Jia XY, Zhu SS, Li WW. Joint label-specific features and correlation information for multi-label learning. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 35(2): 247–258 Mar. 2020. DOI 10.1007/s11390-020-9900-z

Joint Label-Specific Features and Correlation Information for Multi-Label Learning

Xiu-Yi Jia[1], *Member*, *CCF*, *IEEE*, Sai-Sai Zhu[1], and Wei-Wei Li[2], *Member*, *CCF*

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*
[2] *College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*

E-mail: jiaxy@njust.edu.cn; zhusaisworld@163.com; liweiwei@nuaa.edu.cn

Received August 1, 2019; revised January 2, 2020.

Abstract Multi-label learning deals with the problem where each instance is associated with a set of class labels. In multi-label learning, different labels may have their own inherent characteristics for distinguishing each other, and the correlation information has shown promising strength in improving multi-label learning. In this study, we propose a novel multi-label learning method by simultaneously taking into account both the learning of label-specific features and the correlation information during the learning process. Firstly, we learn a sparse weight parameter vector for each label based on the linear regression model, and the label-specific features can be extracted according to the corresponding weight parameters. Secondly, we constrain label correlations directly on the output of labels, not on the corresponding parameter vectors which conflicts with the label-specific feature learning. Specifically, for any two related labels, their corresponding models should have similar outputs rather than similar parameter vectors. Thirdly, we also exploit the sample correlations through sparse reconstruction. The experimental results on 12 benchmark datasets show that the proposed method performs better than the existing methods. The proposed method ranks in the 1st place at 66.7% case and achieves optimal average rank in terms of all evaluation measures.

Keywords multi-label learning, label-specific feature, sparse reconstruction, label correlation, sample correlation

1 Introduction

Multi-label learning deals with instances having a set of class labels simultaneously, which widely exist in real-world applications. The task of multi-label learning is to induce a sophisticated model to assign a set of proper labels for an unseen instance. In recent years, this technique has been increasingly studied and widely applied to various fields including text annotation [1–3], automatic image annotation [4–6], music emotions categorization [7, 8] and so on [9, 10].

During the past decade, a large number of algorithms have been proposed for multi-label learning. To improve the performance of the model, most of these algorithms mainly apply the following two strategies. The first strategy is to exploit the label-specific features for each label during the learning process, since each label might be determined by some specific features of its own. The typical algorithms include LIFT [11], LLSF [12], JFSC [13], and LSFCI [14]. The second commonly used strategy is to utilize the correlations among different labels. RankSVM [15], Random k-Labelsets [16], and MLFE [17] are the typical multi-label learning algorithms by utilizing label correlations.

In this study, we propose a novel multi-label learning method by jointing label-specific features and correlation information, while the correlation information includes label correlations and sample correlations. By reviewing existing work, we find that there are several multi-label learning algorithms which simultaneously consider label-specific features and label correlations [12–14, 18]. In [18], the feature weights and label correlation based features are defined as two unknown variables, which requires additional features for an unseen instance in the testing phase. Thus, they need to adopt a k-NN-like method to predict the addi-

Regular Paper
Special Section on Learning and Mining in Dynamic Environments
This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 61773208 and 61906090, and the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK20191287 and BK20170809.

tional features first. However, the additional introduced prediction part may bring some errors to the final multi-label learning model. On the contrary, we directly integrate the label-specific feature selection and correlation information into the proposed model and optimize the solution. In [12–14], to exploit the label correlations, it is assumed that if the two labels are strongly correlated, the similarity between their corresponding parameter vectors should be large. However, constraining on the parameter vectors will make the corresponding specific features tend to be the same when the two labels are correlated. Obviously, this is not sufficient to characterize all possible relationships between features and labels. Even two labels are correlated, their specific features may also be totally different. In contrast, we constrain label correlations directly on the output of labels, which can implicitly explore the complex relationships between features and labels.

In order to extract the label-specific features, we use l_1-regularization to learn sparse weight parameter vectors. For each label, the non-zero items on the corresponding parameter vector represent the selected label-specific features. Then, the label correlations are applied to improve the learning of label-specific features. Different from previous studies, we constrain that if two labels are correlated to each other, they should have similar outputs rather than similar parameter vectors. Finally, to further exploit the sample correlations, we assume that the outputs of an instance can be reconstructed by others based on the idea of LLE [19]. Specifically, we learn the reconstruction coefficients between one sample and all the other samples in feature space via sparse reconstruction. After that, for each instance, the corresponding reconstruction coefficients are utilized to reconstruct its outputs by others, and the reconstruction error should be as small as possible. To verify the effectiveness of our proposed method, we conduct comprehensive experiments on 12 multi-label datasets. The experimental results indicate that our method achieves highly competitive performance against other state-of-the-art multi-label learning algorithms.

The main contributions of this paper are summarized as follows.

- We propose a novel multi-label learning method by learning label-specific features based on correlation information, named LFCMLL.
- l_1-regularization is applied to sparse the weight parameter vector in which non-zero items represent the selected label-specific features.
- We constrain that if two labels are strongly correlated, they should have similar outputs on the labels rather than on their weight parameter vectors.
- We propose a new regularization term to exploit sample correlations, which aims to minimize the reconstruction error in output space based on the reconstruction relationships among training samples.

The rest of the paper is organized as follows. Firstly, the related work of multi-label learning with correlations and label-specific features is briefly discussed. Secondly, the details of the proposed approach are introduced. Thirdly, experimental results and analysis on 12 multi-label benchmark datasets are reported. Finally, we conclude this paper.

2 Related Work

Many recent studies have suggested that exploiting the label correlations can bring significant benefits to multi-label classification performance [15, 18, 20]. The methodologies of applying label correlations in these studies can be classified into three categories: first-order strategy, second-order strategy, and high-order strategy [9, 10].

The first-order strategy directly decomposes the multi-label learning problem into multiple single-label classification problems followed by the traditional machine learning methods for classification. This strategy completely ignores the label correlations. A representative first-order strategy is BR (Binary Relevance) algorithm [21]. The second-order strategy utilizes the pair wised relationships between two labels, such as the ordering of the related labels and the unrelated labels, the interaction of any two labels, and so on. RankSVM [15] and CLR [22] are two typical multi-label learning algorithms by applying the second-order strategy. The high-order strategy constructs a multi-label learning algorithm by examining high-order label correlations, such as the impact of each label on all other labels, the relevance of a set of random label set, and so on. LEAD [23] and Random k-Labelsets [16] are two representative high-order algorithms.

Since each label may be determined by its own features, label-specific feature selection in multi-label learning has attracted a lot of attention in recent years. LIFT [11] is the first algorithm to exploit label-specific features for multi-label learning, which assumes that if the most pertinent and discriminative features for each class label, called label-specific features, could be used in the learning process, a more effective solution to the

problem of multi-label learning can be expected. It constructs specific features for each label before the training process. For each label, LIFT obtains centers of its positive and negative instances via the algorithm of k-means. Then, it calculates the distances between the original instances and the clustering centers as label-specific features. Finally, SVM is applied on the process of training and testing based on the label-specific features. Moreover, Xu *et al.*[24] presented a multi-label learning approach based on fuzzy rough set, which utilized the approximation quality to evaluate the significance of features. Through applying the forward greedy search strategy, the proposed method achieved label-specific feature reduction. Yan *et al.*[25] employed the information theory to implement label-specific feature selection and assign different weights to the different class instances according to imbalance rate. Huang *et al.*[12] exploited feature selection to obtain label-specific features and proposed the algorithm called LLSF. In LLSF, l_1-regularization is applied to get the weights of features for each label. If the weight of some features is zero for a label, it has no effect on the discrimination of that label. Furthermore, in order to incorporate label correlation, LLSF requires that strongly correlated labels should have large similarity between their weight vectors.

To further improve the learning of label-specific features, LSFCI[14] was proposed to learn label specific features for each label with consideration of label correlation in label space and instance correlation in feature space simultaneously. It directly calculates the instance correlation matrix by the similarity between any two instances using common similarity measures, and then incorporates the instance correlation matrix into model training for improving the learning of label-specific features. However, the instance correlations are simply obtained by common similarity measures, which may not reflect complex relationships among instances. In [13], it proposed to learn label-specific features and shared features by exploiting pairwise label correlation and utilize a Fisher discriminant-based regularization term to minimize the inner-class distance and maximize the intraclass distance for each label.

3 Proposed Approach

3.1 Learning Label-Specific Features

In multi-label learning, suppose $\mathcal{X} \in R^d$ be the d-dimensional input space and $\mathcal{Y} = \{y_1, y_2, \cdots, y_q\}$ represents the label space with q class labels. $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{Y}_i)\}_{i=1}^n$ is a training dataset that consists of n instances, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \cdots, x_{id})$ and $\boldsymbol{Y}_i = (Y_{i1}, Y_{i2}, \cdots, Y_{iq}) \in \{0,1\}^q$ is a binary label vector of $\boldsymbol{x}_i$. For each element $Y_{ij} = 1$ while the label y_j is related to $\boldsymbol{x}_i$; otherwise $Y_{ij} = 0$.

The goal of multi-label learning is to learn a classifier: $h : \mathcal{X} \to \mathcal{Y}$ which can predict a label set for an unseen instance. Firstly, we usually learn a real-valued function $f : \mathcal{X} \times \mathcal{Y} \to R$, and the value of $f(\boldsymbol{x}, y)$ can be seen as the confidence of y being a proper label of $\boldsymbol{x}$. Then, the multi-label classifier $h(\boldsymbol{x})$ can be defined as: $h(\boldsymbol{x}) = \{y | f(\boldsymbol{x}, y) > t, y \in \mathcal{Y}\}$, where $t = 0.5$ is a predefined threshold. Here, we assume f consists of q subfunctions, one for each label, i.e., $f = [f_1, f_2, \cdots, f_q]$. For simplicity, in this study, we apply the linear model for each f_j:

$$f_j(\boldsymbol{x}_i) = \boldsymbol{x}_i \boldsymbol{W}_j, \tag{1}$$

where $\boldsymbol{W}_j = (W_{j1}, W_{j2}, ..., W_{jd}, b_j)^{\mathrm{T}}$ represents the linear regression parameters corresponding to the j-th label. The offset term b_j is expanded into $\boldsymbol{W}_j$ and the constant value 1 is added as an additional dimension for each data $\boldsymbol{x}_i$ $(1 \leqslant i \leqslant n)$.

Considering that each label may carry specific characteristics of its own, we assume that each label is determined by a subset of the original features. According to (1), if $W_{jk} = 0$, the k-th feature has no effect on the discrimination of the j-th label y_j. Only the features corresponding to the non-zero items in $\boldsymbol{W}_j$ are used to discriminate the j-th label, and these features can be viewed as the label-specific features of label y_j. To do this, we apply l_1-regularization on $\boldsymbol{W}_j$ to extract label-specific features. The objective function is defined as:

$$\min_{\boldsymbol{W}} \frac{1}{2} ||\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}||_F^2 + \lambda_1 \sum_{j=1}^{q} ||\boldsymbol{W}_j||_1,$$

where $\boldsymbol{X} \in \mathbb{R}^{n \times (d+1)}$ represents the instances matrix in training data and $\boldsymbol{W} \in \mathbb{R}^{(d+1) \times q}$ is the weight parameters matrix.

3.2 Jointing Label Correlations

During the training process, the correlations among different labels provide additional information for multi-label learning, which has shown to be useful for performance improvement of the classifier[10, 12, 26]. Two labels are called positively correlated if they often belong to one instance at the same time. Previous studies assumed that two related labels y_i and y_j have a large similarity between $\boldsymbol{W}_i$ and $\boldsymbol{W}_j$, which means they tend to select similar features. However, this assumption is in conflict with the learning of label-specific

features. For two related labels y_i and y_j, their label-specific features may be different, and the corresponding parameter vectors $\boldsymbol{W}_i$ and $\boldsymbol{W}_j$ may have different zero items.

In order to reasonably exploit the correlations among different labels, we consider that if labels y_i and y_j are positively correlated, their corresponding functions f_i and f_j should have similar outputs, and vice versa. Thus, we constrain the label correlations on the output matrix $\boldsymbol{XW}$ rather than on the parameter matrix $\boldsymbol{W}$, which can keep the learned label-specific features to some extent. By jointing label-specific features and label correlations, we define the objective function as:

$$\min_{\boldsymbol{W}} \frac{1}{2}||\boldsymbol{XW}-\boldsymbol{Y}||_F^2+\lambda_1\sum_{j=1}^{q}||\boldsymbol{W}_j||_1+\frac{\lambda_2}{4}\sum_{i=1}^{q}\sum_{j=1}^{q}\theta_{ij}||\boldsymbol{XW}_i-\boldsymbol{XW}_j||_2^2,$$

where $\boldsymbol{\theta}\in\mathbb{R}^{q\times q}$ represents the label correlation matrix, which is calculated by cosine similarity, and $\theta_{ij}=\boldsymbol{Y}_{\cdot i}\boldsymbol{Y}_{\cdot j}^{\mathrm{T}}/(||\boldsymbol{Y}_{\cdot i}||||\boldsymbol{Y}_{\cdot j}||)$ denotes the correlation between the i-th label and the j-th label. $\boldsymbol{Y}_{\cdot i}$ is the i-th column of $\boldsymbol{Y}$ and $\boldsymbol{Y}_{\cdot j}$ is the j-th column of $\boldsymbol{Y}$.

3.3 Exploiting the Sample Correlations

In order to further improve the performance of multi-label learning algorithm, LSFCI incorporates instance correlations into model training. However, it calculates instance correlations by common similarity measures, which may not reflect complex relationships among instances. In this study, we propose to exploit the reconstruction relationships among samples which can indicate the underlying structure of the training data. To characterize the reconstruction relationships among samples in features space, our proposed LFCMLL learns a reconstruction coefficient matrix $\boldsymbol{S}\in\mathbb{R}^{n\times n}$, where S_{ij} represents the reconstruction contribution of the j-th instance to the i-th instance. In this study, the coefficient matrix is learned by modeling the relationship between one instance and all the other instances via sparse reconstruction. Thus, coefficient matrix $\boldsymbol{S}$ can be learned by solving the following optimization problem:

$$\min_{\boldsymbol{S}} \frac{1}{2}||\boldsymbol{SX}-\boldsymbol{X}||_F^2+\mu\sum_{i=1}^{n}||\boldsymbol{S}_i||_1 \quad \text{s.t.} \quad \mathrm{diag}(\boldsymbol{S})=0, \tag{2}$$

where the first term is the linear reconstruction error for all instances, and the second term is l_1 norm to control the sparsity of reconstruction coefficients for each instance. The trade-off parameter μ is used to balance the relative importance of each term and $\mathrm{diag}(\boldsymbol{S})=0$ means that any instance has no contribution to reconstruct itself. The optimization problem in (2) can be solved by accelerated proximal gradient method.

We assume that the reconstruction relationships among training samples in feature space should also be maintained in label space. Thus, for each instance, its output can be reconstructed by others based on the reconstruction coefficients in $\boldsymbol{S}$, and the reconstruction error should be as small as possible. To do this, a new regularization term can be defined as:

$$\min_{\boldsymbol{W}} \frac{1}{2}\sum_{i=1}^{n}||\boldsymbol{x}_i\boldsymbol{W}-\boldsymbol{S}_{i\cdot}\boldsymbol{XW}||_2^2,$$

where $\boldsymbol{S}_{i\cdot}=(S_{i1},S_{i2},...,S_{in})$ is the i-th row of coefficient matrix $\boldsymbol{S}$, and $S_{ii}=0$.

Then, the final objective function can be defined as:

$$\min_{\boldsymbol{W}} \frac{1}{2}||\boldsymbol{XW}-\boldsymbol{Y}||_F^2+\lambda_1\sum_{j=1}^{q}||\boldsymbol{W}_j||_1+\frac{\lambda_2}{4}\sum_{i=1}^{q}\sum_{j=1}^{q}\theta_{ij}||\boldsymbol{XW}_i-\boldsymbol{XW}_j||_2^2+\frac{\lambda_3}{2}\sum_{i=1}^{n}||\boldsymbol{x}_i\boldsymbol{W}-\boldsymbol{S}_{i\cdot}\boldsymbol{XW}||_2^2. \tag{3}$$

The first term is the loss function to measure the distance between the predicting label distribution and the ground truth. The second term is an l_1-regularization term to extract the label-specific features. The third term represents that if two labels are positively correlated, the distance between the two predicted label values should be small. The fourth term is used to maintain the reconstruction relationships among training samples in feature space. λ_1, λ_2 and λ_3 are the balance factors.

3.4 Optimization

The objective function in (3) can be represented as:

$$G(\boldsymbol{W})=\frac{1}{2}||\boldsymbol{XW}-\boldsymbol{Y}||_F^2+\lambda_1\sum_{j=1}^{q}||\boldsymbol{W}_j||_1+\frac{\lambda_2}{2}tr(\boldsymbol{XWL}(\boldsymbol{XW})^{\mathrm{T}})+\frac{\lambda_3}{2}||(\boldsymbol{I}-\boldsymbol{S})\boldsymbol{XW}||_F^2, \tag{4}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{\theta}$, and $\boldsymbol{D} = Diag(d_1, d_2, ..., d_q)$ is a dialog matrix, $d_i = \sum_{j=1}^{q} \theta_{ij}$, $\boldsymbol{I} \in \mathbb{R}^{n \times n}$ is a unit matrix. $tr(*)$ means the trace of $*$.

Since (4) is non-smooth due to the non-smoothness of l_1-regularization term, we utilize the accelerated proximal gradient method to solve the minimization of the function $G(\boldsymbol{W})$. The accelerated proximal gradient method is usually applied to solve the non-smooth convex optimization problem, which can be represented by a general optimization framework as follows:

$$\min_{\boldsymbol{W} \in \boldsymbol{\mathcal{H}}} G(\boldsymbol{W}) = s(\boldsymbol{W}) + g(\boldsymbol{W}), \tag{5}$$

where $\mathcal{H}$ represents Hilbert space, $s(\boldsymbol{W})$ is convex and smooth, and $g(\boldsymbol{W})$ is convex and typically non-smooth. $s(\boldsymbol{W})$ is further Lipschitz continuous, that is, $s(\boldsymbol{W})$ satisfies the following condition:

$$||\nabla s(\boldsymbol{W}_i') - \nabla s(\boldsymbol{W}_i)||_2 \leqslant Lip||\Delta \boldsymbol{W}||_2 (\forall \boldsymbol{W}_i', \boldsymbol{W}_i),$$

where $1 \leqslant i \leqslant q$, $\Delta \boldsymbol{W} = \boldsymbol{W}_i' - \boldsymbol{W}_i$, and Lip is the Lipschitz constant.

According to (4) and (5), $s(\boldsymbol{W})$ and $g(\boldsymbol{W})$ can be represented as:

$$\begin{aligned} s(\boldsymbol{W}) &= \frac{1}{2}||\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}||_F^2 + \frac{\lambda_2}{2} tr(\boldsymbol{X}\boldsymbol{W}\boldsymbol{L}(\boldsymbol{X}\boldsymbol{W})^{\mathrm{T}}) + \\ &\quad \frac{\lambda_3}{2}||(\boldsymbol{I} - \boldsymbol{S})\boldsymbol{X}\boldsymbol{W}||_F^2, \\ g(\boldsymbol{W}) &= \lambda_1 \sum_{j=1}^{q} ||\boldsymbol{W}_j||_1. \end{aligned}$$

Thus, we can get the gradient of $s(\boldsymbol{W})$ with respect to $\boldsymbol{W}$ as:

$$\nabla s(\boldsymbol{W}) = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}) + \lambda_2 \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{L} + \boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}\boldsymbol{W},$$

where $\boldsymbol{E} = (\boldsymbol{I} - \boldsymbol{S})\boldsymbol{X}$.

Then, we have

$$\begin{aligned} &||\nabla s(\boldsymbol{W}') - \nabla s(\boldsymbol{W})||_F^2 \\ &= ||\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\Delta\boldsymbol{W} + \lambda_2 \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\Delta\boldsymbol{W}\boldsymbol{L} + \lambda_3 \boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}\Delta\boldsymbol{W}||_F^2 \\ &\leqslant 3(||\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\Delta\boldsymbol{W}||_F^2 + ||\lambda_2 \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\Delta\boldsymbol{W}\boldsymbol{L}||_F^2 + \\ &\quad ||\lambda_3 \boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}\Delta\boldsymbol{W}||_F^2) \\ &\leqslant 3(||\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}||_2^2||\Delta\boldsymbol{W}||_F^2 + \\ &\quad ||\lambda_2 \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}||_2^2||\boldsymbol{L}||_2^2||\Delta\boldsymbol{W}||_F^2 + \\ &\quad ||\lambda_3 \boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}||_2^2||\Delta\boldsymbol{W}||_F^2) \\ &= 3(||\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}||_2^2 + ||\lambda_2 \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}||_2^2||\boldsymbol{L}||_2^2 + \\ &\quad ||\lambda_3 \boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}||_2^2)||\Delta\boldsymbol{W}||_F^2. \end{aligned}$$

Therefore, the Lipschitz constant can be calculated by:

$$\begin{aligned} &Lip \\ &= \sqrt{3(||\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}||_2^2 + ||\lambda_2 \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}||_2^2||\boldsymbol{L}||_2^2 + ||\lambda_3 \boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}||_2^2||)}. \end{aligned}$$

Considering the second order Taylor series of $s(\boldsymbol{W})$ at the current estimate of the parameter vector $\boldsymbol{W}^{(t)}$:

$$\begin{aligned} s(\boldsymbol{W}) &\cong s(\boldsymbol{W}^{(t)}) + < \nabla s(\boldsymbol{W}^{(t)}), \boldsymbol{W} - \boldsymbol{W}^{(t)} > + \\ &\quad \frac{1}{2}||\boldsymbol{W} - \boldsymbol{W}^{(t)}||_F^2 \\ &= \frac{Lip}{2}||\boldsymbol{W} - (\boldsymbol{W}^{(t)} - \frac{1}{Lip}\nabla s(\boldsymbol{W}^{(t)}))||_F^2 + \\ &\quad const, \end{aligned} \tag{6}$$

where $const$ is a constant unrelated to $\boldsymbol{W}$, and $< \cdot, \cdot >$ represents the inner product. The minimum value of (6) can be obtained on $\boldsymbol{W}^{(t+1)}$:

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \frac{1}{Lip}\nabla s(\boldsymbol{W}^{(t)}).$$

Then, we can get the optimal solution of $\boldsymbol{W}$ iteratively by:

$$\boldsymbol{W}^{(t+1)} = \arg\min_{\boldsymbol{W}} \frac{Lip}{2}||\boldsymbol{W} - \boldsymbol{Z}||_F^2 + g(\boldsymbol{W}), \tag{7}$$

where $\boldsymbol{Z} = \boldsymbol{W}^{(t)} - \frac{1}{Lip}\Delta s(\boldsymbol{W})$. The previous work [27] has shown that setting $\boldsymbol{W}^{(t)} = \boldsymbol{W}^{(t)} + \frac{b^{(t-1)}-1}{b^{(t)}}(\boldsymbol{W}^{(t)} - \boldsymbol{W}^{(t-1)})$ for a sequence b^t satisfying $(b^t)^2 - b^t \leqslant (b^{t-1})^2$ can improve the convergence rate to $O(t^{-2})$, and $\boldsymbol{W}^{(t)}$ is the result of $\boldsymbol{W}$ at the t-th iteration.

The closed solution of (7) can be calculated by a soft threshold method which is defined as:

$$W_{ij}^{(t+1)} = \begin{cases} Z_{ij}^{(t)} - \lambda_1/Lip, & \text{if } \lambda_1/Lip < Z_{ij}^{(t)}, \\ 0, & \text{if } |Z_{ij}^{(t)}| \leqslant \lambda_1/Lip, \\ Z_{ij}^{(t)} + \lambda_1/Lip, & \text{if } Z_{ij}^{(t)} < -\lambda_1/Lip. \end{cases}$$

The similar procedure can be used to solve the optimization problem of (2). And, to satisfy the constraint $\mathrm{diag}(\boldsymbol{S}) = 0$, we set all diagonal elements in $\boldsymbol{S}$ to 0 after each update.

The above procedure of joint label-specific features and correlation information for multi-label learning is summarized in Algorithm 1.

4 Experiments

4.1 Evaluation Measures

In this study, we utilize five commonly used multi-label evaluation measures to evaluate the performance of each comparison algorithm: Hamming loss, ranking

Algorithm 1. Multi-Label Learning Algorithm of LFCMLL

Input: the training set $D = \{\boldsymbol{x}_i, \boldsymbol{Y}_i\}_{i=1}^n$, parameters λ_1, λ_2, λ_3, γ, and the converagence criterion ξ

Output: the regression parameters matrix $\boldsymbol{W}$

1: Initialize the parameters;
2: b_0, $b_1 \leftarrow 1$, $\boldsymbol{W}_0$, $\boldsymbol{W}_1 \leftarrow (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \gamma \boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$;
3: $t \leftarrow 0$;
4: Calculate the reconstruction coefficient matrix $\boldsymbol{S}$ according to (2);
5: while not converged do
6: $\boldsymbol{W}^{(t)} = \boldsymbol{W}^{(t)} + \frac{b^{(t-1)}-1}{b^{(t)}}(\boldsymbol{W}^{(t)} - \boldsymbol{W}^{(t-1)})$;
7: $\boldsymbol{Z} = \boldsymbol{W}^{(t)} - \frac{1}{Lip}\Delta s(\boldsymbol{W})$;
8: $W_{ij}^{(t+1)} = \begin{cases} Z_{ij}^{(t)} - \lambda_1/Lip, & \text{if } \lambda_1/Lip < Z_{ij}^{(t)}, \\ 0, & \text{if } |Z_{ij}^{(t)}| \leqslant \lambda_1/Lip, \\ Z_{ij}^{(t)} + \lambda_1/Lip, & \text{if } Z_{ij}^{(t)} < -\lambda_1/Lip; \end{cases}$
9: $b^{(t+1)} \leftarrow \frac{1+\sqrt{4((b^{(t)})^2+1)}}{2}$;
10: end
11: Return $\boldsymbol{W}$;

loss, one error, coverage, and average precision [9, 10]. As mentioned before, given a test dataset $T = \{(\boldsymbol{x}_i, \boldsymbol{Y}_i)\}_{i=1}^t$, where $\boldsymbol{Y}_i \in \{0,1\}^q$ represents the true label set of the instance $\boldsymbol{x}_i$. The predictive label set of the instance $\boldsymbol{x}_i$ is represented by $h(\boldsymbol{x}_i)$, and $f(\boldsymbol{x}_i, y)$ outputs its real-valued predictive of the label y. The definitions of evaluation measures are shown as follows.

• Hamming loss:

$$hloss(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{q}|h(\boldsymbol{x}_i)\Delta \boldsymbol{Y}_i|.$$

Here, Δ stands for the symmetric difference between two sets. The Hamming loss evaluates the proportion of misclassified instance-label pairs. For example, a relevant label is missed or an irrelevant is predicted.

• Ranking loss:

$$rloss(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\boldsymbol{Y}_i||\overline{\boldsymbol{Y}_i}|}|\{(y', y'')|f(\boldsymbol{x}_i, y') \leqslant f(\boldsymbol{x}_i, y''), (y', y'') \in \boldsymbol{Y}_i \times \overline{\boldsymbol{Y}_i}\}|.$$

This measure evaluates the proportion of incorrectly ordered label pairs, i.e., an irrelevant label yields larger output value than a relevant label.

• One error:

$$er(f) = \frac{1}{n}\sum_{i=1}^{n}[\![[\arg\max_{y\in\mathcal{Y}} f(\boldsymbol{x}_i, y)] \notin \boldsymbol{Y}_i]\!].$$

This measure evaluates the proportion of test samples whose top-ranked label is not in the relevant label set. Here, $[\![[\pi]]\!]$ returns 1 if predicate π holds, and 0 otherwise.

• Coverage:

$$cov(f) = \frac{1}{n}\sum_{i=1}^{n}\max_{y\in\boldsymbol{Y}_i} rank_f(\boldsymbol{x}_i, y) - 1.$$

This measure evaluates the number of steps needed to move down the ranked label list so as to cover all relevant labels of the test samples. Here, $rank_f(\boldsymbol{x}_i, y)$ returns the rank of class label y within label space $\mathcal{Y}$ according to the descending order specified by $f(\boldsymbol{x}, \cdot)$. In this study, the coverage is normalized by the number of class labels.

• Average precision:

$$avgprec(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\boldsymbol{Y}_i|} \times \sum_{y\in\boldsymbol{Y}_i}\frac{|\{y'|rank_f(\boldsymbol{x}_i, y') \leqslant rank_f(\boldsymbol{x}_i, y), y' \in \boldsymbol{Y}_i\}|}{rank_f(\boldsymbol{x}_i, y)}.$$

This measure evaluates the average proportion of labels ranked higher than a relevant label $y \in \boldsymbol{Y}_i$ that are also relevant.

For Hamming loss, ranking loss, one error, and coverage, the smaller the value, the better the generalization performance. For the average precision, the larger the value, the better the performance.

4.2 Experimental Setup

We conduct extensive experiments on various multi-label benchmark datasets to validate the effectiveness of LFCMLL, in comparison with other state-of-the-art approaches.

Those datasets are widely used in the previous studies [11–13, 17]. The characteristics of the datasets are summarized in Table 1.

LFCMLL is compared with the following five multi-label learning algorithms: 1) LSFCI [14], proposing to learn label specific features for each label with consideration of label correlation in label space and instance correlation in feature space simultaneously. The values of the parameters λ_1, λ_2 and λ_3 in LSFCI are selected from $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$. 2) LIFT [11], proposing the idea of label-specific features for multi-label learning. 3) LLSF [12], using both label-specific features and label correlations. The values of the parameters λ_1 and λ_2 in LLSF are tuned from $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$. 4)

MLFE [17], applying the multi-output regression techniques to train the prediction model under the MLFE framework. MLFE is a kind of multi-label learning framework, which enriches the label information by utilizing the structural information of the feature space; parameters β_1, β_2, and β_3 in MLFE are chosen among $\{1, 2, ..., 10\}$, $\{1, 10, 15\}$, and $\{1, 10\}$ respectively. 5) JFSC [13], jointing features selection and classification for multi-label learning based on label correlation and Fisher discriminant-based regularization. The values of the parameters λ_1, λ_2 and λ_3 in JFSC are selected from $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$.

Table 1. Characteristics of 12 Multi-Label Datasets

Dataset	#instance	#dim	#label	Domain
Arts	5 000	462	26	Text
Birds	645	260	19	Audio
Cal500	502	68	174	music
Flags	194	19	7	Image
Genbase	662	1 185	27	Biology
Medical	978	1 449	45	Text
Slashdot	3 782	1 079	22	Text
Rcvlsubset1	6 000	994	101	Text
Rcvlsubset2	6 000	994	101	Text
Rcvlsubset3	6 000	994	101	Text
Rcvlsubset4	6 000	994	101	Text
Rcvlsubset5	6 000	994	101	Text

Note: "#instance" is the number of instances, "#dim" is the feature dimensionality, and "#label" is the total size of the class label set.

The values of the parameters λ_1, λ_2 and λ_3 in LFCMLL are selected among $\{2^{-10}, 2^{-9}, \cdots, 2^{5}\}$. For all the algorithms, the optimal parameters are determined according to the classification results of 5-fold cross-validation on the training set. For performance evaluation, we perform a 10×5-fold cross-validation on each dataset, where the mean metric value and the standard deviation are recorded for each compared algorithm.

4.3 Experimental Results

From Table 2 to Table 6, we summarize the detailed experimental results (the smaller the value, the better the performance in Table 2 to Table 5; the larger the value, the better the performance in Table 6) of six compared algorithms on each dataset, where the best performance among the six comparing algorithms is highlighted in boldface. In these tables, "AvgRank" means the average rank of algorithms.

Across all the 60 configurations (i.e., 5 criteria $\times$ 12 datasets as shown in experimental results), LFCMLL ranks in the 1st place at 40 cases, in the 2nd place at 17 cases, in the 3rd place at 3 cases, and never ranks in other places.

To analyze the relative performance among the comparing algorithms, Friedman test, which is a favorable statistical test for comparisons of more than two algorithms over multiple datasets [28], is further examined. Table 7 summarizes the Friedman statistics F_F and the corresponding critical value on each evaluation measure. All evaluation measures have the same critical value. For each evaluation measure, the null hypothesis of indistinguishable performance among the compared algorithms is rejected at 0.05 significance level. Consequently, the post-hoc Bonferroni-Dunn test [28] is applied to test whether our proposed method

Table 2. Experimental Results of Each Comparison Algorithm "Mean ± Std(Rank)" on 12 Datasets in Terms of Hamming Loss

Dataset	LSFCI	LIFT	LLSF	MLFE	JFSC	LFCMLL
Arts	0.0553±0.0001(6)	**0.0530±0.0002(1)**	0.0541±0.0001(5)	0.0537±0.0001(4)	0.0533±0.0001(2)	0.0534±0.0001(3)
Birds	0.0457±0.0007(3)	0.0459±0.0008(4)	0.0511±0.0013(6)	0.0461±0.0006(5)	0.0456±0.0008(2)	**0.0449±0.0006(1)**
Cal500	0.1382±0.0003(5)	0.1379±0.0005(2.5)	0.1379±0.0002(2.5)	0.1390±0.0002(6)	0.1381±0.0004(4)	**0.1372±0.0003(1)**
Flags	0.4713±0.0024(6)	0.2692±0.0065(2)	0.2771±0.0086(5)	0.2749±0.0081(4)	0.2710±0.0075(3)	**0.2688±0.0074(1)**
Genbase	0.0011±0.0001(4)	0.0024±0.0002(6)	0.0010±0.0001(2.5)	0.0010±0.0001(2.5)	0.0014±0.0001(5)	**0.0009±0.0002(1)**
Medical	0.0101±0.0002(2.5)	0.0122±0.0001(6)	0.0101±0.0002(2.5)	0.0111±0.0002(5)	0.0105±0.0002(4)	**0.0098±0.0002(1)**
Slashdot	0.0176±0.0001(6)	0.0164±0.0002(4)	0.0171±0.0001(5)	0.0158±0.0002(3)	**0.0151±0.0001(1)**	0.0156±0.0001(2)
Rcvsubset1	0.0268±0.0005(5)	**0.0262±0.0001(1)**	0.0267±0.0001(4)	0.0270±0.0001(6)	0.0266±0.0000(3)	0.0265±0.0001(2)
Rcvsubset2	0.0234±0.0000(4.5)	0.0235±0.0001(6)	0.0234±0.0001(4.5)	0.0233±0.0001(3)	**0.0228±0.0000(1)**	0.0231±0.0000(2)
Rcvsubset3	0.0229±0.0000(2)	0.0233±0.0000(5.5)	0.0233±0.0000(5.5)	0.0232±0.0001(4)	**0.0226±0.0000(1)**	0.0230±0.0000(3)
Rcvsubset4	**0.0184±0.0000(2)**	0.0191±0.0001(6)	0.0186±0.0001(5)	**0.0184±0.0000(2)**	0.0185±0.0001(4)	**0.0184±0.0000(2)**
Rcvsubset5	0.0229±0.0000(3)	0.0233±0.0001(6)	0.0232±0.0001(5)	0.0230±0.0001(4)	**0.0224±0.0000(1)**	0.0228±0.0000(2)
AvgRank	4.083	4.167	4.375	4.042	2.583	**1.75**

Table 3. Experimental Results of Each Comparison Algorithm "Mean ± Std(Rank)" on 12 Datasets in Terms of Ranking Loss

Dataset	LSFCI	LIFT	LLSF	MLFE	JFSC	LFCMLL
Arts	0.1352±0.0010(6)	**0.1143±0.0007(1)**	0.1314±0.0007(4)	0.1313±0.0011(3)	0.1318±0.0007(5)	0.1211±0.0007(2)
Birds	0.1644±0.0033(4)	0.1920±0.0072(6)	0.1783±0.0066(5)	0.1641±0.0063(2.5)	0.1641±0.0034(2.5)	**0.1611±0.0046(1)**
Cal500	0.1868±0.0010(4)	0.1819±0.0005(2)	0.1842±0.0004(3)	0.2023±0.0013(6)	0.1885±0.0012(5)	**0.1792±0.0004(1)**
Flags	0.4345±0.0099(6)	**0.2169±0.0071(1)**	0.2439±0.0082(5)	0.2369±0.0075(4)	0.2285±0.0061(3)	0.2198±0.0061(2)
Genbase	0.0025±0.0007(4)	0.0041±0.0008(5)	0.0014±0.0004(2)	0.0024±0.0006(3)	0.0044±0.0009(6)	**0.0013±0.0005(1)**
Medical	0.0219±0.0027(4)	0.0276±0.0012(6)	0.0213±0.0022(3)	0.0183±0.0009(2)	0.0241±0.0032(5)	**0.0151±0.0015(1)**
Slashdot	0.0603±0.0010(5)	0.0417±0.0011(2)	0.0601±0.0010(4)	0.0591±0.0012(3)	0.0693±0.0012(6)	**0.0394±0.0012(1)**
Rcvsubset1	0.0507±0.0004(4)	**0.0484±0.0003(1)**	0.0504±0.0002(3)	0.0714±0.0002(6)	0.0546±0.0004(5)	0.0492±0.0004(2)
Rcvsubset2	0.0501±0.0006(2)	0.0510±0.0004(3)	0.0743±0.0008(6)	0.0691±0.0012(5)	0.0534±0.0005(4)	**0.0480±0.0002(1)**
Rcvsubset3	0.0517±0.0003(3)	0.0503±0.0004(2)	0.0726±0.0006(6)	0.0682±0.0006(5)	0.0539±0.0003(4)	**0.0479±0.0004(1)**
Rcvsubset4	0.0379±0.0005(3)	**0.0356±0.0003(1)**	0.0541±0.0008(6)	0.0496±0.0011(5)	0.0411±0.0004(4)	0.0362±0.0002(2)
Rcvsubset5	0.0485±0.0002(3)	**0.0458±0.0004(1)**	0.0703±0.0008(6)	0.0657±0.0005(5)	0.0515±0.0005(4)	0.0466±0.0003(2)
AvgRank	4.000	2.583	4.417	4.125	4.583	**1.417**

Table 4. Experimental Results of Each Comparison Algorithm "Mean ± Std(Rank)" on 12 Datasets in Terms of One Error

Dataset	LSFCI	LIFT	LLSF	MLFE	JFSC	LFCMLL
Arts	0.4536±0.0027(2)	0.4611±0.0037(6)	0.4605±0.0034(5)	0.4539±0.0029(3)	0.4567±0.0028(4)	**0.4482±0.0020(1)**
Birds	0.3530±0.0110(2)	0.4394±0.0098(6)	0.3797±0.0121(5)	0.3595±0.0178(4)	**0.3524±0.0117(1)**	0.3540±0.0144(3)
Cal500	0.1416±0.0010(4.5)	0.1258±0.0069(2)	0.1303±0.0035(3)	0.1735±0.0063(6)	0.1416±0.0063(4.5)	**0.1176±0.0020(1)**
Flags	0.4298±0.0170(6)	0.2329±0.0258(3)	0.2421±0.0170(5)	0.2364±0.0217(4)	0.2107±0.0166(2)	**0.1994±0.0208(1)**
Genbase	0.0023±0.0010(4.5)	**0.0003±0.0006(1)**	0.0033±0.0006(6)	0.0018±0.0010(2.5)	0.0023±0.0010(4.5)	0.0018±0.0006(2.5)
Medical	0.1333±0.0043(3)	0.1618±0.0067(6)	0.1309±0.0057(2)	0.1340±0.0034(4)	0.1420±0.0049(5)	**0.1290±0.0043(1)**
Slashdot	0.0846±0.0013(2)	0.0912±0.0012(5)	0.0851±0.0014(3)	0.0897±0.0012(4)	0.0934±0.0018(6)	**0.0838±0.0011(1)**
Rcvsubset1	0.4259±0.0029(4)	0.4238±0.0033(2)	0.4254±0.0045(3)	0.4407±0.0048(6)	0.4318±0.0028(5)	**0.4206±0.0010(1)**
Rcvsubset2	0.4055±0.0067(2)	0.4349±0.0036(6)	0.4194±0.0039(4)	0.4213±0.0054(5)	0.4130±0.0030(3)	**0.4034±0.0036(1)**
Rcvsubset3	**0.4113±0.0028(1)**	0.4421±0.0045(6)	0.4219±0.0036(4)	0.4234±0.0028(5)	0.4143±0.0034(3)	0.4139±0.0009(2)
Rcvsubset4	0.3281±0.0014(2)	0.3764±0.0048(6)	0.3415±0.0020(4)	0.3449±0.0010(5)	0.3404±0.0034(3)	**0.3272±0.0013(1)**
Rcvsubset5	0.4035±0.0050(2)	0.4381±0.0058(6)	0.4172±0.0042(5)	0.4144±0.0017(4)	0.4075±0.0031(3)	**0.4003±0.0032(1)**
AvgRank	2.917	4.583	4.083	4.375	3.917	**1.375**

Table 5. Experimental Results of Each Comparison Algorithm "Mean ± Std(Rank)" on 12 Datasets in Terms of Coverage

Dataset	LSFCI	LIFT	LLSF	MLFE	JFSC	LFCMLL
Arts	0.2025±0.0011(5)	**0.1739±0.0007(1)**	0.1923±0.0008(3)	0.2023±0.0015(4)	0.2048±0.0010(6)	0.1897±0.0011(2)
Birds	0.1200±0.0026(3)	0.1330±0.0043(6)	0.1306±0.0034(5)	0.1190±0.0033(2)	0.1201±0.0025(4)	**0.1185±0.0030(1)**
Cal500	0.7835±0.0040(4)	0.7546±0.0039(2)	0.7739±0.0022(3)	0.8211±0.0045(6)	0.7865±0.0034(5)	**0.7478±0.0012(1)**
Flags	0.6743±0.0074(6)	**0.5398±0.0056(1)**	0.5607±0.0075(5)	0.5447±0.0064(3)	0.5486±0.0061(4)	0.5424±0.0039(2)
Genbase	0.0128±0.0008(3)	0.0168±0.0013(6)	0.0113±0.0007(2)	0.0136±0.0010(4)	0.0142±0.0009(5)	**0.0108±0.0007(1)**
Medical	0.0332±0.0035(4)	0.0424±0.0018(6)	0.0319±0.0028(3)	0.0296±0.0015(2)	0.0367±0.0043(5)	**0.0247±0.0020(1)**
Slashdot	0.0443±0.0007(4)	0.0385±0.0012(2)	0.0435±0.0009(3)	0.0569±0.0012(5)	0.0665±0.0013(6)	**0.0360±0.0011(1)**
Rcvsubset1	0.1272±0.0007(4)	**0.1210±0.0010(1)**	0.1266±0.0005(3)	0.1672±0.0002(6)	0.1354±0.0008(5)	0.1236±0.0011(2)
Rcvsubset2	0.1240±0.0014(3)	0.1231±0.0009(2)	0.1698±0.0015(6)	0.1593±0.0018(5)	0.1296±0.0010(4)	**0.1188±0.0006(1)**
Rcvsubset3	0.1257±0.0005(3)	0.1205±0.0008(2)	0.1643±0.0012(6)	0.1551±0.0019(5)	0.1288±0.0007(4)	**0.1158±0.0009(1)**
Rcvsubset4	0.0935±0.0009(3)	**0.0860±0.0006(1)**	0.1226±0.0012(6)	0.1137±0.0023(5)	0.0989±0.0009(4)	0.0899±0.0002(2)
Rcvsubset5	0.1225±0.0006(3)	**0.1126±0.0006(1)**	0.1636±0.0014(6)	0.1546±0.0013(5)	0.1273±0.0010(4)	0.1180±0.0006(2)
AvgRank	3.750	2.583	4.250	4.333	4.467	**1.417**

Table 6. Experimental Results of Each Comparison Algorithm "Mean ± Std(Rank)" on 12 Datasets in Terms of Average Precision

Dataset	LSFCI	LIFT	LLSF	MLFE	JFSC	LFCMLL
Arts	0.6224±0.0022(5)	0.6244±0.0018(4)	0.6200±0.0016(6)	0.6257±0.0015(3)	0.6289±0.0016(2)	**0.6343±0.0010(1)**
Birds	0.6568±0.0078(3)	0.5958±0.0087(6)	0.6348±0.0077(5)	0.6527±0.0103(4)	0.6571±0.0073(2)	**0.6582±0.0086(1)**
Cal500	0.5024±0.0015(4)	0.4987±0.0019(5)	0.5043±0.0009(2)	0.4917±0.0017(6)	0.5036±0.0016(3)	**0.5055±0.0007(1)**
Flags	0.6693±0.0081(6)	0.8081±0.0070(3)	0.7915±0.0071(5)	0.7972±0.0073(4)	0.8089±0.0060(2)	**0.8134±0.0071(1)**
Genbase	0.9952±0.0005(3.5)	0.9946±0.0006(5)	0.9957±0.0006(2)	0.9952±0.0008(3.5)	0.9944±0.0009(6)	**0.9968±0.0006(1)**
Medical	0.9015±0.0036(3)	0.8722±0.0034(6)	0.9050±0.0036(2)	0.9002±0.0020(4)	0.8948±0.0044(5)	**0.9079±0.0027(1)**
Slashdot	0.8981±0.0007(2.5)	0.8928±0.0010(4)	0.8981±0.0009(2.5)	0.8865±0.0007(5)	0.8797±0.0015(6)	**0.9008±0.0010(1)**
Rcvsubset1	0.6033±0.0011(3)	0.5964±0.0023(4.5)	0.6047±0.0013(2)	0.5801±0.0014(6)	0.5964±0.0016(4.5)	**0.6089±0.0003(1)**
Rcvsubset2	0.6293±0.0026(2)	0.6073±0.0017(6)	0.6109±0.0020(4)	0.6104±0.0025(5)	0.6189±0.0014(3)	**0.6295±0.0016(1)**
Rcvsubset3	0.6282±0.0015(2)	0.6057±0.0023(6)	0.6065±0.0019(4.5)	0.6065±0.0012(4.5)	0.6132±0.0014(3)	**0.6326±0.0012(1)**
Rcvsubset4	0.7097±0.0012(2)	0.6745±0.0023(6)	0.6900±0.0013(4)	0.6890±0.0002(5)	0.6925±0.0015(3)	**0.7099±0.0006(1)**
Rcvsubset5	0.6319±0.0017(2)	0.6096±0.0022(6)	0.6145±0.0018(5)	0.6152±0.0013(4)	0.6233±0.0018(3)	**0.6346±0.0019(1)**
AvgRank	3.167	5.125	3.667	4.5	3.542	**1.000**

LFCMLL achieves competitive performance against the other compared algorithms.

Table 7. Friedman Statistics F_F in Terms of Each Evaluation Measure and the Critical Value at 0.05 Significance Level

Evaluation Measure	F_F	Critical Value
Hamming loss	5.378	
Ranking loss	10.900	
One error	11.805	2.383
Coverage	8.971	
Average precision	14.776	

Note: The number of compared algorithms $k = 6$, the number of datasets $N = 12$.

Here, LFCMLL is considered as the control algorithm whose average rank difference against the compared algorithm is calibrated with the critical difference (CD). Accordingly, LFCMLL is deemed to have significantly different performance against one compared algorithm if their average ranks differ by at least one CD (CD = 1.967 in this paper: the number of compared algorithms $k = 6$, the number of datasets $N = 12$).

Fig.1 shows the CD diagrams [28] on each evaluation metric, where the average rank of each compared algorithm is marked along the axis (lower ranks to the right). In each sub-figure, any compared algorithm whose average rank is within one CD to that of LFCMLL is connected with a thick line. LFCMLL is deemed to have a significantly different performance from any algorithm which does not connect with LFCMLL. According to Fig.1, we can get the following observations.

• LFCMLL achieves optimal average rank in terms of all evaluation measures.

• LFCMLL significantly outperforms LLSF in terms of all evaluation measures.

• LFCMLL significantly outperforms all the compared algorithms in terms of average precision.

• LFCMLL is comparable to JFSC in terms of hamming loss, comparable to LIFT in terms of ranking loss and coverage, comparable to LSFCI in terms of one error, and significantly outperforms LSFCI, LIFT, LLSF, MLFE and JFSC on all the other cases.

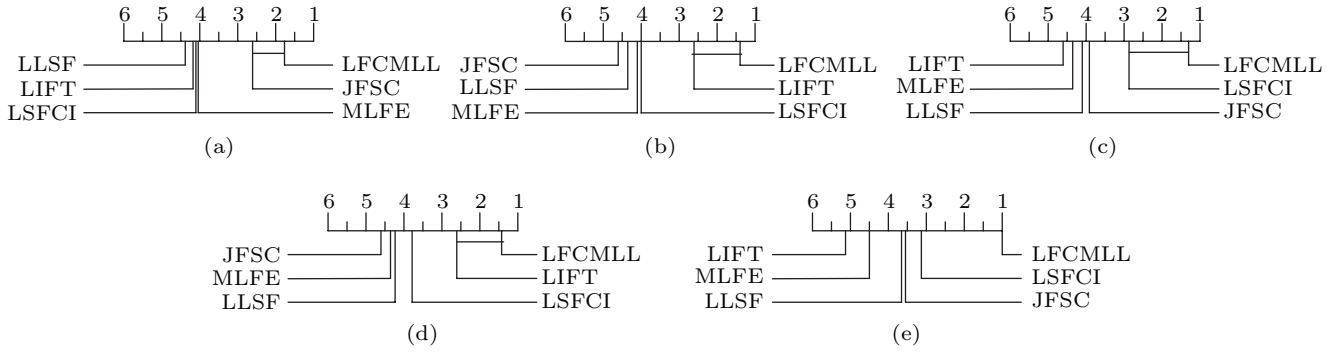


Fig.1. Comparison of LFCMLL (control algorithm) against five compared algorithms with the Bonferroni-Dunn test. Algorithms not connected with LFCMLL in the CD diagram are considered to have significantly different performance from the control algorithm (CD = 1.967 at 0.05 significance level). (a) Hamming loss. (b) Ranking loss. (c) One error. (d) Coverage. (e) Average precision.

4.4 Parameters Analysis

4.4.1 Influence of Parameter μ

Fig.2 shows the effect of varying μ on the flags dataset. As can be seen, when μ is too small, S is not sparse, an instance will be reconstructed by most of the other instances and the noise may be introduced. With the increasing μ, the performance improves. When μ is too large, the underlying structure of the training data is not fully exploited, and the performance starts to get worse. For dataset flags, we can choose the value of μ within a certain range $[2^3, 2^5]$. And the same procedure can be utilized to determine the value of parameter μ for the other datasets.

4.4.2 Influence of Regularization Parameters

In this subsection, we investigate the influence of three parameters in (3) on the experimental results. These parameters include λ_1 (trade-off parameter for the l_1-regularization), λ_2 (trade-off parameter for label correlations), and λ_3 (trade-off parameter for reconstruction relationships). The larger the value of trade-off parameter, the more important the corresponding regularization term. Fig.3 shows the effects of varying model parameters on the four evaluation measures for the flags dataset. The same procedure of parameter sensitivity analysis is used for the other datasets. From Fig.3, we can observe that LFCMLL can get better performance when the values of λ_1, λ_2, and λ_3 are within a certain range ($\lambda_1 \in [2^{-3}, 2^{-1}]$, $\lambda_2 \in [2^{-7}, 2^{-5}]$, $\lambda_3 \in [2^1, 2^3]$).

5 Conclusions

In this study, we proposed a novel multi-label learning algorithm by jointing label-specific features learning and correlation information. Specifically, we assumed that the label-specific features for each label are a subset of the original features. Then, we applied l_1-regularization to extract the label-specific features. Moreover, in order to reasonably utilize the correlations among different labels, we considered that if the

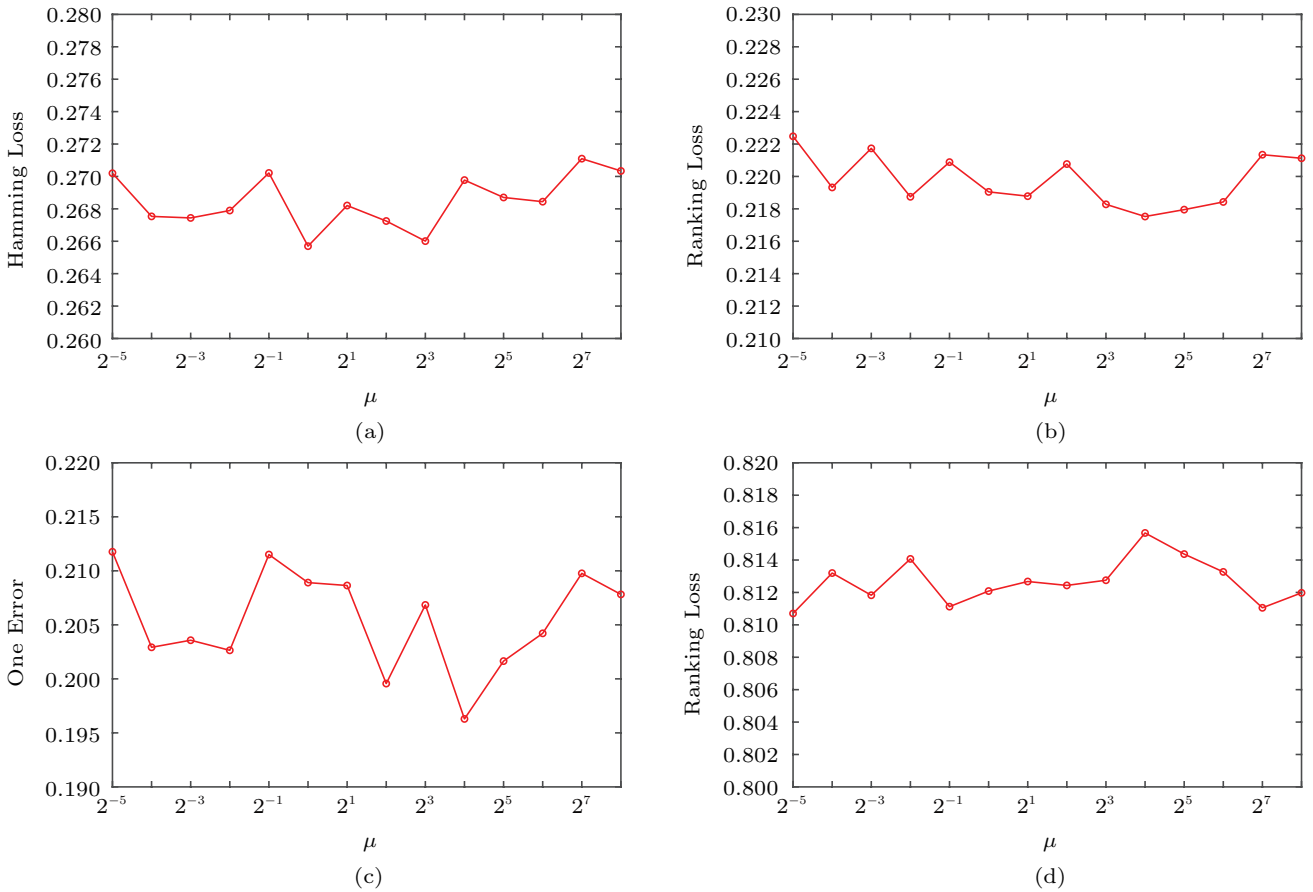


Fig.2. Influence of parameter μ with 4 measures on dataset flags. (a) Hamming loss. (b) Ranking loss. (c) One error. (d) Average precision.

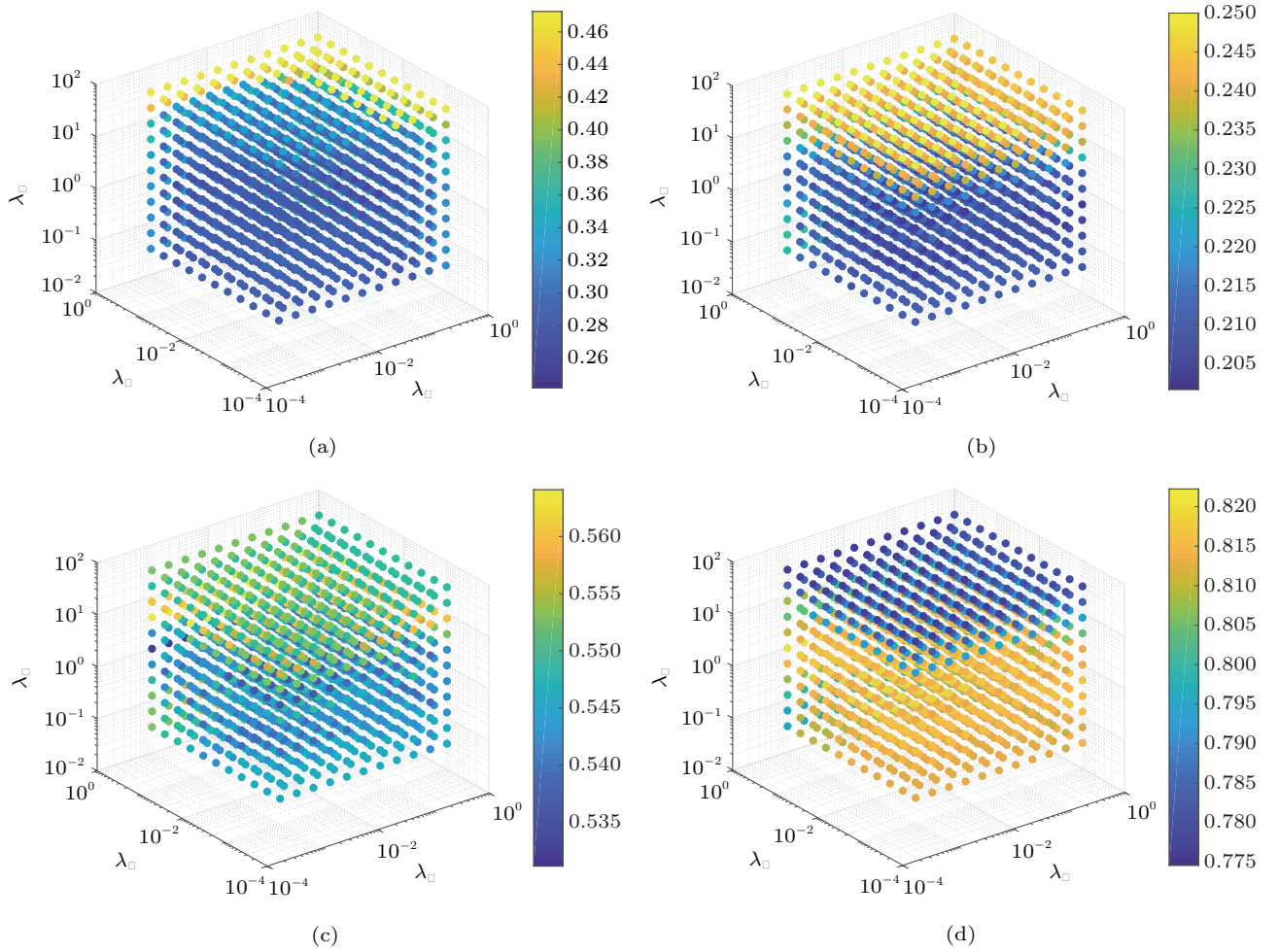


Fig.3. Influence of λ_1, λ_2 and λ_3 with 4 measures on dataset flags. (a) Hamming loss. (b) Ranking loss. (c) Coverage. (d) Average precision.

two labels are positively correlated, their corresponding real-valued functions should have similar outputs. To further improve the performance of the model, we captured the reconstruction relationships among samples which can characterize the underlying structure of the training data, and integrate the reconstruction relationships to our model for improving the learning of label-specific features. At last, we carried out extensive experiments to validate the effectiveness of our algorithm in comparison with other state-of-the-art approaches on various datasets. The experimental results show that learning with label-specific features and correlation information can significantly improve the multi-label classification performance.

References

[1] He Z Y, Wu J, Lv P. Multi-label text classification based on the label correlation mixture model. *Intelligent Data Analysis*, 2017, 21(6): 1371-1392.

[2] Kazawa H, Izumitani T, Taira H *et al.* Maximal margin labeling for multi-topic text categorization. In *Proc. the 18th Annual Conference on Neural Information Processing Systems*, December 2004, pp.649-656.

[3] de Almeida A M G, Ricardo C, Paraiso E C *et al.* Applying multi-label techniques in emotion identification of short texts. *Neurocomputing*, 2018, 320: 35-46.

[4] Li Y, Song Y, Luo J. Improving pairwise ranking for multi-label image classification. In *Proc. the 30th IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp.1837-1845.

[5] Tan M, Shi Q, van den Hengel A *et al.* Learning graph structure for multi-label image classification via clique generation. In *Proc. the 28th IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp.4100-4109.

[6] Sun F, Tang J, Li H *et al.* Multi-label image categorization with sparse factor representation. *IEEE Transactions on Image Processing*, 2014, 23(3): 1028-1037.

[7] Trohidis K, Tsoumakas G, Kalliris G *et al.* Multi-label classification of music into emotions. In *Proc. the 9th Interna-*

tional Conference on Music Information Retrieval, September 2008, pp.325-330.

[8] Wu B, Zhong E, Horner A *et al.* Music emotion recognition by multi-label multi-layer multi-instance multi-view learning. In *Proc. the 22nd ACM International Conference on Multimedia*, November 2014, pp.117-126.

[9] Zhang M L, Zhou Z H. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 2014, 26(8): 1819-1837.

[10] Zhou Z H, Zhang M L. Multi-label learning. In *Encyclopedia of Machine Learning and Data Mining*, Sammut C, Webb G (eds.), Springer, 2016.

[11] Zhang M L, Wu L. Lift: Multi-label learning with label-specific features. In *Proc. the 22nd International Joint Conference on Artificial Intelligence*, July 2011, pp.1609-1614.

[12] Huang J, Li G, Huang Q *et al.* Learning label specific features for multi-label classification. In *Proc. the 15th IEEE International Conference on Data Mining*, November 2015, pp.181-190.

[13] Huang J, Li G, Huang Q *et al.* Joint feature selection and classification for multilabel learning. *IEEE Transactions on Cybernetics*, 2018, 48(3): 876-889.

[14] Han H, Huang M, Zhang Y *et al.* Multi-label learning with label specific features using correlation information. *IEEE Access*, 2019, 7: 11474-11484.

[15] Elisseeff A, Weston J. A kernel method for multi-labelled classification. In *Proc. the 15th Annual Conference on Neural Information Processing Systems*, December 2001, pp.681-687.

[16] Tsoumakas G, Katakis I, Vlahavas I. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 2011, 23(7): 1079-1089.

[17] Zhang Q W, Zhong Y, Zhang M L. Feature-induced labeling information enrichment for multi-label learning. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, February 2018, pp.4446-4453.

[18] Zhang J, Li C, Cao D *et al.* Multi-label learning with label-specific features by resolving label correlations. *Knowledge-Based Systems*, 2018, 159: 148-157.

[19] Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000, 290(5500): 2323-2326.

[20] Read J, Pfahringer B, Holmes G *et al.* Classifier chains for multi-label classification. *Machine Learning*, 2011, 85(3): 333-359.

[21] Boutell M R, Luo J, Shen X *et al.* Learning multi-label scene classification. *Pattern Recognition*, 2004, 37(9): 1757-1771.

[22] Furnkranz J, Hüllermeier E, Mencia E L *et al.* Multi-label classification via calibrated label ranking. *Machine Learning*, 2008, 73(2): 133-153.

[23] Zhang M L, Zhang K. Multi-label learning by exploiting label dependency. In *Proc. the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 2010, pp.999-1008.

[24] Xu S, Yang X, Yu H *et al.* Multi-label learning with label-specific feature reduction. *Knowledge-Based Systems*, 2016, 104: 52-61.

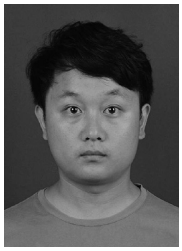
[25] Yan Y, Li S, Yang Z *et al.* Multi-label learning with label-specific feature selection. In *Proc. the 24th International Conference on Neural Information Processing*, November 2017, pp.305-315.

[26] Huang S J, Zhou Z H. Multi-label learning by exploiting label correlations locally. In *Proc. the 26th AAAI Conference on Artificial Intelligence*, July 2012, pp.949-955.

[27] Lin Z, Ganesh A, Wright J *et al.* Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. Technical Report, University of Illinois at Urbana-Champaign, 2009. https://www.ideals.illinois.edu/bitstream/handle/2142/74352/B40-DC_246.pdf?sequence=2&isAllowed=y, Dec. 2019.

[28] Demisar J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 2006, 7(1): 1-30.

Xiu-Yi Jia received his Ph.D. degree in computer science from Nanjing University, Nanjing, in 2011. He is currently an associate professor of School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing. His recent research focuses on machine learning, granular computing, and data mining.



Sai-Sai Zhu received his B.S. degree in software engineering from Anhui University of Technology, Maanshan, in 2017. Currently, he is a Master student in computer technology, Nanjing University of Science and Technology, Nanjing. His research interests include multi-label learning and data mining.

Wei-Wei Li received her Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics, Nanjing, in 2016. She is currently an assistant professor of Nanjing University of Aeronautics and Astronautics, Nanjing. Her research interests include machine learning, software data mining and knowledge engineering.