}{2}\|\mathbf{M} + \mathbf{W} - \mathbf{Q} + \mathbf{U}\|_F^2. \end{aligned} \tag{16}$$

Equation (16) can be solved by the following alternative methods at k-th iteration:

$$\begin{cases} \mathbf{W}^{k+1} = \min_{\mathbf{W}} \mu_1\|\mathbf{W}\|_{2,1} + \frac{\rho^k}{2}\|\mathbf{M}^k + \mathbf{W} - \mathbf{Q}^k + \mathbf{U}^k\|_F^2 \\ \mathbf{M}^{k+1} = \min_{\mathbf{M}} \mu_2\|\mathbf{M}\|_1 + \frac{\rho^k}{2}\|\mathbf{M} + \mathbf{W}^{k+1} - \mathbf{Q}^k + \mathbf{U}^k\|_F^2 \\ \widehat{\mathbf{Y}}^{k+1} = \min_{\widehat{\mathbf{Y}}} \|\mathbf{XQ}^k - \widehat{\mathbf{Y}}\|_{2,1} + \lambda_1\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_{2,1} + \lambda_2\|\mathbf{S}\widehat{\mathbf{Y}}\|_{2,1} \\ \mathbf{Q}^{k+1} = \min_{\mathbf{Q}} \|\mathbf{XQ} - \widehat{\mathbf{Y}}^{k+1}\|_{2,1} + \frac{\rho^k}{2}\|\mathbf{M}^{k+1} + \mathbf{W}^{k+1} \\ \qquad\qquad - \mathbf{Q} + \mathbf{U}^k\|_F^2 \\ \mathbf{U}^{k+1} = \mathbf{M}^{k+1} + \mathbf{W}^{k+1} - \mathbf{Q}^{k+1} + \mathbf{U}^k \\ \rho^{k+1} = \beta\rho^k \end{cases} \tag{17}$$

where $\beta > 1$ is utilized to update ρ.

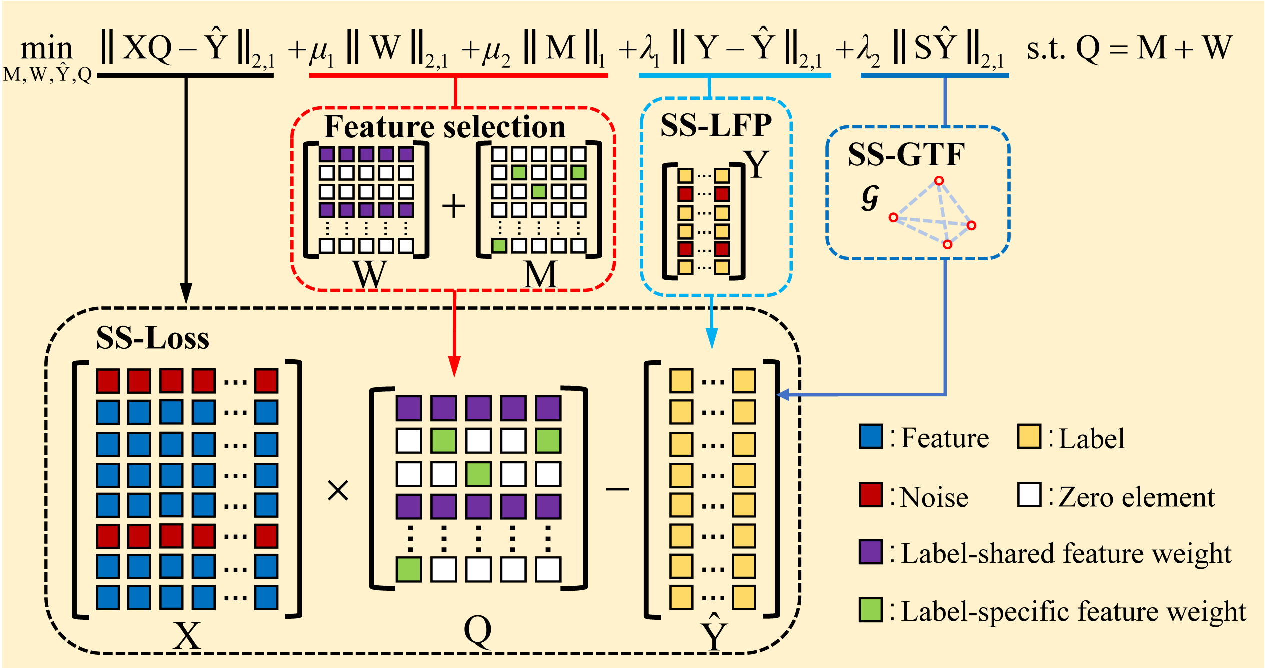

$\min_{\mathrm{M,W,\hat{Y},Q}} \|\mathrm{XQ}-\hat{\mathrm{Y}}\|_{2,1} + \mu_1\|\mathrm{W}\|_{2,1} + \mu_2\|\mathrm{M}\|_1 + \lambda_1\|\mathrm{Y}-\hat{\mathrm{Y}}\|_{2,1} + \lambda_2\|\mathrm{S}\hat{\mathrm{Y}}\|_{2,1}$ s.t. $\mathrm{Q}=\mathrm{M}+\mathrm{W}$
Feature selection
W
+
M
SS-LFP
Y
SS-GTF
$\mathcal{G}$
SS-Loss
X
×
Q
−
Ŷ
🟦: Feature
🟨: Label
🟥: Noise
□: Zero element
🟪: Label-shared feature weight
🟩: Label-specific feature weight

FIGURE 2. The illustration of proposed tri-structured-sparsity induced joint feature selection and classification (TriS-JFSC) model.

Update W: According to [39], the subproblem related to **W** has a close-form solution, which is represented as

$$\mathbf{w}_i^{k+1} = \frac{\max(\|\mathbf{q}_i^k-\mathbf{m}_i^k-\mathbf{u}_i^k\|_2-\frac{\mu_1}{\rho^k}, 0)}{\|\mathbf{q}_i^k-\mathbf{m}_i^k-\mathbf{u}_i^k\|_2+\epsilon}(\mathbf{q}_i^k-\mathbf{m}_i^k-\mathbf{u}_i^k) \tag{18}$$

where $\mathbf{w}_i^{k+1}$, $\mathbf{q}_i^k$, $\mathbf{m}_i^k$, and $\mathbf{u}_i^k$ represent the i-th row of $\mathbf{W}^{k+1}$, $\mathbf{Q}^k$, $\mathbf{M}^k$, and $\mathbf{U}^k$, respectively. And ϵ is a small positive number inserted to avoid division by 0.

Update M: M-subproblem is a standard proximal operator, which has the closed-form solution as follows [40]:

$$\mathbf{M}^{k+1} = \mathrm{sgn}(\mathbf{Q}^k - \mathbf{W}^{k+1} - \mathbf{U}^k) \odot \max(\mathbf{Q}^k - \mathbf{W}^{k+1} - \mathbf{U}^k - \frac{\mu_2}{\rho^k}, 0) \tag{19}$$

where operator $\odot$ denotes the Hadamard product and sgn$(\cdot)$ is the signum function.

Update $\widehat{\mathbf{Y}}$: To update $\widehat{\mathbf{Y}}$, we rewrite the subproblem as following form:

$$\begin{aligned}\Lambda(\widehat{\mathbf{Y}}) &= \|\mathbf{X}\mathbf{Q}^k - \widehat{\mathbf{Y}}\|_{2,1} + \lambda_1\|\mathbf{Y}-\widehat{\mathbf{Y}}\|_{2,1} + \lambda_2\|\mathbf{S}\widehat{\mathbf{Y}}\|_{2,1}\\ &= \mathrm{tr}((\mathbf{X}\mathbf{Q}^k - \widehat{\mathbf{Y}})^T\mathbf{D}_1(\mathbf{X}\mathbf{Q}^k - \widehat{\mathbf{Y}}))\\ &\quad +\lambda_1\,\mathrm{tr}((\mathbf{Y}-\widehat{\mathbf{Y}})^T\mathbf{D}_2(\mathbf{Y}-\widehat{\mathbf{Y}}))\\ &\quad +\lambda_2\,\mathrm{tr}((\mathbf{S}\widehat{\mathbf{Y}})^T\mathbf{D}_3(\mathbf{S}\widehat{\mathbf{Y}}))\end{aligned} \tag{20}$$

where tr$(\cdot)$ denotes the trace of the matrix. The diagonal matrices $\mathbf{D}_1 \in \mathbb{R}^{n\times n}$, $\mathbf{D}_2 \in \mathbb{R}^{n\times n}$, and $\mathbf{D}_3 \in \mathbb{R}^{\varepsilon\times\varepsilon}$ are calculated with

$$d_{i,i}^1 = \frac{1}{\|\mathbf{x}_i\mathbf{Q}^k - \widehat{\mathbf{y}}_i^k\|_2+\epsilon} \tag{21}$$

$$d_{i,i}^2 = \frac{1}{\|\mathbf{y}_i - \widehat{\mathbf{y}}_i^k\|_2+\epsilon} \tag{22}$$

$$d_{i,i}^3 = \frac{1}{\|\mathbf{s}_i\widehat{\mathbf{Y}}^k\|_2+\epsilon}. \tag{23}$$

where $\mathbf{x}_i$, $\mathbf{y}_i$, $\widehat{\mathbf{y}}_i^k$ and $\mathbf{s}_i$ represent the i-th row of $\mathbf{X}$, $\mathbf{Y}$, $\widehat{\mathbf{Y}}^k$ and $\mathbf{S}$, respectively.

Setting the derivative of (20) with respect to $\widehat{\mathbf{Y}}$ to zero, we obtain the following equation:

$$\frac{\mathrm{d}(\Lambda(\widehat{\mathbf{Y}}))}{\mathrm{d}\widehat{\mathbf{Y}}} = \mathbf{D}_1(\widehat{\mathbf{Y}} - \mathbf{X}\mathbf{Q}^k) + \lambda_1\mathbf{D}_2(\widehat{\mathbf{Y}} - \mathbf{Y}) + \lambda_2\mathbf{S}^T\mathbf{D}_3\mathbf{S}\widehat{\mathbf{Y}} = 0. \tag{24}$$

Then we have

$$\widehat{\mathbf{Y}} = (\mathbf{D}_1 + \lambda_1\mathbf{D}_2 + \lambda_2\mathbf{S}^T\mathbf{D}_3\mathbf{S})^{-1}(\mathbf{D}_1\mathbf{X}\mathbf{Q}^k + \lambda_1\mathbf{D}_2\mathbf{Y}). \tag{25}$$

Note that $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$ are also unknown variables, hence $\widehat{\mathbf{Y}}$ cannot be solved directly. However, [15] provides an iterative solution: calculate $\widehat{\mathbf{Y}}$ with the current $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$, and then updating $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$ based on the calculated $\widehat{\mathbf{Y}}$. Repeat this process and the optimal solution will be obtained ultimately.

Update Q: Similar to $\widehat{\mathbf{Y}}$, we rewrite the subproblem as

$$\begin{aligned}\mathcal{T}(\mathbf{Q}) &= \|\mathbf{X}\mathbf{Q} - \widehat{\mathbf{Y}}^{k+1}\|_{2,1} + \frac{\rho^k}{2}\|\mathbf{M}^{k+1} + \mathbf{W}^{k+1} - \mathbf{Q} + \mathbf{U}^k\|_F^2\\ &= \mathrm{tr}((\mathbf{X}\mathbf{Q} - \widehat{\mathbf{Y}}^{k+1})^T\mathbf{D}_1(\mathbf{X}\mathbf{Q} - \widehat{\mathbf{Y}}^{k+1}))\\ &\quad + \frac{\rho^k}{2}\|\mathbf{M}^{k+1} + \mathbf{W}^{k+1} - \mathbf{Q} + \mathbf{U}^k\|_F^2.\end{aligned} \tag{26}$$

Set the derivative to zeros, then we have

$$\mathbf{Q} = (\mathbf{X}^T\mathbf{D}_1\mathbf{X} + \rho^k\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{D}_1\widehat{\mathbf{Y}}^{k+1} + \rho^k(\mathbf{M}^{k+1} + \mathbf{W}^{k+1} + \mathbf{U}^k)) \tag{27}$$

where $\mathbf{I} \in \mathbb{R}^{d\times d}$ is the identical matrix.

Algorithm 1

```
Input: Trianing data: $\mathbf{X}$, $\mathbf{Y}$, and the parameters:
        $\lambda_1, \lambda_2, \mu_1, \mu_2$
Output: $\mathbf{M}$, $\mathbf{W}$, $\widehat{\mathbf{Y}}$, $\mathbf{Q}$
Initialization:$\mathbf{W}^0 = \mathbf{M}^0 = \mathbf{Q}^0 = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + 0.1 \times \mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$,
        set $\mathbf{U}^0 = \mathbf{0}, k = 0, d^1_{i,i} = d^2_{i,i} = d^3_{i,i} = 1$,
        $\rho^0 = 10^{-4}$, $\beta = 1.5$;
repeat
    $\mathbf{w}_i^{k+1} = \frac{\max(\|\mathbf{q}_i^k - \mathbf{m}_i^k - \mathbf{u}_i^k\|_2 - \frac{\mu_1}{\rho^k}, 0)}{\|\mathbf{q}_i^k - \mathbf{m}_i^k - \mathbf{u}_i^k\|_2 + \varepsilon}(\mathbf{q}_i^k - \mathbf{m}_i^k - \mathbf{u}_i^k)$
    $\mathbf{M}^{k+1} = \mathrm{sgn}(\mathbf{Q}^k - \mathbf{W}^{k+1} - \mathbf{U}^k) \odot \max(\mathbf{Q}^k - \mathbf{W}^{k+1} - \mathbf{U}^k - \frac{\mu_2}{\rho}, 0)$
    repeat
        $\widehat{\mathbf{Y}}^{k+1} = (\mathbf{D}_1 + \lambda_1\mathbf{D}_2 + \lambda_2\mathbf{S}^{\mathrm{T}}\mathbf{D}_3\mathbf{S})^{-1}(\mathbf{D}_1\mathbf{X}\mathbf{Q}^k + \lambda_1\mathbf{D}_2\mathbf{Y})$
        $d^1_{i,i} = 1/(\|\mathbf{x}_i\mathbf{Q}^k - \widehat{\mathbf{y}}_i^k\|_2 + \epsilon)$,
        $d^2_{i,i} = 1/(\|\mathbf{y}_i - \widehat{\mathbf{y}}_i^k\|_2 + \epsilon)$,
        $d^3_{i,i} = 1/(\|\mathbf{s}_i\widehat{\mathbf{Y}}^k\|_2 + \epsilon)$,
    until convergence;
    repeat
        $\mathbf{Q}^{k+1} = (\mathbf{X}^{\mathrm{T}}\mathbf{D}_1\mathbf{X} + \rho\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}\mathbf{D}_1\widehat{\mathbf{Y}}^{k+1} + \rho(\mathbf{M}^{k+1} + \mathbf{W}^{k+1} + \mathbf{U}^k))$
        $d^1_{i,i} = 1/(\|\mathbf{x}_i\mathbf{Q}^k - \widehat{\mathbf{y}}_i^k\|_2 + \epsilon)$
    until convergence;
    $\mathbf{U}^{k+1} = \mathbf{M}^{k+1} + \mathbf{W}^{k+1} - \mathbf{Q}^{k+1} + \mathbf{U}^k$; $\rho^{k+1} = \beta\rho^k$
until convergence;
```

The procedure of optimizing TriS-JFSC is summarized in Algorithm 1.

B. COMPLEXITY ANALYSIS

We mainly analysis the complexity of the optimized parts in Algorithm 1. Recall that $\mathbf{X} \in \mathbb{R}^{n\times(d+1)}$, $\mathbf{Y}, \widehat{\mathbf{Y}} \in \{0, 1\}^{n\times c}$, $\mathbf{Q}, \mathbf{W}, \mathbf{M} \in \mathbb{R}^{(d+1)\times c}$, $\mathbf{S} \in \{-1, 0, 1\}^{\varepsilon\times n}$, $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{n\times n}$, and $\mathbf{D}_3 \in \mathbb{R}^{\varepsilon\times\varepsilon}$, where n is the number of instances, d is the dimensionality of features, c is the number of labels and ε is the number of edges in the Graph $\mathcal{G}$. The algorithm consists of five parts: initialization and four subproblems, that is, $\mathbf{W}$-subproblem, $\mathbf{M}$-subproblem, $\widehat{\mathbf{Y}}$-subproblem, and $\mathbf{Q}$-subproblem. For initialization, the complexity is $\mathcal{O}(nd^2 + d^3 + ndc + d^2c)$. The complexity of updating $\mathbf{W}$ and $\mathbf{M}$ are both $\mathcal{O}(dc)$. For updating $\widehat{\mathbf{Y}}$, the complexity is $\mathcal{O}(n\varepsilon^2 + n^2\varepsilon + n^3 + n^2c + ndc + nc\varepsilon)$. For updating $\mathbf{Q}$, the complexity is $\mathcal{O}(n^2d + nd^2 + d^3 + n^2c + ndc + d^2c)$.

V. EXPERIMENTAL RESULTS

A. DATASETS

We conduct our experiments on 7 multilabel benchmark datasets. All these datasets can be found on Mulan[2] and MEKA[3] website. The details of the datasets are summarized in Table 1. Label cardinality (LCard) indicates the average number of labels associated with each sample. We randomly select 2/3 of the total samples from each dataset as training data and the rest as testing data. To avoid randomness, this process is repeated 10 times independently, and the average results with standard deviation are reported as the final performance.

TABLE 1. Experiment datasets.

Dataset	Domain	#Instances	#Features	#Labels	LCard
Birds[2]	AUDIO	645	260	19	1.0
CAL500[2]	MUSIC	502	68	174	26.0
Emotions[2]	MUSIC	593	72	6	1.9
Genbase[2]	BIOLOGY	662	1186	27	1.3
Yeast[2]	BIOLOGY	2417	103	14	4.2
Language log[3]	TEXT	1459	1004	75	1.2
Scene[2]	IMAGE	2407	294	6	1.1

B. EVALUATION METRICS

We employ five popular metrics for evaluation which favour different properties for multilabel classification: *Hamming Loss*, *Ranking Loss*, *One-error*, *Coverage* and *Average Precision*.

1) *Hamming Loss* calculates the proportion of misclassified labels for each sample.
2) *Ranking Loss* calculates the fraction that an irrelevant label is ranked higher than a relevant label.
3) *One-error* evaluates the fraction of samples whose top-ranked label does not belong to the relevant label set.
4) *Coverage* calculates how many steps are needed to move down the predicted label ranking to cover all the relevant labels of the instances.
5) *Average Precision* evaluates the Jaccard similarity between the predicted results and the groundtruth.

Please refer to the work in [16] for detailed information of these metrics. For the first four metrics, smaller value indicates better performance of the classifier. For *Average Precision*, larger value indicates better performance.

C. COMPARING METHODS

We compare our method with several state-of-the-art multilabel classification methods, including the baseline method *BR* [4], the lazy learning approach based on k-nearest neighbours (*ML-kNN*) [18], three feature selection methods: 1) *JFSC* [38]; 2) *SFUS* [36]; and 3) *RFS* [15]. *JFSC*, *SFUS* and *RFS* also employ the structured sparsity to alleviate feature noise. To show our robustness for different types of noise, we also compare our method with *HNOML* [14], which is designed to handle hybrid noise. The parameters of each comparing algorithm are tuned as suggested ways, which is shown as follows:

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

[3] http://waikato.github.io/meka/datasets

TABLE 2. Experimental results (mean±std) of each comparing algorithm in terms of each evaluation metric. ↓ (↑) indicates the smaller (larger), the better. The best values are marked as bold and the second best values are marked as underline. The last line counts the times each method achieves the best and the second best on different datasets with respect to different metrics.

Datasets	Metrics	BR	RFS	HNOML	JFSC	ML-kNN	SFUS	Ours
Birds	*Hamming Loss* ↓	0.0509 ± 0.0021	0.0519 ± 0.0028	0.0497 ± 0.0022	**0.0495 ± 0.0022**	0.0548 ± 0.0030	0.0515 ± 0.0029	0.0519 ± 0.0023
	Ranking Loss ↓	0.1090 ± 0.0225	0.1153 ± 0.0073	0.1123 ± 0.0090	0.1075 ± 0.0079	0.1109 ± 0.0055	0.1117 ± 0.0076	**0.1012 ± 0.0067**
	One-error ↓	0.3042 ± 0.0343	0.2981 ± 0.0186	0.3005 ± 0.0229	**0.2842 ± 0.0264**	0.3270 ± 0.0264	0.2926 ± 0.0204	0.2902 ± 0.0159
	Coverage ↓	0.1624 ± 0.0190	0.1733 ± 0.0067	0.1647 ± 0.0101	0.1617 ± 0.0090	0.1606 ± 0.0063	0.1689 ± 0.0086	**0.1524 ± 0.0099**
	Average Precision ↑	0.7392 ± 0.0228	0.7401 ± 0.0118	0.7450 ± 0.0121	**0.7553 ± 0.0139**	0.7322 ± 0.0161	0.7445 ± 0.0125	0.7526 ± 0.0126
CAL500	*Hamming Loss* ↓	0.1361 ± 0.0023	0.1366 ± 0.0024	**0.1357 ± 0.0021**	**0.1357 ± 0.0021**	0.1378 ± 0.0017	0.1363 ± 0.0019	0.1358 ± 0.0016
	Ranking Loss ↓	0.1799 ± 0.0054	0.1878 ± 0.0038	0.1808 ± 0.0052	0.1782 ± 0.0052	0.1834 ± 0.0037	0.1839 ± 0.0056	**0.1765 ± 0.0051**
	One-error ↓	**0.1024 ± 0.0000**	0.1317 ± 0.0234	0.1084 ± 0.0188	0.1048 ± 0.0214	0.1066 ± 0.0159	0.1126 ± 0.0193	0.1042 ± 0.0200
	Coverage ↓	0.7534 ± 0.0013	0.7893 ± 0.0103	0.7606 ± 0.0138	0.7526 ± 0.0126	0.7506 ± 0.0116	0.7708 ± 0.0143	**0.7487 ± 0.0121**
	Average Precision ↑	0.5043 ± 0.0035	0.5045 ± 0.0099	0.5124 ± 0.0077	0.5119 ± 0.0077	0.4928 ± 0.0050	0.5101 ± 0.0083	**0.5143 ± 0.0079**
Emotions	*Hamming Loss* ↓	0.2120 ± 0.0103	0.2072 ± 0.0070	0.2077 ± 0.0087	0.2132 ± 0.0092	**0.2003 ± 0.0075**	0.2073 ± 0.0056	0.2029 ± 0.0090
	Ranking Loss ↓	0.1732 ± 0.0144	0.1793 ± 0.0194	0.1781 ± 0.0202	0.1797 ± 0.0192	0.1636 ± 0.0135	0.1809 ± 0.0195	**0.1632 ± 0.0177**
	One-error ↓	0.2934 ± 0.0257	0.2869 ± 0.0144	0.2869 ± 0.0187	0.2914 ± 0.0191	**0.2636 ± 0.0276**	0.2874 ± 0.0167	0.2692 ± 0.0210
	Coverage ↓	0.3077 ± 0.0184	0.3144 ± 0.0219	0.3136 ± 0.0222	0.3146 ± 0.0204	**0.2963 ± 0.0159**	0.3161 ± 0.0213	0.2982 ± 0.0187
	Average Precision ↑	0.7857 ± 0.0142	0.7877 ± 0.0141	0.7879 ± 0.0159	0.7863 ± 0.0160	**0.8014 ± 0.0144**	0.7864 ± 0.0140	0.8005 ± 0.0163
Genbase	*Hamming Loss* ↓	0.0017 ± 0.0029	0.0017 ± 0.0034	0.0042 ± 0.0034	0.0021 ± 0.0032	0.0024 ± 0.0007	**0.0012 ± 0.0036**	0.0013 ± 0.0045
	Ranking Loss ↓	0.0051 ± 0.0058	0.0045 ± 0.0049	0.0033 ± 0.0047	0.0037 ± 0.0047	0.0063 ± 0.0037	**0.0031 ± 0.0048**	0.0061 ± 0.0041
	One-error ↓	0.0018 ± 0.0076	0.0018 ± 0.0076	0.0054 ± 0.0086	**0.0014 ± 0.0063**	0.0041 ± 0.0066	0.0023 ± 0.0066	0.0018 ± 0.0056
	Coverage ↓	0.0183 ± 0.0086	0.0171 ± 0.0062	0.0147 ± 0.0053	0.0126 ± 0.0054	0.0197 ± 0.0069	0.0145 ± 0.0060	**0.0122 ± 0.0053**
	Average Precision ↑	0.9914 ± 0.0047	0.9937 ± 0.0051	0.9911 ± 0.0051	0.9941 ± 0.0048	0.9910 ± 0.0054	0.9937 ± 0.0048	**0.9947 ± 0.0051**
Yeast	*Hamming Loss* ↓	0.2047 ± 0.0006	0.2018 ± 0.0003	0.2022 ± 0.0378	0.2027 ± 0.0004	**0.1953 ± 0.0037**	0.2017 ± 0.0024	0.2002 ± 0.0003
	Ranking Loss ↓	0.1775 ± 0.0164	0.1780 ± 0.0093	0.1784 ± 0.0315	0.1761 ± 0.0147	**0.1667 ± 0.0052**	0.1773 ± 0.0071	0.1736 ± 0.0131
	One-error ↓	0.2331 ± 0.0244	0.2282 ± 0.0266	0.2295 ± 0.0215	**0.2266 ± 0.0201**	0.2304 ± 0.0119	0.2275 ± 0.0367	0.2267 ± 0.0210
	Coverage ↓	0.4657 ± 0.0130	0.4635 ± 0.0090	0.4644 ± 0.0248	0.4607 ± 0.0153	**0.4471 ± 0.0080**	0.4625 ± 0.0076	0.4589 ± 0.0101
	Average Precision ↑	0.7560 ± 0.0148	0.7579 ± 0.0190	0.7571 ± 0.0175	0.7581 ± 0.0161	**0.7645 ± 0.0056**	0.7585 ± 0.0195	0.7582 ± 0.0125
Language log	*Hamming Loss* ↓	0.0153 ± 0.0032	**0.0152 ± 0.0029**	0.1189 ± 0.0032	0.0155 ± 0.0036	0.0159 ± 0.0002	0.0173 ± 0.0029	**0.0152 ± 0.0029**
	Ranking Loss ↓	**0.1381 ± 0.0116**	0.1423 ± 0.0054	0.4043 ± 0.0049	0.2220 ± 0.0056	0.1671 ± 0.0098	0.1794 ± 0.0049	0.1434 ± 0.0043
	One-error ↓	0.7514 ± 0.0071	**0.6944 ± 0.0147**	0.8877 ± 0.0124	0.7201 ± 0.0128	0.8011 ± 0.0135	0.8570 ± 0.0140	0.6997 ± 0.0090
	Coverage ↓	**0.1498 ± 0.0077**	0.1544 ± 0.0045	0.3953 ± 0.0047	0.2314 ± 0.0050	0.1764 ± 0.0093	0.1863 ± 0.0041	0.1561 ± 0.0038
	Average Precision ↑	0.3824 ± 0.0094	**0.3990 ± 0.0082**	0.1888 ± 0.0071	0.3664 ± 0.0076	0.3064 ± 0.0155	0.2497 ± 0.0075	0.3944 ± 0.0053
Scene	*Hamming Loss* ↓	0.1050 ± 0.0004	0.1160 ± 0.0005	0.1089 ± 0.0010	0.1089 ± 0.0013	**0.0891 ± 0.0039**	0.1159 ± 0.0005	0.1147 ± 0.0004
	Ranking Loss ↓	0.0893 ± 0.0041	0.0928 ± 0.0035	0.0987 ± 0.0019	0.0986 ± 0.0036	**0.0724 ± 0.0053**	0.0905 ± 0.0022	0.0837 ± 0.0035
	One-error ↓	0.2557 ± 0.0000	0.2670 ± 0.0023	0.2680 ± 0.0036	0.2663 ± 0.0031	**0.2244 ± 0.0183**	0.2620 ± 0.0032	0.2435 ± 0.0023
	Coverage ↓	0.0882 ± 0.0035	0.0911 ± 0.0066	0.0965 ± 0.0041	0.0964 ± 0.0032	**0.0741 ± 0.0038**	0.0892 ± 0.0044	0.0837 ± 0.0028
	Average Precision ↑	0.8462 ± 0.0032	0.8398 ± 0.0031	0.8364 ± 0.0034	0.8371 ± 0.0035	**0.8687 ± 0.0092**	0.8430 ± 0.0031	0.8543 ± 0.0027
	Total	**3** + 2	**3** + 3	**1** + 3	**6** + 4	**13** + 3	**2** + 1	**9** + 19

1) *BR*: The LIBSVM toolbox [41] is utilized as the basic binary classifier of BR. We select the linear kernel and tune the parameter C in $\{10^{-3}, 10^{-2}, \dots, 10^{3}\}$.
2) *ML-kNN*:[4] The parameter k is tuned in $\{3, 5, \dots, 21\}$.
3) *JFSC*: The threshold τ is set to 0.5. The parameters α, β, and γ are searched in $\{4^{-5}, 4^{-4}, \dots, 4^{5}\}$, and η is tuned in $\{10^{-1}, 10^{0}, 10^{1}\}$.
4) *SFUS*:[5] The parameters α and β are searched in $\{10^{-3}, 10^{-2}, \dots, 10^{3}\}$.
5) *RFS*:[6] The parameter γ is tuned in $\{10^{-5}, 10^{-4}, \dots, 10^{1}\}$.
6) *HNOML*: The parameters α, β, and γ are searched in $\{10^{-3}, 10^{-2}, \dots, 10^{3}\}$.

In our experiments, the proposed TriS-JFSC model involves 7 parameters that need to be determined, i.e. ξ, β, k, μ_1, μ_2, λ_1, and λ_2,. Empirically, we set $\xi = 1$, $\beta = 1.2$ and $k = 5$ as their changes have little impact on the performance. Then we tune the other four parameters from the set $\{10^{-3}, 10^{-2}, \dots, 10^{1}\}$ and use the gird searching strategy to determine the optimal values for each parameter. Finally, the optimal configuration is applied to the testing data.

D. QUANTITATIVE RESULTS

Table 2 shows the results of these aforementioned algorithm over 7 datasets in terms of 5 evaluation metrics, through which the following conclusions can be obtained:

1) Our method achieves the most competitive performance on all datasets in terms of five metrics. As we can see in the last line of Table 2, our method ranks first for 9 times and ranks second for 19 times, which are much more than other methods.
2) The nearset competitor is ML-kNN, which performs slightly better on a few datasets. However, our method exhibits stronger stability on all datasets.
3) Most feature selection methods (JFSC, our method and RFS) perform well on the datasets. Among them, JFSC and our method consider both the label-specific and label-shared features, whereas conventional selection methods like RFS only select shared features. Hence, the former two methods generate higher performance than the latter.
4) BR cannot achieve satisfactory performance because it merely converts multilabel classification into multiple binary classification problems, which does not consider the label correlation and feature selection.
5) HNOML does not consider the feature selection problem, which results in its poor performance on the original data compared with other feature selection methods. Nonetheless, HNOML can achieve stable performance for various types of noise, which can be seen in the Robustness Results.

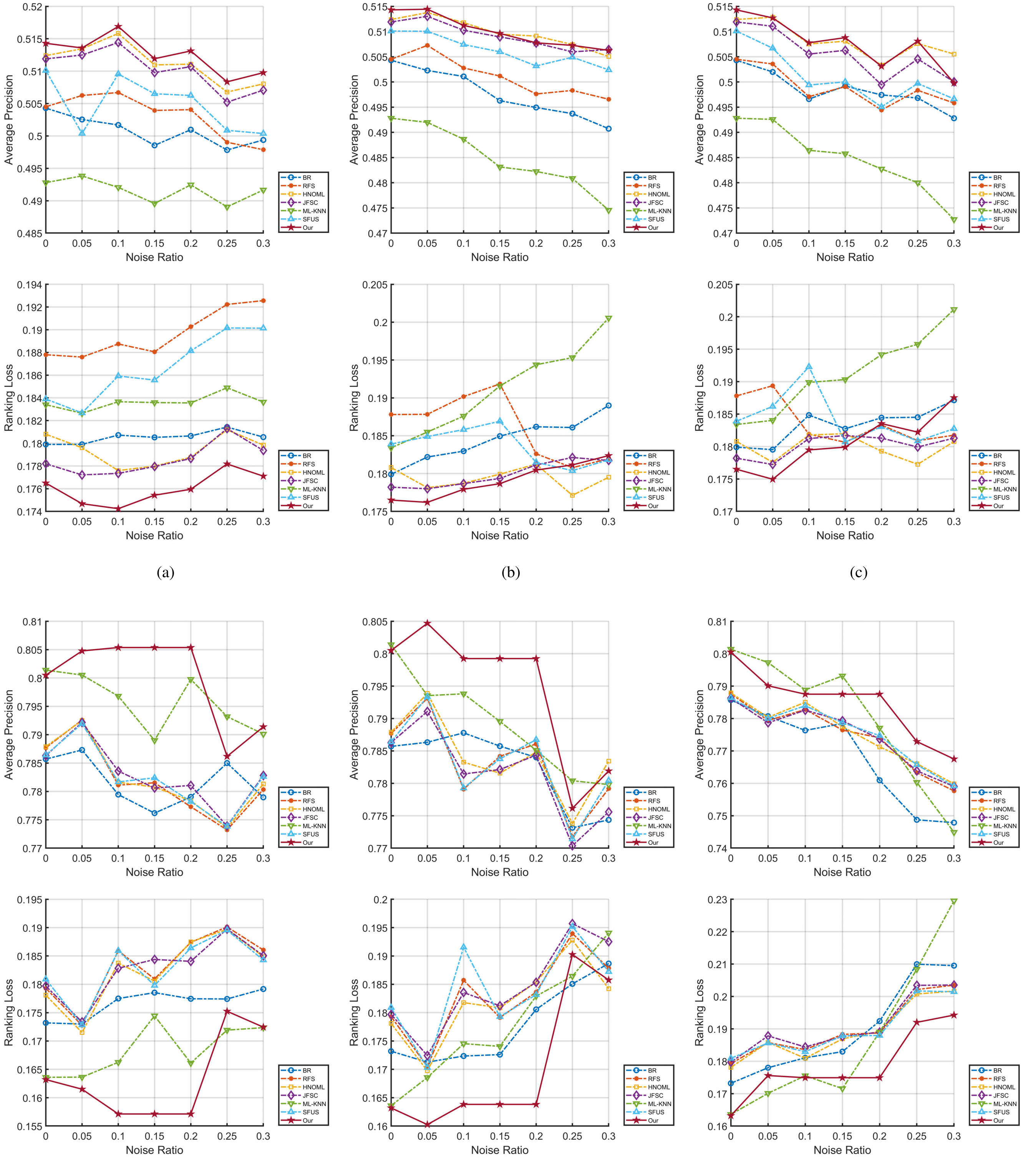
E. ROBUSTNESS RESULTS

In order to evaluate the robustness of the proposed method to different types of noise, we illustrate the performance

[4] http://www.lamda.nju.edu.cn/code_MLkNN.ashx

[5] http://www.cs.cmu.edu/∼ kevinma/

[6] http://www.escience.cn/people/fpnie/index.html



(a) (b) (c)

(d) (e) (f)

FIGURE 3. Robustness results with different types of noise. The first to third columns correspond to feature noise, label noise, and hybrid noise, respectively. (a)-(c) and (d)-(f) correspond to CAL500 and Emotions respectively.

of all methods on Emotions and CAL500 with different types and different degrees of noise, as shown in Fig. 3. Specifically, three types of noise are added to the datasets: feature noise, label noise, and hybrid noise. For label noise,

TABLE 3. Comparison results (mean±std) of TriS-JFSC and its two ablation models. ↓ (↑) indicates the smaller (larger), the better. The best values are marked as bold.

Datasets	Metrics	TriS-JFSC-nF	TriS-JFSC-nG	TriS-JFSC
Birds	*Hamming Loss* ↓	0.0561±0.0028	**0.0511±0.0024**	0.0519±0.0023
	Ranking Loss ↓	0.1214±0.0086	0.1128±0.0079	**0.1012±0.0067**
	One-error ↓	0.3219±0.0179	0.2935±0.0189	**0.2902±0.0159**
	Coverage ↓	0.1761±0.0095	0.1657±0.0077	**0.1524±0.0099**
	Average Precision ↑	0.7270±0.0117	0.7486±0.0119	**0.7526±0.0126**
CAL500	*Hamming Loss* ↓	0.1390±0.0018	0.1403±0.0022	**0.1358±0.0016**
	Ranking Loss ↓	0.2016±0.0045	0.1852±0.0047	**0.1765±0.0051**
	One-error ↓	0.1515±0.0155	0.1102±0.0187	**0.1042±0.0200**
	Coverage ↓	0.8162±0.0108	0.7517±0.0111	**0.7487±0.0121**
	Average Precision ↑	0.4934±0.0076	0.4956±0.0074	**0.5143±0.0079**
Emotions	*Hamming Loss* ↓	0.2056±0.0115	0.2053±0.0074	**0.2029±0.0090**
	Ranking Loss ↓	0.1664±0.0186	0.1720±0.0211	**0.1632±0.0177**
	One-error ↓	0.2717±0.0215	**0.2687±0.0276**	0.2692±0.0210
	Coverage ↓	0.3008±0.0195	0.3050±0.0177	**0.2982±0.0187**
	Average Precision ↑	0.7984±0.0160	0.7972±0.0197	**0.8005±0.0163**
Genbase	*Hamming Loss* ↓	0.0031±0.0003	0.0014±0.0003	**0.0013±0.0045**
	Ranking Loss ↓	**0.0030±0.0018**	0.0031±0.0032	0.0061±0.0041
	One-error ↓	0.0036±0.0024	0.0032±0.0000	**0.0018±0.0056**
	Coverage ↓	0.0136±0.0041	0.0137±0.0027	**0.0122±0.0053**
	Average Precision ↑	0.9926±0.0023	0.9947±0.0027	**0.9947±0.0051**
Yeast	*Hamming Loss* ↓	0.2215±0.0161	0.2127±0.0100	**0.2002±0.0003**
	Ranking Loss ↓	0.2030±0.0175	0.1825±0.0140	**0.1736±0.0131**
	One-error ↓	0.2760±0.0362	0.2473±0.0427	**0.2267±0.0210**
	Coverage ↓	0.5000±0.0188	0.4672±0.0199	**0.4589±0.0101**
	Average Precision ↑	0.7270±0.0203	0.7408±0.0198	**0.7582±0.0125**
Language log	*Hamming Loss* ↓	0.0157±0.0003	**0.0152±0.0004**	0.0152±0.0029
	Ranking Loss ↓	0.1685±0.0089	**0.1237±0.0088**	0.1434±0.0043
	One-error ↓	0.8899±0.0133	0.7407±0.0262	**0.6997±0.0090**
	Coverage ↓	0.2053±0.0103	0.1564±0.0096	**0.1561±0.0038**
	Average Precision ↑	0.2160±0.0100	0.3402±0.0193	**0.3944±0.0053**
Scene	*Hamming Loss* ↓	**0.1144±0.0024**	0.1622±0.0016	0.1147±0.0004
	Ranking Loss ↓	0.0942±0.0049	0.0938±0.0051	**0.0837±0.0035**
	One-error ↓	0.2647±0.0085	0.2781±0.0148	**0.2435±0.0023**
	Coverage ↓	0.0928±0.0040	0.0920±0.0038	**0.0837±0.0028**
	Average Precision ↑	0.8403±0.0060	0.8354±0.0079	**0.8543±0.0027**

we randomly exchange the 0 and 1 value of the labels with the ratio of selected samples from 0% to 30% with step size 5% (0% means the original data). To simulate the feature noise, according to [42], we generate the noise matrix $\mathbf{E}$ with $\delta = 0.5$ to control the noise magnitude, and add $\mathbf{E}$ to the selected samples with the proportion from 0% to 30%. For hybrid noise, We combine the two types of noise directly and set the ratio from 0% to 30%.

From Fig. 3, we can see that our method can maintain stable and competitive performance on noisy data of different types and different degrees compared with other methods. Meanwhile, we find that BR and ML-kNN, which do not consider noise, perform significantly worse than the other methods in terms of noisy data. As the degree of noise increases, their *Average Precision* decreases more significantly. Besides, we can see that HNOML is stable for different types of noise, especially for hybrid noise.

F. ABLATION STUDY

To validate the effectiveness of the feature selection scheme and SS-GTF regularization in the proposed Tris-JFSC model, we conduct an ablation study by comparing TriS-JFSC with its two ablation models: (1) TriS-JFSC-nF, which drops the feature selection terms (the second and the third term in (14)); (2) TriS-JFSC-nG, which replaces SS-GTF ($\|\mathbf{S}\widehat{\mathbf{Y}}\|_{2,1}$) with Laplacian regularization ($tr(\widehat{\mathbf{Y}}^{\mathrm{T}}\mathbf{L}\widehat{\mathbf{Y}})$). $\mathbf{L} = \mathbf{B} - \mathbf{A} \in \mathbb{R}^{n\times n}$ is a Laplacian matrix, where $\mathbf{B}$ is a diagonal degree matrix with $b_{i,i} = \sum_{j=1}^{n} a_{i,j}$. The comparison results between TriS-JFSC and its two ablation models are shown in Table 3. We can observe that the performance declines rapidly when TriS-JFSC drops either of these two components in the model, which demonstrates that the feature selection scheme and SS-GTF regulation are helpful to the proposed TriS-JFSC model.

G. SENSITIVITY ANALYSIS OF PARAMETERS

In our method, there are four different parameters need to be tuned. Now we study the sensitivity of proposed method to parameter setting on Emotions, CAL500, Genbase, and Birds datasets. To this end, we first vary one of the four parameters and fix the values of the other parameters to the optimal configuration. We randomly select 2/3 of the total samples from each dataset as training data and the rest as testing data. We repeat this process for 10 times and the average result with different values of μ_1, μ_2, λ_1, and λ_2 are reported in Fig. 4. We can observe that the optimal performance of Tris-JFSC is usually achieved in $\{0.001, 0.01, 0.1\}$ for each parameter. Our model is relatively not sensitive to λ_1 and can maintain a stable performance as λ_1 changes. For the other three parameters, the performance will decline as the value increases.

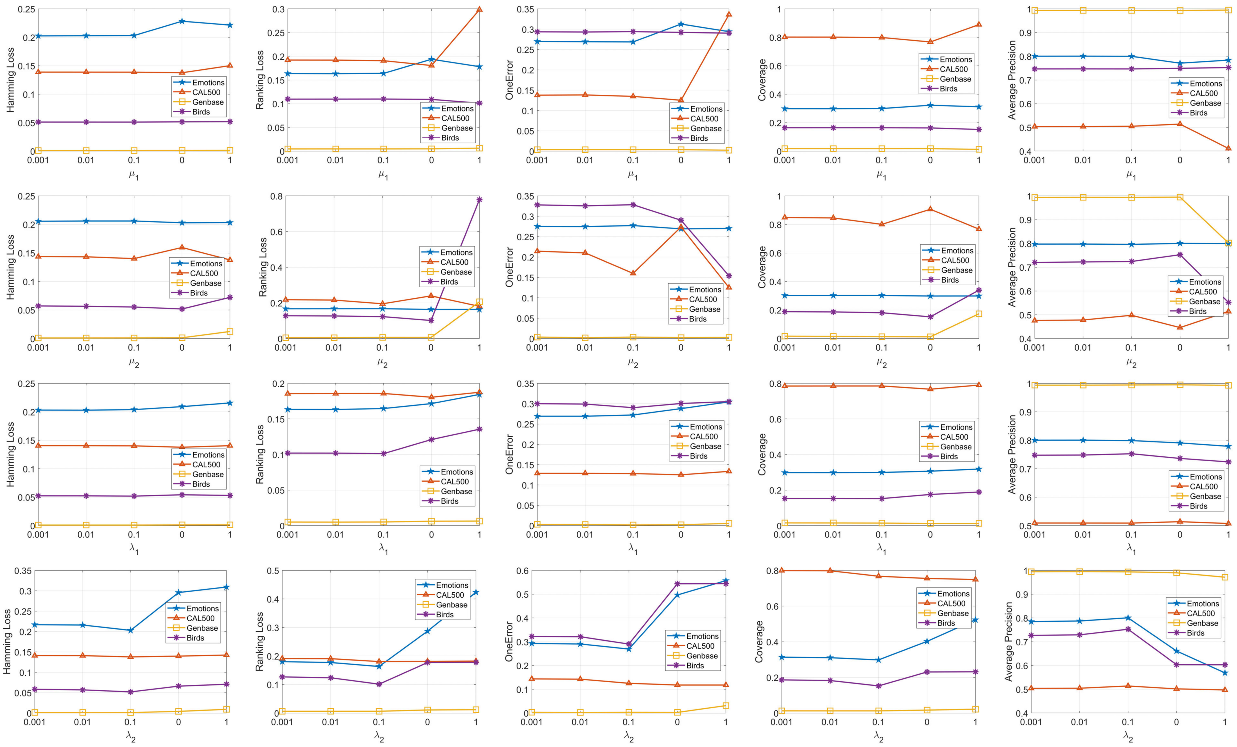


FIGURE 4. Sensitivity analysis of the parameters $\mu_1, \mu_2, \lambda_1, \lambda_2$ in our proposed TriS-JFSC method on Emotions, CAL500, Genbase, and Birds datasets. For *Average Precision*, bigger value indicates the better performance. For the other four metrics, smaller value indicates the better performance.

VI. CONCLUSION

In this paper, we aim to address the robust multilabel learning problem on the imperfect training data with hybrid noise. To this end, we propose a robust Tris-JFSC model to simultaneously smooth the feature and label noise by employing the tri-structured-sparsity regularization scheme, i.e., SS-GTF, SS-LFP and SS-Loss regularizations. In addition, the proposed Tris-JFSC model also utilizes an adaptive feature selection mechanism to boost the learning performance. The experimental results demonstrate the proposed Tris-JFSC model's outstanding performance on noisy data. In the future, we will extend our model to deal with the data with missing labels. Moreover, more general nonlinear dependence assumptions between the samples and labels will be explored.

REFERENCES

[1] Y. Guo, F.-L. Chung, G. Li, and L. Zhang, "Multi-label bioinformatics data classification with ensemble embedded feature selection," *IEEE Access*, vol. 7, pp. 103863–103875, 2019.

[2] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[3] M. Huang, H. Han, H. Wang, L. Li, Y. Zhang, and U. A. Bhatti, "A clinical decision support framework for heterogeneous data sources," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 6, pp. 1824–1833, Nov. 2018.

[4] 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, Sep. 2004.

[5] L. Tang, S. Rajan, and V. K. Narayanan, "Large scale multi-label classification via metalabeler," in *Proc. 18th Int. Conf. World Wide Web (WWW)*, 2009, pp. 211–220.

[6] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *Proc. ECML PKDD Discovery Challenge*, Antwerp, Belgium, 2008, pp. 75–83.

[7] N. Ueda and K. Saito, "Parametric mixture models for multi-label text," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA, USA: MIT Press, 2003, pp. 721–728.

[8] L. Jing, L. Yang, J. Yu, and M. K. Ng, "Semi-supervised low-rank mapping learning for multi-label classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1483–1491.

[9] 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.

[10] F.-F. Luo, W.-Z. Guo, and G.-L. Chen, "Addressing imbalance in weakly supervised multi-label learning," *IEEE Access*, vol. 7, pp. 37463–37472, 2019.

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

[12] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 895–903.

[13] Y. Zhang and Z.-H. Zhou, "Multilabel dimensionality reduction via dependence maximization," in *Proc. AAAI Conf. Artif. Intell.*, Menlo Park, CA, USA, 2008, pp. 1503–1505.

[14] C. Zhang, Z. Yu, H. Fu, P. Zhu, L. Chen, and Q. Hu, "Hybrid noise-oriented multilabel learning," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2837–2850, Jun. 2020.

[15] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell 2$, 1-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[16] 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.

[17] 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.

[18] 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, Jul. 2007.

[19] H. Liu, X. Li, and S. Zhang, "Learning instance correlation functions for multilabel classification," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 499–510, Feb. 2017.

[20] A. Elisseeff and W. Jason, "A kernel method for multi-labelled classification," in *Proc. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2001, pp. 681–687.

[21] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label specific features for multi-label classification," in *Proc. IEEE Int. Conf. Data Mining*, Atlantic City, NJ, USA, Nov. 2015, pp. 181–190.

[22] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2010, pp. 667–685.

[23] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Proc. Eur. Conf. Mach. Learn.*, Warsaw, Poland, 2007, pp. 406–417.

[24] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Pisa, Italy, Dec. 2008, pp. 995–1000.

[25] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. Eur. Conf. Mach. Learn.*, Bled, Slovenia, 2019, pp. 254–269.

[26] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. Eur. Conf. Mach. Learn.*, Catania, Italy, 1994, pp. 171–182.

[27] M. Huang, L. Sun, J. Xu, and S. Zhang, "Multilabel feature selection using relief and minimum redundancy maximum relevance based on neighborhood rough sets," *IEEE Access*, vol. 8, pp. 62011–62031, 2020.

[28] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. 14th Int. Conf. Mach. Learn.*, Jul. 1997, pp. 412–420.

[29] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2000.

[30] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.

[31] M. Monirul Kabir, M. Monirul Islam, and K. Murase, "A new wrapper feature selection approach using neural network," *Neurocomputing*, vol. 73, nos. 16–18, pp. 3273–3283, Oct. 2010.

[32] H. Shao, G. Li, G. Liu, and Y. Wang, "Symptom selection for multi-label data of inquiry diagnosis in traditional chinese medicine," *Sci. China Inf. Sci.*, vol. 56, no. 5, pp. 1–13, May 2013.

[33] M. You, J. Liu, G.-Z. Li, and Y. Chen, "Embedded feature selection for multi-label classification of music emotions," *Int. J. Comput. Intell. Syst.*, vol. 5, no. 4, pp. 668–678, Aug. 2012.

[34] S. Wang, J. Tang, and H. Liu, "Embedded unsupervised feature selection," in *Proc. 29th AAAI Conf. Artif. Intell.*, Feb. 2015, pp. 470–476.

[35] S. Maldonado and J. López, "An embedded feature selection approach for support vector classification via second-order cone programming," *Intell. Data Anal.*, vol. 19, no. 6, pp. 1259–1273, Nov. 2015.

[36] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, and N. Sebe, "Web image annotation via subspace-sparsity collaborated feature selection," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1021–1030, Aug. 2012.

[37] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., B (Methodol.)*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[38] J. Huang, G. Li, Q. Huang, and X. Wu, "Joint feature selection and classification for multilabel learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 876–889, Mar. 2018.

[39] L. Chen, G. Yang, Z. Chen, F. Xiao, and J. Xu, "Web services QoS prediction via matrix completion with structural noise," *J. Commun.*, vol. 36, no. 6, pp. 49–59, 2015.

[40] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[41] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[42] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4333–4341.

LEI XU is currently pursuing the bachelor's degree with the Bell Honors School (BHS), Nanjing University of Posts and Telecommunications, Nanjing, China. His main research interests include machine learning and pattern recognition.

CHUANCHENG SONG (Student Member, IEEE) is currently pursuing the bachelor's degree in engineering with the Bell Honors School (BHS), Nanjing University of Posts and Telecommunications, Nanjing, China. He is also the President of the BHS Association for Science and Technology and also an Intern with the Jiangsu Association for Science and Technology. His research interests include machine learning and complex networks.

LEI CHEN (Member, IEEE) received the M.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 system from the Nanjing University of Posts and Telecommunications, Nanjing, in 2014. He was a Visiting Researcher with The University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, from June 2016 to June 2017. He is currently a Professor with the School of Computer Science, Nanjing University of Posts and Telecommunications. His research interests include machine learning, pattern recognition, and medical image analysis.