IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 March 2024, accepted 1 June 2024, date of publication 7 June 2024, date of current version 17 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3411095

RESEARCH ARTICLE

Learning Common and Label-Specific Features for Multi-Label Classification With Missing Labels

RUNXIN LI[1,2], ZEXIAN OUYANG[2], ZHENHONG SHANG[2], LIANYIN JIA[2], AND XIAOWU LI[2]
[1]Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming 650500, China
[2]Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

Corresponding author: Xiaowu Li (lxwlxw66@kust.edu.cn)

This work was supported in part by the Open Fund of Yunnan Key Laboratory of Computer Technology Applications, and in part by the National Natural Science Foundation of China under Grant 12063002 and Grant 62262035.

ABSTRACT Multi-label learning is a subfield of machine learning that addresses the issue of each instance belonging to numerous class labels at the same time. However, in some real applications, we can only receive a partial set of labels for each instance due to the difficulty and high cost of labeling data. The vast majority of existing multi-label classification methods on missing labels rely on first- or second-order label correlation learning to fill in the original label space while building multi-label learning models with label-specific features; nevertheless, the single label correlation learning mechanism used in these methods is insufficient to maintain the consistency of the feature-label space. To address this issue, we propose the CLSML approach, which incorporates higher-order label correlation learning constraints in the classifier training model to complete missing labels while training the classifier. In addition, to improve the consistency of the feature-label space, we develop a two-stage second-order label correlation learning technique based on cosine similarity to further confine the label output. Furthermore, we employ the l_1-norm regularizer to learn label-specific feature representations, followed by the $l_{2,1}$-norm regularizer to constrain the row sparsity of the classification matrix and select label-common features. Experimental results comparing ten cutting-edge multi-label learning algorithms with missing labels on fourteen multi-label benchmark datasets demonstrate the effectiveness of our suggested approach.

INDEX TERMS Multi-label learning, label correlations, missing labels, label-specific feature.

I. INTRODUCTION

In recent years, multi-label learning (MLL) play an increasingly important role in our real-world applications. It tackles with the issue that an instance in real-world might have multiple semetic meanings simultaneously, which the single-label learning cannot solve. A multi-label learning algorithm's main goal is typically predicting a set of correlated labels for an unknown instance. In the past few years, multi-label learning have been applied in a plenty of domains, such as information retrieval [1], text categorization [2], image recognition [3], music emotion classification [4], biological informatics [5] and so on.

Among all method of multi-label learning, it could be usually categorized into Problem Transformation (PT) approaches and Algorithm Adaption (AA) approaches. Problem Transformation (PT), as an intuitive and theoretically simple approach, transforms a multi-label classification problem into either one or more single-label subproblems, which could be solved by a single-label algorithm directly. The first-order approach Binary Relevance (BR) [3], which fail to consider the correlation information among different labels, is the representative approach of PT. Internally adapted to multi-label scenarios, algorithm adaptation (AA) approaches train classifiers by learning the intrinsic correlations between instance features and features, features and class labels, and class labels and class labels. Because label correlations characterize the dependency relationships between different classes of labels, effective label correlations use

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

can considerably improve label classification performance. As a result, current multi-label classification approvals differ in how they benefit from label correlations, such as the second order strategy, which measures pairwise dependencies between labels, and the high order strategy, which evaluates correlations between classes and within classes all labels.

Although label correlations estimations have been demonstrated to improve the effectiveness of multi-label classification, there are certain potential issues that must be handled in practice. The most significant is that, due to high labeling costs, it is difficult in many real-world applications to annotate enough labeled instances and collect complete labeling information for each instance. This may result in inaccurate and incorrect label correlations calculations. As such, building an effective label filling mechanism to datasets with missing labels is required. To accomplish this purpose, it is logical to devise a special label correlation calculation method that uses existing labels to fill in unknown labels via the correlation relationship between labels. Several such strategies, such as [6], [7], [8], [9], and [10], have been proposed. These approaches are classified into two categories. One approach is to fulfill the labels first, as shown in [6] and [10]. Another approach is to incorporate label recovery into training models, as illustrated in [7], [8], and [9].

It is important to note that in multi-label learning, an instance's label is determined by its features, and each class label could potentially be identified by the most relevant and discriminative features associated with its own. For example, the features of smell and taste might be used to detect whether it is vinegar, which could be regarded label-specific features of the vinegar. As consequently, feature selection and extraction have become a hot topic of research in multi-label learning and are being embedded in the training of multi-label classifiers [7], [8], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21].

Following on the LSML [8], we include learning of the label correlation matrix in model training and use it to fill in missing labels while learning label-specific features using the l_1-norm regularizer. However, in [8], the learned label correlations are employed to constrain the label-specific features under the assumption that if two labels are strongly correlated, they will share similar label-specific features. Unfortunately, this assumption might not necessarily be true in some cases, as seen in [11]. For this reason, at the data preprocessing stage, we develop a particular label correlation estimation employing the missing label set in advance, similar to our proposed SMLMFC approach [10]. After recovering unknown labels, the model's label outputs are constrained by the cosine similarity of the class labels, under the assumption that two strongly related labels will yield similar outputs [12].

Furthermore, many existing MLL algorithms only use label-specific features while failing to consider common features, which are important in multi-label learning. Common features, which are a subset of features shared by all labels, indicate common discrimination for all labels, whereas label-specific features are discriminative for different labels. The two kinds of features can be beneficial in improving classification performance. In this research, we therefore propose CLSML, a novel strategy to learning common and label-specific features simultaneously for multi-label classification with missing labels. The following is a summary of this paper's contributions.

- Integrating label correlation matrix learning into model training with the goal of recovering unknown labels while developing label classifiers. To use label correlation to constrain label output, a particular calculation of label similarity estimation is designed during the data preprocessing stage by employing missing label sets to constrain label output.
- In our approach, the l_1-norm and $l_{2,1}$-norm regularizers are concurrently incorporated to choose label-specific and common features, respectively.
- Experiment results on fourteen multi-label benchmark datasets show that our proposed method outperforms state-of-the-art multi-label learning algorithms in multi-label classification with missing labels.

II. RELATED WORK

Feature learning provides a new study direction for multi-label classification by not only reducing feature size but also removing irrelevant and redundant feature interference, hence boosting classifier performance. Existing approaches include label-specific feature learning, which learns a low-dimensional feature representation for each category label, and common feature learning, which learns features that are comparable to all labels. Furthermore, the incompleteness of training labels has a significant impact on the performance of multi-label learning systems. As consequently, many approaches for mitigating classifier performance loss due to missing labels have been proposed. We summarize label-specific and common feature learning, as well as multi-label learning algorithms for missing labels, in the sections that follow.

A. COMMON AND LABEL-SPECIFIC FEATURES LEARNING

The first multi-label learning approach for label-specific features, multi-label learning with label-specific features [13] (LIFT), handles the multi-label learning problem in two straightforward phases. It conducts a clustering analysis on both positive and negative instances, and then uses the clustering findings to generate the specific features. Second, a classifier family is trained using the produced label-specific features rather than the original features from the training dataset. However, this technique derives label-specific features via label-wise clustering multi-label instances, which does not take label correlation information into account to enhance generalization performance. Following that, numerous multi-label learning algorithms for label-specific features have been proposed in an effort to refine the LIFT algorithm. For example, RFS-LIFT [14], which

incorporates approximation quality to assess the significance of a certain dimension and employs a forward greedy search method, aims to choose a subset of the label-specific features generated by LIFT using the fuzzy rough set approach. This algorithm, however, fails to model label correlations. ML-DFL [15] depends on a new spectral clustering approach called Spectral Instance Alignment (SIA), which can build the similarity matrix and then apply spectral clustering analysis to identify the closely located latent structures between positive and negative instances for each label. LIFTAce [16] adopts clustering ensemble techniques to more robustly and effectively detect label-specific features. It not only relies on a single clustering process run over the training data for each class label, but also consolidates an ensemble of clustering results to exploit label correlations for label-specific feature learning. JFSC [17] integrates sparse feature selection and multi-label classification into a single framework capable of selecting the most discriminative features for each label and learning an effective classification model. It also includes a Fisher discriminant-based regularization term to minimize the inner-class distance and maximize the intra-class distance for each label.

In comparison to the preceding techniques, Huang et al. presented the LLSF [18], which is based on the idea that each label is only connected with a subset of the original feature set. And it uses the linear regression model to pick label-specific features. This technique has gained popularity since it can readily discover which features are label-specific by modeling the coefficent matrix. Many improved multi-label learning algorithms for label-specific features have evolved based on LLSF. Huang et al. [19] developed the LLSFDL, which incorporates the label matrix predicted by LLSF into the training set and subsequently learns both label-specific feature and class-dependent labels using sparse stacking. The LSML [8] approach combines missing label set recovery and label-specific feature learning into a single framework. The incomplete label data is then deployed to learn high-order label correlations, which are then used to supplement the incomplete label matrix and guide the construction of multi-label classification models by learning label-specific data representation for each class label iteratively.

The approaches described above solely focus on label-specific feature learning while disregarding common features, which are equally crucial for enhancing classifier performance. SCMFS [20] exploits Coupled Matrix Factorization to extract the shared common features by taking into account the comprehensive data information between the feature matrix and the label matrix. To minimize the negative effects of incomplete label information in detecting label correlations, the MIFS [21] decomposes the multi-label information into a low-dimensional space, which is then utilized for leading the process of selecting common features. FSLCLC [9], introduced by Jiang et al., exploits low-rank matrix factorization on the sparse sample-label association matrix to compress labels and recover missing labels in the compressed label space. In a joint objective function with sparsity regularization, it recovers the missing labels while also selecting common features. In CLML [11], common and label-specific features are investigated simultaneously in a framework, and the l_1-norm and $l_{2,1}$-norm regularizers are used in selecting the corresponding features, resulting in improved classification performance. In fact, numerous research have demonstrated that l_1-norm and $l_{2,1}$-norm are advantageous for extracting and identifying label-specific features and shared common features, respectively, thereby enhancing the performance of multi-label classification.

B. MULTI-LABEL LEARNING WITH MISSING LABELS

Many applications involving multi-label classification, especially when the label space is too big, make it difficult to mark all of the class labels, and only a subset of labels may be observed. Label incompleteness or missingness usually has a negative impact on the performance of multi-label learning algorithms. To be more explicit, it may affect the label correlations learned from the label set of the training data, which will further influence class imbalance and decision functions. As a result, how to effectively carry out multi-label learning with partial labels has become an urgent problem to be solved. Many techniques to addressing this problem have been offered in recent years, which can be classified into two categories:

- Pre-processing techniques attempt to recover the incomplete label matrix before to learning a multi-label classifier, and then make use of the newly recovered label sets to train a multi-label classifier. Maxide [22] speedups matrix completion with two side information matrices, and assumes that the target matrix and the side information matrices share the same latent information. MLMG [23] recovers the incomplete label matrix based on a mixed graph that integrates instance-level label similarity and class co-occurrence as undirected edges and semantic hierarchy between class labels as directed edges. FastTag [24] relies on the assumption that missing labels that have been observed can potentially be linearly translated into complete labels that have not been observed, and that this linear mapping can then be learned using the idea of training a denoising auto-encoder. In SMLMFC [10], it suggested a semi-supervised multi-label learning technique (SMLMFC). Specifically, SMLMFC develops a novel two-stage label correlation approach based on the missing label set to restory missing labels. SMILE [25] first employs a predefined label correlation matrix estimated from partially labeled cases to fill in the missing labels of training instances. The known labels and replenished labels, as well as unlabeled instances, are then used to train a graph-based semi-supervised linear classifier.
- Synchronized techniques, which combine label matrix recovery and multi-label classifier construction, seek to recover the missing label while simultaneously training a multi-label classifier. However, the missing label entries are typically unknown in advance. Weak

label learning [26] (WELL), which first derives a similarity matrix from the feature space, and on the assumption that similar instances frequently share the same characteristic of linked labels, the prediction of missing labels is embedded while the classifier is learning. Large-scale multi-label learning with missing labels(LEML) [27] directly addresses the missing label problem by considering the multi-label problem in a generic empirical risk minimization (ERM) paradigm. GLOCAL [28] solves multi-label learning with missing labels by modeling global and local label correlations, learning a latent label representation, and optimizing label manifolds. MLLRC [29] was proposed by Xu et al. It adopts a low rank structure to capture complex label correlations and integrates the framework that learns the supplemental label matrix by exploiting label correlations while simultaneously training the multi-label model. MLR-GL [30] provide a multi-label learning framework based on the idea of label ranking. To deal with missing labels, it extends multi-label ranking by employing the group lasso technique to combine errors in ranking the assigned classes against the unassigned classes. Furthermore, by penalizing ranking errors selectively, it may be able to determine which unassigned class is truly the missing correct class assignment. The label enrichment matrix is used in HNOML [31] to explore the underlying correlations between distinct classes in order to recover the missing label. The ML-LEML [32] algorithm extracts the low-dimensional shared subspace to exploit latent correlations between features and labels, and then estimates missing labels using low rank and sparse features. LSML [8] learns high-order label correlations from incomplete training data and then applies them to supplement the incomplete label matrix, which is then used to drive the formulation of multi-label classification models by iteratively learning label-specific data representation for each class label. MLLCRS-ML [33] not only considers label-specific features but also utilize the structural property of data and pairwise label correlation (both positive and negative label correlations) to replenish the missing labels. Utilizing the class imbalance sensitive weights and auxiliary label correlations, CIMML [34] introduce a weighted squared loss function with discriminatory label weights which guides the missing label completion and learns the multi-label classifier efficiently. the proposed model uses missing label sensitive discriminator label-specific weights for multi-label learning with missing labels. It also utilizes label correlations and instance similarity for missing label recovery and building classifier for new data. LRMML [35] proposes a unified framework to capture the label correlations, which utilize both auxiliary label matrix and the low rank constraints on estimated labels. Besides, it also enforces maximal separation among different label subspaces for better label differentiation.

TABLE 1. Definitions of symbols.

Symbol	Definition
W	Coefficient matrix $W = [w^1, w^2, ..., w^l] \in \mathbb{R}^{d\times l}$;
w_i	The i-th row of W;
X	Input instance matrix, denoted as $X = [x_1, x_2, ..., x_n]^T \in \mathbb{R}^{n\times d}$;
x_i	The i-th instance, namely the i-th row of input data matrix X;
Y	Label matrix, denoted as $Y = [y_1, y_2, ..., y_n]^T \in \mathbb{R}^{n\times l}$;
y_i	The label set of x_i, namely the i-th row of label matrix Y;
S	The cosine correlation matrix resulting from supplementing the label set;
C	The label correlation matrix obtained through training;
L_1	The $l \times l$ label Laplacian matrix of S.

It recovers missing labels and trains the classifier by building an auxiliary label matrix and imposing low and high rank constraints on label subspace. To deal with the issues emerging from incomplete labels and high-dimensional input space, SGMML [36] propose a multi-label learning approach based on identifying the label-specific features and constraining them with a sparse global structure. The sparse structural constraint helps maintain the typical characteristics of the multi-label learning data. Besides, it also constructs supplementary label correlations to assist missing label recovery as part of the optimization problem, and learns label-specific features and constrain them with sparse global structure for multi-label learning with missing labels.

III. PROPOSED METHOD

In this section, we will first discuss how to model multi-label classification methods utilizing both common and label-specific characteristics, and then we will build two forms of label correlation estimation. To augment the partial label matrix, one is incorporated in the model and obtained during model training. The second form of correlation is proposed for constraining label output on the assumption that two class labels are strongly correlated and that the associated output is also similar. This correlation's design is comparable to our proposed SMLMFC [10]approach. Finally, we will go over the suggested CLSML model and accelerated approximate optimization approach in depth.

A. PRELIMINARIES

Let $D = \{(x_i, y_i)\}_{i=1}^n$ be the training data set with n instances, d dimensions and l class labels, where $x_i \in \mathbb{R}^d$ depicts the i-th instance and $y_i \in \{0, 1\}^l$ stands for the corresponding label vector. $y_{ik} = 1$ $(k = 1, 2, \ldots, l)$ means instance x_i is with k-th class label, while $y_{ik} = 0$ $(k = 1, 2, \ldots, l)$ indicates the unknown status of the corresponding class k. Table 1 summarizes the symbol definitions used in this article for clarity.

B. MULTI-LABEL LEARNING WITH COMMON AND LABEL-SPECIFIC FEATURES

Because the linear regression model offers the characteristics of fast training and prediction speed, interpretable results, and straightforward explanation, we chose it as the classifier. Based on the assumption that each class label is only

determined by a subset of specific features from the original feature set of a given data set, we introduce l_1-norm regularizer to extract label-specific features, because l_1-norm could shrink some elements to zero and these label-specific features are determined by the *non-zero* entries of each $w^i \in \mathbb{R}^d$, so it could be utilized to select to label-specific features. Furthermore, the addition of the $l_{2,1}$-norm forces the sparsity of the row vector in W, which is favourable to choosing common features shared by all labels. The proposed model can be expressed as follows:

$$\min_W \frac{1}{2}\|XW-Y\|_F^2 + \lambda_1\|W\|_1 + \lambda_2\|W\|_{2,1}, \quad (1)$$

where W indicates the coefficient matrix obtained in the regression model, and $w^i = [w_{1i}, w_{2i}, \ldots, w_{di}]^T \in \mathbb{R}^d$ indicates the coefficient vector of i-th label, and the discriminative weight of the i-th feature to the j-th label denote as w_{ij}. Besides, the sparsity of column and row vectors in the coefficient matrix W is controlled by λ_1 and λ_2, respectively.

C. MULTI-LABEL LEARNING WITH MISSING LABELS

At present, the great majority of multi-label learning methods are predicated on the assumption that all class labels of each training instance are known. In the real world, however, it is difficult to collect complete label information for each train instance due to the large personnel and material resources required to annotate instance labels. As a consequence, learning effective label correlations from incomplete label sets and augmenting incomplete label matrix has proven an effective technique to address this problem. This motivates us to include a learning mechanism for label correlations in Eq. (1) to replenish these missing labels. First, let $C^{l\times l}$ represents the label correlation matrix, and C_{ij} represents the degree of correlation between labels i and j. According to LSML [8], which is based on the presumption that missing labels can be reconstructed by exploiting correlations with other labels, the objective function can be represented as

$$\min_{W,C} \frac{1}{2}\|XW - YC\|_F^2 + \frac{\alpha}{2}\|YC-Y\|_F^2 + \beta\|C\|_1 + \lambda_1\|W\|_1 + \lambda_2\|W\|_{2,1}. \quad (2)$$

Note that one class label may be correlated with only a subset of class labels, thus, we add the l_1-norm regularizer over C to learn sparse label dependencies.

D. INCORPORATING LABEL CORRELATION

Considering that improving classification performance can be achieved by fully utilizing the correlations between labels to steer the learning process. The objective function need to incorporate label correlation constraints as a result of this. In order to improve the consistency of the feature-label space, a two-stage second-order label correlation learning technique based on cosine similarity is developed to further confine the label output. Due to the fact that in some cases the assumption "if labels i and j are strongly correlated, the corresponding coefficient vectors w^i and w^j will be similar" is not true [11]. In light of this, we construct the correlation constraint under the premise that "if two labels are strongly correlated, the corresponding outputs of these two labels are highly similar. " Drawing on the idea of SMLMFC method [10], we first introduce the label correlation estimator by the between-class label distribution prior, then use it to refill the label matrix and calculate the cosine similarity of the labels. The following two formulas are put to work to reconstruct the label to calculate cosine similarity.

$$L(c_1, c_2) = \frac{|\ Y_{c_1} \cap Y_{c_2}\ | + s}{|\ Y_{c_1}\ | + 2s}, \quad c_1, c_2 = 1, 2 \ldots, l, \quad (3)$$

where $L \in \mathbb{R}^{l\times l}$, and Y_{c_1} is the set of labeled instances annotated with c_1, $|\ Y_{c_1}\ |$ is the number of labeled instances annotated with c_1. $|\ Y_{c_1} \cap Y_{c_2}\ |$ represents the number of labeled instances annotated with both c_1 and c_2, and $s > 0$ is a smoothness parameter which can avoid label imbalance to some extent. The meaning of such a construction L can be found in [10]. It is observed that L is asymmetric, so it cannot be used directly as the label correlations, but rather to augment the incomplete label matrix with the help of it.

$$\tilde{\boldsymbol{y}}_{ik} = \begin{cases} y_i L(\cdot, k), & \text{if } _{ik} = 0, \\ 1, & \text{otherwise}, \end{cases} \quad (4)$$

where $i \in \{1, 2, \ldots, n\}$ and $k \in \{1, 2, \ldots, l\}$. To ensure $\tilde{\boldsymbol{y}}_{ik} \in [0, 1]$, $\tilde{\boldsymbol{y}}_{ik}$ is normalized as $\tilde{\boldsymbol{y}}_{ik} / \mathbf{max}_{k\in\{1,2,\ldots,c\}}\tilde{\boldsymbol{y}}_{ik}$ for $\boldsymbol{y}_{ik} = 0$. Intuitively, If $\tilde{\boldsymbol{y}}_{ik}$ is assigned a large value, then label k has strong correlations with these labels that have already been annotated to the i-th instance, meaning that k is most likely a missing label. Let $\tilde{L} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n]^T$, then we compute the cosine similarity of paired labels by $\tilde{L}$ and construct the following constrained regular term.

$$\sum_{i,j}^{l} S_{ij}\|(XW)^i - (XW)^j\| = tr((XW)L_1(XW)^T), \quad (5)$$

where $S_{ij} \in S^{l\times l}$, obtained by calculating the cosine similarity between the i-th and j-th columns of L, reveal the degree of label correlation between labels i and j, $(XW)^i$ denotes the i-th column of XW, and $L_1 = S^* - S$ indicates the $l \times l$ label Laplacian matrix of S. $S_{ii}^* = \sum_j^l S_{ij}$. After adding this regularizer, the final optimization formulation can be written as Eq. (6).

$$\min_{W,C} \frac{1}{2}\|XW-YC\|_F^2 + \frac{\alpha}{2}\|YC-Y\|_F^2 + \beta\|C\|_1 + \lambda_1\|W\|_1 + \lambda_2\|W\|_{2,1} + \lambda_3 tr((XW)L_1(XW)^T). \quad (6)$$

Eq. (6), similar to the LSML method, learns higher-order label correlations from incomplete training data and uses them to augment incomplete label matrices, guiding the construction of a multi-label classification model by iteratively learning the label-specific data representation for each class label using the l_1-norm. However, it also considers

the importance of learning label-common features using the $l_{2,1}$-norm, which is not available in the LSML approach. In addition, Eq. (6) includes a second-order label correlation constraint term to ensure that the features selected for highly correlated labels are also consistent, and the computation of these label correlations is based on the SMLMFC approach, which is Eqs. (3) and (4).

E. OPTIMIZATION

Because of the presence of $l_{2,1}$-norm and l_1-norm regularization terms, the optimization problem of our suggested objective function Eq. (6) is convex but non-smooth. To solve this problem, we can first make the $\|W\|_{2,1}$ released by $tr(W^TAW)$ through [37], where A represents the $d \times d$ diagonal matrix, and $A_{ii} = \frac{1}{2\|w_i\|_2}$ indicates the i-th diagonal value in A. Then, we employ the accelerated proximal gradient descent (APG) method to solve the non-smooth optimization problem that includes the l_1-norm regularization term. For simplicity, let $\boldsymbol{\theta} = (W, C)$, and then the optimization model Eq. (6) can be re-expressed as

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + g(\boldsymbol{\theta}), \tag{7}$$

where

$$f(\boldsymbol{\theta}) = \frac{1}{2}\|XW - YC\|_F^2 + \frac{\alpha}{2}\|YC - Y\|_F^2 + \lambda_2 tr(W^TAW) + \lambda_3 tr((XW)L_1(XW)^T), \tag{8}$$

and

$$g(\boldsymbol{\theta}) = \lambda_1\|W\|_1 + \beta\|C\|_1. \tag{9}$$

Here $g(\boldsymbol{\theta})$ is convex but not smooth, while $f(\boldsymbol{\theta})$ is convex and smooth, and $f(\boldsymbol{\theta})$ has a Lipschitz continuous gradient, i.e., $\|\nabla f(\boldsymbol{\theta}_1) - \nabla f(\boldsymbol{\theta}_2)\|_F^2 \leq L_f\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|$, in which L_f is the lipschitz constant, $\boldsymbol{\theta}_1 = (W_1, C_1)$ and $\boldsymbol{\theta}_2 = (W_2, C_2)$. Proximal gradient algorithms minimize a sequence of separable quadratic approximations to $F(\boldsymbol{\theta})$, denoted as

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = f(\boldsymbol{\theta}^{(t)}) + \left\langle \nabla f(\boldsymbol{\theta}^{(t)}), \boldsymbol{\theta} - \boldsymbol{\theta}^{(t)} \right\rangle + \frac{L_f}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\|_F^2 + g(\boldsymbol{\theta}), \tag{10}$$

where $\boldsymbol{\theta}^{(t)} = (W^{(t)}, C^{(t)})$. Let $\boldsymbol{G}^{(t)} = \boldsymbol{\theta}^{(t)} - \frac{1}{L_f}\nabla f(\boldsymbol{\theta}^{(t)})$, then the optimization of $\boldsymbol{\theta}$ can be optimized by

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \arg\min_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \\ &= \arg\min_{\boldsymbol{\theta}} g(\boldsymbol{\theta}) + \frac{L_f}{2}\|\boldsymbol{\theta} - \boldsymbol{G}^{(t)}\|_F^2.\end{aligned} \tag{11}$$

According to the work in [40], the convergence rate can be improved to $O(t^{-2})$ on setting $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}_t + \frac{b_{t-1}-1}{b_t}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$ for a sequence b_t satisfying $b_{t+1}^2 - b_{t+1} \leq b_t^2$, where $\boldsymbol{\theta}_t$ is the result of $\boldsymbol{\theta}$ at the t-th iteration. There are two parameters in each iteration, and we update one variable while fixing the other.

1) UPDATING W

We can begin by calculating $\nabla_W f(\boldsymbol{\theta})$ from Eq. (8), as shown below.

$$\nabla_W f(\boldsymbol{\theta}) = X^TXW - X^TYC + \lambda_2 AW + \lambda_3 X^TXWL_1. \tag{12}$$

With C fixed, Eq. (11) can optimize the coefficient matrix W as follows.

$$W_{t+1} = \arg\min_W \frac{1}{2}\|W - \boldsymbol{G}_W^{(t)}\|_F^2 + \frac{\lambda_1}{L_f}\|W\|_1, \tag{13}$$

where $\boldsymbol{G}_W^{(t)} = W^{(t)} - \frac{1}{L_f}\nabla_W f(W, C)$. Noting that Eq. (13) contains a l_1-norm regularization term, W_{t+1} is generated by soft-thresholding the entries of $\boldsymbol{G}_W^{(t)}$ as

$$W_{t+1} = S_{\frac{\lambda_1}{L_f}}[\boldsymbol{G}_W^{(t)}] = \arg\min_W \frac{\lambda_1}{L_f}\|W\|_1 + \frac{1}{2}\|W - \boldsymbol{G}_W^{(t)}\|_F^2, \tag{14}$$

where $S_\epsilon[\cdot]$ represents the soft-thresholding operator. For any $\omega \in \mathbb{R}$, $S_\epsilon[\omega]$ is defined as

$$S_\epsilon[\omega] = \begin{cases} \epsilon - \mu, & \text{if } \epsilon > \mu, \\ \epsilon + \mu, & \text{if } \epsilon < \mu, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

2) UPDATING C

According to Eq. (8), we can calculate $\nabla_C f(\boldsymbol{\theta})$ as follows.

$$\nabla_C f(\boldsymbol{\theta}) = (1+\alpha)Y^TYC - Y^TXW - \alpha Y^TY. \tag{16}$$

To update C with fixed W, the coefficient matrix C can be optimized by Eq. (11).

$$C = \arg\min_C \frac{1}{2}\|C - \boldsymbol{G}_C^{(t)}\|_F^2 + \frac{\beta}{L_f}\|C\|_1, \tag{17}$$

where $\boldsymbol{G}_C^{(t)} = C^{(t)} - \frac{1}{L_f}\nabla_C f(W, C)$. Then, similarly to Eq. (14), C can be updated with the following equation.

$$C_{t+1} = S_{\frac{\beta}{L_f}}[\boldsymbol{G}_C^{(t)}] = \arg\min_C \frac{\beta}{L_f}\|C\|_1 + \frac{1}{2}\|C - \boldsymbol{G}_C^{(t)}\|_F^2, \tag{18}$$

3) LIPSCHITZ CONSTANT L_F

We estimate the Lipschitz constant L_f in this subsection. With $\boldsymbol{\theta}_1 = (W_1, C_1)$ and $\boldsymbol{\theta}_2 = (W_2, C_2)$, we get

$$\begin{aligned}&\|\nabla f(\boldsymbol{\theta}_1) - \nabla f(\boldsymbol{\theta}_2)\|_F^2 \\ &= \|X^TX\Delta W + \lambda_2 A\Delta W + \lambda_3 X^T X\Delta WL_1\|_F^2 \\ &\quad + \|(1+\alpha)Y^TY\Delta C - Y^TX\Delta W\|_F^2 \\ &\leq 3\|X^TX\Delta W\|_F^2 + 3\|\lambda_2 A\Delta W\|_F^2 + 3\|\lambda_3 X^TX\Delta WL_1\|_F^2 \\ &\quad + 2\|(1+\alpha)Y^TY\Delta C\|_F^2 + 2\|Y^TX\Delta W\|_F^2 \\ &\leq 3(\|X^TX\|_2^2 + \|\lambda_2 A\|_2^2 + \|\lambda_3 X^TX\|_2^2\|L_1\|_2^2)\|\Delta W\|_F^2 \\ &\quad + 2\|(1+\alpha)Y^TY\|_2^2\|\Delta C\|_F^2 + 2\|Y^TX\|_2^2\|\Delta W\|_F^2,\end{aligned} \tag{19}$$

TABLE 2. Statistics of Datasets.

Data set	Domin	#Instances	#Attributes	#Labels	Cardinality
Arts	text(web)	5000	462	26	1.6360
Computers	text	5000	682	33	1.4990
Entertainment	text	5000	641	21	1.4264
Rcv1subset5_top944	text	6000	944	101	2.642
Rcv1subset4_top944	text	6000	944	101	2.484
Rcv1subset3_top944	text	6000	944	101	2.614
Rcv1subset2_top944	text	6000	944	101	2.9342
Rcv1subset1_top944	text	6000	944	101	2.8797
Recreation	text(web)	5000	606	22	1.4232
Science	text(web)	2000	743	40	1.4506
Social	text	5000	1047	39	1.2710
Languagelog	text	1459	1004	75	1.18
Stackex_chess	text	1675	585	227	2.41
Stackex_cs	text	9270	635	274	2.5562

TABLE 3. Friedman Statistics F_F and the Critical Value of Each Evaluation Metric.

Metric	$F_F(K=11,N=14)$	Critical Value $\alpha=0.05$
Hamming loss	14.0032	
Ranking loss	37.7386	
One error	11.9804	1.9042
Average precision	47.2688	
Coverage	39.2910	
MicroF1	18.7490	
AUC	37.3557	

where $\Delta W = W_1 - W_2$, $\Delta C = C_1 - C_2$. Then, the Lipschitz constant is formulated as

$$L_f = \Big\{3\big(\sigma^2_{max}(\lambda_2 A) + (1+(\lambda_3)^2\sigma^2_{max}(L_1))\sigma^2_{max}(X^TX)\big) + 2\big(\sigma^2_{max}((1+\alpha)Y^TY)) + \sigma^2_{max}(Y^TX)\big)\Big\}^{\frac{1}{2}} \quad (20)$$

where $\sigma^2_{max}(\cdot)$ denotes maximum singular value operator of the matrix. To ensure L_f as a constant value, we fix the value of A, which is determined by the initial value of W. Algorithm 1 summarizes the overall optimization steps of the proposed approach CLSML.

Algorithm 1 CLSML: Learning Common and Label-Specific Features for Multi-Label Classification With Missing Labels

Require:
Training data matrix $X \in \mathbb{R}^{n\times d}$, label matrix $Y \in \mathbb{R}^{n\times l}$, and trade-off parameters $\alpha, \beta, \lambda_1, \lambda_2, \lambda_3$.

Ensure: Coefficient matrix $W \in \mathbb{R}^{d\times l}$ and the label correlation matrix $C \in \mathbb{R}^{l\times l}$.

1: initialize $W_0, W_1 \leftarrow (X^TX+\gamma I)^{-1}X^TY$;
2: $C_0, C_1 \leftarrow \text{zeros}(l,l), b_0, b_1 \leftarrow 1, k \leftarrow 1$;
3: Calculate the cosine correlation matrix S corresponding to labels;
4: Calculate the Lipschitz constant L_f;
5: repeat:
Calculate the diagonal matrix A;
$W^{(k)} \leftarrow W_k + \frac{b_{k-1}-1}{b_k}(W_k - W_{k-1})$;
$C^{(k)} \leftarrow C_k + \frac{b_{k-1}-1}{b_k}(C_k - C_{k-1})$;
$\boldsymbol{G}_W^{(k)} \leftarrow W^{(k)} - \frac{1}{L_f}(\nabla f_W(W^{(k)}, C^{(k)}))$;
$W_{k+1} \leftarrow S_{\frac{\lambda_1}{L_f}}(\boldsymbol{G}_W^{(k)})$;
$\boldsymbol{G}_C^{(k)} \leftarrow C^{(k)} - \frac{1}{L_f}(\nabla f_C(W^{(k)}, C^{(k)}))$;
$C_{k+1} \leftarrow S_{\frac{\beta}{L_f}}(\boldsymbol{G}_C^{(k)})$;
$k \leftarrow k+1$;
$b_{k-1} \leftarrow b_k, b_{k+1} \leftarrow \frac{1+\sqrt{4b_k^2+1}}{2}$;
untill: converge;
6: $W \leftarrow W_k, C \leftarrow C_k$.

IV. EXPERIMENTS

A. DATASETS

On 14 multi-label benchmark datasets,[1] we compare our proposed method to 10 outstanding multi-label methods to illustrate its effectiveness. Table 2 summarizes the specific information for each dataset, including the number of instances, features, labels, and cardinality (the average number of labels per instance).

B. EVALUATION METRICS

To validate the effectiveness of our proposed method, we use seven popular evaluation metrics, [42], [43], [44] which are often divided into example-based metrics and ranking-based metrics. Hamming Loss and Micro F1 are a couple of example-based metrics, while Average Precision, One-Error, Ranking Loss, and Coverage are indications of ranking-based metrics. For each evaluation metric, "↑" indicates that the higher the value, the better the performance, and "↓" suggests that the lower the value, the better the performance.

Given a multi-label test set $\boldsymbol{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^m$. Let $\Omega = \{\boldsymbol{x}_i : i = 1, \ldots, m\}$ be the feature set, and Θ be the label set containing l labels. Where $\boldsymbol{y}_i \in \{0,1\}^l$ is the ground-truth

[1]These datasets can be download from
http://mulan.sourceforge.net/datasets-mlc.html
http://cometa.ujaen.es
http://computer.njnu.edu.cn/Lab/LABIC/LABIC_Software.html

TABLE 4. Experimental Results of All Comparing Learning Algorithm (mean) on first Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.1.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Rcv1subset5	HM(↓)	0.0264	0.0261	0.0259	0.0258	**0.0257**	0.0261	0.0329	0.0259	0.0259	0.0263	0.0265
	Mi(↑)	0.4650	0.5122	0.5007	0.5113	0.5093	0.5095	0.3532	**0.5129**	0.5005	0.5097	0.5063
	AP(↑)	0.5811	0.6393	0.6260	0.6402	0.6365	0.6381	0.4809	**0.6430**	0.6364	0.6401	0.6388
	OE(↓)	0.4168	0.4022	0.4062	0.4018	0.4077	0.4120	0.5155	0.4007	0.3967	0.3980	**0.3949**
	RL(↓)	0.0644	0.0504	0.0617	0.0459	0.0517	0.0435	0.0669	0.0474	0.0480	0.0467	**0.0433**
	CV(↓)	0.1485	0.1256	0.1484	0.1157	0.1285	0.1126	0.1456	0.1201	0.1215	0.1192	**0.1106**
	AUC(↑)	0.9124	0.9288	0.9151	0.9346	0.9271	0.9361	0.9139	0.9322	0.9319	0.9329	**0.9378**
Rcv1subset4	HM(↓)	0.0217	0.0202	0.0204	0.0202	0.0200	0.0206	0.0218	**0.0199**	0.0200	0.0201	0.0204
	Mi(↑)	0.4896	0.5507	0.5426	0.5435	0.5514	0.5429	0.3522	**0.5510**	0.5434	0.5488	0.5366
	AP(↑)	0.6498	0.7069	0.6997	0.7078	0.7094	0.7007	0.5450	0.7116	0.7083	0.7055	**0.7142**
	OE(↓)	0.3538	0.3355	0.3368	0.3313	0.3248	0.3480	0.5028	0.3265	**0.3238**	0.3418	0.3393
	RL(↓)	0.0508	0.0371	0.0450	0.0338	0.0380	0.0324	0.0518	0.0352	0.0353	0.0347	**0.0323**
	CV(↓)	0.1169	0.0914	0.1064	0.0847	0.0938	0.0813	0.1097	0.0880	0.0880	0.0859	**0.0808**
	AUC(↑)	0.9236	0.9418	0.9324	0.9462	0.9406	0.9481	0.9263	0.9439	0.9444	0.9456	**0.9481**
Rcv1subset3	HM(↓)	0.0273	0.0262	0.0267	0.0263	0.0261	0.0262	0.0418	0.0263	**0.0261**	0.0267	0.0268
	Mi(↑)	0.4641	0.5065	0.4900	0.5072	0.5032	0.5083	0.3538	**0.5128**	0.5048	0.5119	0.5097
	AP(↑)	0.5773	0.6302	0.6159	0.6313	0.6245	0.6313	0.4651	0.6332	0.6274	0.6304	**0.6375**
	OE(↓)	0.4470	**0.4128**	0.4198	0.4228	0.4233	0.4220	0.5832	0.4197	0.4190	0.4282	0.4270
	RL(↓)	0.0648	0.0526	0.0627	0.0485	0.0582	0.0464	0.0714	0.0481	0.0490	0.0486	**0.0447**
	CV(↓)	0.1469	0.1271	0.1471	0.1183	0.1381	0.1145	0.1509	0.1187	0.1191	0.1187	**0.1108**
	AUC(↑)	0.9125	0.9263	0.9146	0.9314	0.9195	0.9337	0.9084	0.9308	0.9312	0.9307	**0.9354**
Rcv1subset2	HM(↓)	0.0275	0.0263	0.0262	0.0264	0.0263	0.0275	0.0433	**0.0256**	0.0265	0.0264	0.0270
	Mi(↑)	0.4666	0.5041	0.4889	0.5063	0.4978	0.5113	0.3602	0.5082	0.5018	**0.5117**	0.5070
	AP(↑)	0.5814	0.6334	0.6209	0.6396	0.6310	0.6372	0.4640	0.6395	0.6332	0.6377	**0.6415**
	OE(↓)	0.4335	0.4167	0.4257	0.4105	0.4122	0.4162	0.5705	0.4112	**0.4078**	0.4135	0.4132
	RL(↓)	0.0633	0.0529	0.0635	0.0479	0.0577	0.0449	0.0693	0.0479	0.0486	0.0475	**0.0443**
	CV(↓)	0.1437	0.1297	0.1503	0.1186	0.1393	0.1107	0.1484	0.1189	0.1213	0.1180	**0.1103**
	AUC(↑)	0.9135	0.9243	0.9117	0.9306	0.9184	0.9357	0.9091	0.9302	0.9297	0.9310	**0.9353**
Rcv1subset1	HM(↓)	0.0307	0.0300	0.0298	0.0296	0.0301	0.0300	0.0410	**0.0290**	0.0291	0.0294	0.0302
	Mi(↑)	0.4962	0.5129	0.4912	0.5227	0.5185	0.5260	0.3976	0.5225	0.5141	0.5246	**0.5302**
	AP(↑)	0.5740	0.6090	0.5932	0.6140	0.6113	0.6164	0.4587	0.6223	0.6102	0.6194	**0.6257**
	OE(↓)	0.4473	0.4287	0.4357	0.4283	0.4207	0.4277	0.5963	**0.4137**	0.4242	0.4145	0.4165
	RL(↓)	0.0593	0.0533	0.0663	0.0488	0.0546	0.0448	0.0717	0.0480	0.0497	0.0477	**0.0443**
	CV(↓)	0.1415	0.1345	0.1598	0.1245	0.1372	0.1138	0.1626	0.1220	0.1261	0.1223	**0.1136**
	AUC(↑)	0.9220	0.9290	0.9155	0.9346	0.9281	0.9395	0.9116	0.9356	0.9339	0.9356	**0.9402**
Stackex_cs	HM(↓)	0.0110	0.0110	0.0107	0.0109	0.0110	0.0109	0.0145	0.0110	**0.0105**	0.0114	0.0106
	Mi(↑)	0.2951	0.4574	0.4556	0.4584	0.4571	0.4576	0.3583	0.4598	0.4548	0.4566	**0.4609**
	AP(↑)	0.3135	0.5255	0.5201	0.5265	0.5233	0.5258	0.3731	0.5275	0.5216	0.5270	**0.5282**
	OE(↓)	0.6211	0.4398	0.4429	0.4401	0.4405	0.4398	0.6793	0.4386	0.4462	0.4396	**0.4386**
	RL(↓)	0.1450	0.0726	0.0775	0.0695	0.0748	0.0697	0.0750	0.0697	0.0680	0.0686	**0.0659**
	CV(↓)	0.2456	0.1573	0.1670	0.1515	0.1616	0.1513	0.1503	0.1515	0.1455	0.1493	**0.1406**
	AUC(↑)	0.8583	0.9206	0.9160	0.9239	0.9190	0.9238	0.9201	0.9239	0.9265	0.9249	**0.9277**
Stackex_chess	HM(↓)	**0.0117**	0.0118	0.0135	0.0126	0.0120	0.0131	0.0168	0.0122	0.0123	0.0165	0.0137
	Mi(↑)	0.3229	0.4230	0.3863	0.4225	0.4241	0.4159	0.2988	**0.4244**	0.4182	0.3622	0.4089
	AP(↑)	0.3833	0.4962	0.4632	0.5082	0.4958	0.5032	0.3682	**0.5104**	0.4966	0.4596	0.5002
	OE(↓)	0.5236	0.4364	0.4693	0.4125	0.4388	0.4269	0.6501	**0.4107**	0.4322	0.5515	0.4370
	RL(↓)	0.1190	0.1149	0.1454	0.0997	0.1167	0.0952	0.1153	0.1035	0.1152	0.1096	**0.0935**
	CV(↓)	0.2198	0.2325	0.2861	0.1997	0.2345	0.1936	0.2264	0.2109	0.2268	0.2008	**0.1907**
	AUC(↑)	0.8749	0.8759	0.8474	0.8925	0.8754	0.8977	0.8755	0.8874	0.8785	0.8809	**0.8983**

label set of the test instance $\boldsymbol{x}_i$, and $h(\boldsymbol{x}_i)$ is the set of predicted labels for the i-th instance $\boldsymbol{x}_i$.

1) HAMMING LOSS

(HM) evaluates the frequency of an instance-label pair being misclassified, that is, a label not belonging to the instance is predicted or a label belonging to the instance is not predicted, denoted as

$$\mathrm{HM} = \frac{1}{m}\sum_{i=1}^{m}\frac{\left\|h(\boldsymbol{x}_i) \oplus \boldsymbol{y}_i\right\|_1}{l}, \tag{21}$$

where $\oplus$ indicates the symmetric difference between two sets, and $\|\cdot\|_1$ is the l_1-norm. The smaller the value, the better the performance.

2) MICRO F1

(Mi) calculates the F1 score on the predictions of different labels as a whole, it evaluates both micro average of precision and micro average of recall with equal importance.

$$\mathrm{Mi} = \frac{2\sum_{i=1}^{m}\sum_{j=1}^{l}(h(\boldsymbol{x}_i)_j)y_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{l}h(\boldsymbol{x}_i)_j + \sum_{i=1}^{m}\sum_{j=1}^{l}y_{ij}}, \tag{22}$$

where $h(\boldsymbol{x}_i)_j$ is the j-th element of $h(\boldsymbol{x}_i)$. The larger the value, the better the performance.

These ranking-based metrics rely on the real value output function $f : \Omega \times \Theta \to \mathbb{R}$ of each algorithm. For $(\boldsymbol{x}_i, y) \in \Omega \times \Theta$, $f(\boldsymbol{x}_i, y)$ means the confidence score that $\boldsymbol{x}_i$ marked as label y. $f(\cdot,\cdot)$ can be transformed into a ranking function

TABLE 5. Experimental Results of All Comparing Learning Algorithm (mean) on last Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.1.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Science	HM(↓)	0.0380	0.0387	0.0415	0.0380	0.0378	0.0385	0.0425	0.0391	0.0386	**0.0375**	0.0401
	Mi(↑)	0.4552	0.4593	0.4364	0.4641	0.4597	0.4603	0.3146	**0.4643**	0.4553	0.4627	0.4535
	AP(↑)	0.6025	0.6087	0.5883	0.6104	0.6044	0.6129	0.5053	0.6082	0.6055	0.6083	**0.6112**
	OE(↓)	0.4902	0.4782	0.4984	0.4816	0.4816	0.4838	0.6274	**0.4724**	0.4920	0.4854	0.4898
	RL(↓)	0.1211	0.1219	0.1397	0.1054	0.1262	0.1068	0.1098	0.1153	0.1127	0.1146	**0.0993**
	CV(↓)	0.1666	0.1701	0.1916	0.1492	0.1754	0.1511	0.1435	0.1629	0.1598	0.1618	**0.1401**
	AUC(↑)	0.8476	0.8460	0.8279	0.8638	0.8410	0.8627	0.8664	0.8525	0.8563	0.8526	**0.8716**
Recreation	HM(↓)	0.0653	0.0703	0.0708	0.0654	0.0675	0.0677	0.0754	0.0667	0.0681	0.0657	**0.6248**
	Mi(↑)	0.4734	0.4640	0.4508	0.4740	0.4646	0.4731	0.3612	0.4709	0.4619	**0.4749**	0.4744
	AP(↑)	0.6427	0.6369	0.6261	0.6460	0.6366	0.6486	0.5653	0.6405	0.6421	0.6439	**0.6503**
	OE(↓)	0.4476	0.4518	0.4626	0.4458	0.4506	0.4448	0.5702	0.4498	0.4534	0.4446	**0.4440**
	RL(↓)	0.1386	0.1447	0.1563	0.1362	0.1458	0.1292	0.1407	0.1405	0.1351	0.1389	**0.1281**
	CV(↓)	0.1884	0.1964	0.2086	0.1847	0.1970	0.1770	0.1799	0.1904	0.1844	0.1884	**0.1755**
	AUC(↑)	0.8200	0.8127	0.8016	0.8230	0.8117	0.8291	0.8204	0.8176	0.8221	0.8192	**0.8305**
Languagelog	HM(↓)	0.0187	**0.0175**	0.0238	0.0186	0.0176	0.0188	0.0544	0.0184	0.0181	0.0200	0.0219
	Mi(↑)	0.2728	0.2734	0.2765	0.2856	0.2744	0.2885	0.1400	0.2881	0.2835	0.2813	**0.2889**
	AP(↑)	0.3328	**0.3603**	0.3371	0.3569	0.3573	0.3572	0.2031	0.3581	0.3542	0.3553	0.3508
	OE(↓)	0.7498	**0.7108**	0.7272	0.7210	0.7114	0.7252	0.8807	0.7183	0.7231	0.7327	0.7444
	RL(↓)	0.1448	0.1433	0.1770	0.1296	0.1448	0.1291	0.2516	0.1320	0.1434	0.1275	**0.1153**
	CV(↓)	0.1763	0.1793	0.2194	0.1633	0.1817	0.1629	0.2965	0.1649	0.1776	0.1607	**0.1488**
	AUC(↑)	0.7047	0.7015	0.6683	0.7141	0.7009	0.7152	0.5963	0.7140	0.7043	0.7165	**0.7274**
Arts	HM(↓)	0.0639	0.0661	0.0681	0.0667	0.0666	0.0639	0.0739	**0.0638**	0.0644	0.0673	0.0641
	Mi(↑)	**0.4598**	0.4532	0.4485	0.4589	0.4545	0.4594	0.3327	0.4590	0.4551	0.4591	0.4583
	AP(↑)	0.6297	0.6256	0.6179	0.6298	0.6263	0.6345	0.5422	0.6301	0.6319	0.6307	**0.6351**
	OE(↓)	**0.4506**	0.4520	0.4646	0.4534	0.4508	0.4542	0.5898	0.4522	0.4520	0.4526	0.4560
	RL(↓)	0.1314	0.1360	0.1412	0.1297	0.1379	0.1185	0.1330	0.1280	0.1233	0.1289	**0.1163**
	CV(↓)	0.2039	0.2086	0.2173	0.2003	0.2121	0.1862	0.1898	0.1985	0.1931	0.1997	**0.1826**
	AUC(↑)	0.8262	0.8213	0.8156	0.8274	0.8187	0.8402	0.8358	0.8296	0.8359	0.8284	**0.8436**
Social	HM(↓)	0.0218	0.0219	0.0224	0.0218	0.0213	0.0216	0.0324	**0.0212**	0.0217	0.0214	0.0217
	Mi(↑)	0.6420	0.6579	0.6484	0.6624	0.6627	0.6540	0.5259	**0.6627**	0.6603	0.6612	0.6581
	AP(↑)	0.7755	0.7958	0.7800	0.7963	0.7984	0.7978	0.6861	0.7985	0.7952	0.7991	**0.8011**
	OE(↓)	0.2766	0.2512	0.2666	0.2580	0.2508	0.2588	0.4462	**0.2498**	0.2534	0.2532	0.2628
	RL(↓)	0.0687	0.0612	0.0745	0.0547	0.0609	0.0510	0.0539	0.0564	0.0608	0.0553	**0.0507**
	CV(↓)	0.0988	0.0931	0.1089	0.0829	0.0914	0.0780	0.0732	0.0852	0.0905	0.0848	**0.0704**
	AUC(↑)	0.9007	0.9059	0.8909	0.9156	0.9068	0.9199	0.9213	0.9131	0.9067	0.9136	**0.9259**
Entertainment	HM(↓)	0.0549	0.0568	0.0597	0.0557	0.0582	0.0547	0.0794	0.0579	0.0573	0.0542	**0.0539**
	Mi(↑)	**0.5846**	0.5738	0.5627	0.5804	0.5705	0.5782	0.4747	0.5740	0.5726	0.5808	0.5807
	AP(↑)	0.7227	0.7203	0.7110	0.7304	0.7203	0.7268	0.6383	0.7236	0.7230	0.7250	**0.7275**
	OE(↓)	0.3520	0.3560	0.3602	0.3424	0.3522	0.3532	0.5182	0.3536	**0.3510**	0.3512	0.3532
	RL(↓)	0.1009	0.1058	0.1149	0.0914	0.1064	0.0910	0.1003	0.0990	0.0977	0.0977	**0.0876**
	CV(↓)	0.1442	0.1513	0.1612	0.1327	0.1514	0.1317	0.1349	0.1421	0.1408	0.1411	**0.1267**
	AUC(↑)	0.8625	0.8562	0.8469	0.8727	0.8552	0.8737	0.8688	0.8640	0.8657	0.8655	**0.8780**
Computers	HM(↓)	**0.0371**	0.0378	0.0390	0.0378	0.0377	0.0383	0.0444	0.0372	0.0380	0.0377	0.0384
	Mi(↑)	0.5625	0.5680	0.5624	0.5628	0.5675	0.5534	0.4923	**0.5695**	0.5611	0.5688	0.5521
	AP(↑)	0.7163	0.7201	0.7101	0.7245	0.7181	0.7205	0.6429	0.7225	0.7235	0.7215	**0.7255**
	OE(↓)	0.3352	0.3326	0.3416	0.3374	0.3372	0.3460	0.4670	**0.3282**	0.3366	0.3360	0.3506
	RL(↓)	0.0864	0.0884	0.0987	0.0737	0.0878	0.0691	0.0735	0.0832	0.0758	0.0815	**0.0685**
	CV(↓)	0.1285	0.1322	0.1458	0.1142	0.1323	0.1060	0.1085	0.1266	0.1185	0.1245	**0.1060**
	AUC(↑)	0.8832	0.8791	0.8680	0.8940	0.8800	0.9016	0.8988	0.8844	0.8921	0.8871	**0.9027**

$rank_f(\cdot,\cdot) \in \{1, 2, \ldots, l\}$, satisfying that if $f(\boldsymbol{x}_i, y) > f(\boldsymbol{x}_i, y')$, then $rank_f(\boldsymbol{x}_i, y) < rank_f(\boldsymbol{x}_i, y')$.

3) AVERAGE PRECISION

(AP) evaluates the average fraction of relevant labels ranked higher than a particular relevant label

$$\text{AP} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|\boldsymbol{y}_i|}\sum_{y\in\boldsymbol{y}_i}\frac{\left|\{y' \in \boldsymbol{y}_i : rank_f(\boldsymbol{x}_i, y') \le rank_f(\boldsymbol{x}_i, y)\}\right|}{rank_f(\boldsymbol{x}_i, y)}, \tag{23}$$

where $|\cdot|$ is the cardinality of a set. The larger the value, the better the performance.

4) ONE ERROR

(OE) evaluates how many times the top-ranked label is not in the set of ground-truth labels of the instance.

$$\text{OE} = \frac{1}{m}\sum_{i=1}^{m}\delta([\arg\max_{y\in\boldsymbol{y}} f(\boldsymbol{x}_i, y)] \notin \boldsymbol{y}_i), \tag{24}$$

where $\delta(z)$ is an indicator function that returns 1 if z is true and 0 otherwise. The smaller the value, the better the performance.

5) RANKING LOSS

(RL) evaluates the fraction of reversely ordered label pairs, i.e. an irrelevant label is ranked higher than a relevant label,

TABLE 6. Experimental Results of All Comparing Learning Algorithm (mean) on first Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.2.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Rcv1subset5	HM(↓)	**0.0256**	0.0262	0.0265	0.0263	0.0260	0.0269	0.0304	0.0260	0.0268	0.0265	0.0264
	Mi(↑)	0.4575	**0.4914**	0.4862	0.4910	0.4879	0.4890	0.3657	0.4909	0.4847	0.4910	0.4842
	AP(↑)	0.5763	0.6262	0.6126	0.6291	0.6223	0.6270	0.4710	**0.6324**	0.6250	0.6274	0.6272
	OE(↓)	0.4140	0.4067	0.4115	0.4088	0.4133	0.4150	0.5507	0.4037	0.4025	0.4078	**0.4017**
	RL(↓)	0.0705	0.0626	0.0766	0.0556	0.0631	0.0517	0.0745	0.0568	0.0567	0.0565	**0.0508**
	CV(↓)	0.1594	0.1496	0.1744	0.1348	0.1500	0.1261	0.1587	0.1383	0.1367	0.1381	**0.1243**
	AUC(↑)	0.9052	0.9144	0.8991	0.9232	0.9142	0.9279	0.9054	0.9208	0.9219	0.9218	**0.9289**
Rcv1subset4	HM(↓)	0.0215	0.0209	0.0213	0.0210	0.0209	0.0214	0.0258	0.0208	**0.0207**	**0.0207**	0.0213
	Mi(↑)	0.4666	0.5311	0.5239	0.5312	0.5311	0.5262	0.3507	**0.5403**	0.5282	0.5335	0.5245
	AP(↑)	0.6399	0.6985	0.6904	0.6992	0.6992	0.6950	0.5353	0.6934	0.6996	0.6988	**0.7075**
	OE(↓)	0.3642	0.3427	0.3472	0.3492	0.3407	0.3648	0.5497	0.3448	**0.3403**	0.3533	0.3610
	RL(↓)	0.0547	0.0424	0.0508	0.0383	0.0435	0.0359	0.0537	0.0394	0.0390	0.0393	**0.0355**
	CV(↓)	0.1235	0.1017	0.1178	0.0934	0.1046	0.0884	0.1130	0.0969	0.0954	0.0954	**0.0876**
	AUC(↑)	0.9184	0.9346	0.9246	0.9401	0.9328	0.9430	0.9233	0.9378	0.9389	0.9389	**0.9434**
Rcv1subset3	HM(↓)	0.0266	0.0272	0.0269	0.0268	0.0267	0.0271	0.0378	0.0265	0.0273	0.0272	**0.0263**
	Mi(↑)	0.4488	0.4938	0.4748	0.4910	0.4870	0.4965	0.3426	0.4927	0.4904	**0.4967**	0.4912
	AP(↑)	0.5712	0.6217	0.6061	0.6274	0.6161	0.6274	0.4549	0.6270	0.6232	0.6252	**0.6299**
	OE(↓)	0.4545	0.4295	0.4335	0.4313	0.4278	0.4350	0.6395	0.4270	**0.4257**	0.4352	0.4360
	RL(↓)	0.0674	0.0587	0.0710	0.0518	0.0651	0.0500	0.0747	0.0534	0.0536	0.0534	**0.0487**
	CV(↓)	0.1504	0.1383	0.1633	0.1247	0.1506	0.1211	0.1574	0.1285	0.1284	0.1281	**0.1185**
	AUC(↑)	0.9099	0.9190	0.9051	0.9271	0.9112	0.9287	0.9041	0.9242	0.9254	0.9242	**0.9301**
Rcv1subset2	HM(↓)	0.0273	0.0266	0.0271	0.0268	0.0269	0.0266	0.0447	0.0263	**0.0261**	0.0264	0.0276
	Mi(↑)	0.4480	0.4902	0.4763	0.4950	0.4823	0.4950	0.3578	0.4974	0.4842	0.4956	**0.4977**
	AP(↑)	0.5729	0.6275	0.6144	0.6308	0.6215	0.6320	0.4564	0.6335	0.6251	0.6323	**0.6337**
	OE(↓)	0.4438	**0.4157**	0.4230	0.4263	0.4222	0.4275	0.6185	0.4217	0.4240	0.4233	0.4275
	RL(↓)	0.0662	0.0583	0.0700	0.0519	0.0643	0.0472	0.0722	0.0521	0.0528	0.0515	**0.0481**
	CV(↓)	0.1479	0.1405	0.1633	0.1264	0.1513	0.1156	0.1541	0.1265	0.1297	0.1254	**0.1174**
	AUC(↑)	0.9097	0.9173	0.9039	0.9254	0.9103	0.9315	0.9056	0.9250	0.9240	0.9258	**0.9306**
Rcv1subset1	HM(↓)	0.0309	0.0308	0.0314	0.0305	0.0303	0.0305	0.0392	0.0306	**0.0302**	0.0308	0.0305
	Mi(↑)	0.4877	0.4957	0.4785	0.5073	0.4998	0.5103	0.3980	0.5084	0.5002	0.5078	**0.5112**
	AP(↑)	0.5661	0.5996	0.5817	0.6058	0.6005	0.6115	0.4547	0.6118	0.6022	0.6087	**0.6153**
	OE(↓)	0.4663	0.4407	0.4482	0.4398	0.4342	0.4383	0.6227	0.4377	0.4400	0.4345	**0.4328**
	RL(↓)	0.0614	0.0593	0.0732	0.0543	0.0603	0.0467	0.0744	0.0531	0.0539	0.0529	**0.0484**
	CV(↓)	0.1448	0.1460	0.1751	0.1355	0.1484	0.1186	0.1680	0.1318	0.1344	0.1327	**0.1219**
	AUC(↑)	0.9196	0.9220	0.9068	0.9280	0.9214	0.9370	0.9079	0.9295	0.9288	0.9293	**0.9352**
Stackex_cs	HM(↓)	0.0107	0.0107	0.0112	0.0108	0.0111	0.0112	0.0149	0.0111	**0.0105**	0.0111	0.0108
	Mi(↑)	0.2817	0.4555	0.4493	0.4563	0.4542	0.4536	0.3557	0.4545	0.4518	0.4544	**0.4578**
	AP(↑)	0.3015	0.5202	0.5148	0.5225	0.5181	0.5218	0.3722	0.5191	0.5181	0.5205	**0.5245**
	OE(↓)	0.6286	0.4461	0.4483	0.4429	0.4464	0.4417	0.6833	0.4422	0.4454	0.4417	**0.4391**
	RL(↓)	0.1533	0.0768	0.0813	0.0719	0.0775	0.0731	0.0762	0.0735	0.0708	0.0719	**0.0691**
	CV(↓)	0.2549	0.1666	0.1755	0.1570	0.1675	0.1584	0.1530	0.1599	0.1504	0.1568	**0.1503**
	AUC(↑)	0.8512	0.9161	0.9119	0.9208	0.9157	0.9202	0.9189	0.9196	0.9240	0.9211	**0.9242**
Stackex_chess	HM(↓)	0.0124	**0.0116**	0.0134	0.0119	0.0119	0.0126	0.0174	0.0119	0.0119	0.0159	0.0131
	Mi(↑)	0.3023	0.4097	0.3738	0.4159	0.4167	0.4127	0.2933	**0.4232**	0.4156	0.3586	0.4099
	AP(↑)	0.3656	0.4926	0.4447	0.4935	0.4872	0.4939	0.3627	0.4947	0.4861	0.4535	**0.4957**
	OE(↓)	0.5510	0.4299	0.4913	0.4269	0.4418	0.4328	0.6472	0.4322	0.4364	0.5560	**0.4302**
	RL(↓)	0.1251	0.1194	0.1537	0.1084	0.1267	0.1047	0.1234	0.1148	0.1236	0.1159	**0.1015**
	CV(↓)	0.2302	0.2397	0.2963	0.2143	0.2525	0.2063	0.2419	0.2295	0.2434	0.2112	**0.2055**
	AUC(↑)	0.8688	0.8718	0.8399	0.8838	0.8661	0.8907	0.8678	0.8764	0.8687	0.8745	**0.8912**

denoted as

$$\mathrm{RL} = \frac{1}{m}\sum_{i=1}^{m}\frac{\left|(y_a, y_b) \in \boldsymbol{y}_i \times \hat{\boldsymbol{y}}_i : f(\boldsymbol{x}_i, y_a) \leq f(\boldsymbol{x}_i, y_b)\right|}{\left|\boldsymbol{y}_i\right|\left|\hat{\boldsymbol{y}}_i\right|}, \tag{25}$$

where (y_a, y_b) is the pair class label for an instance $\boldsymbol{x}_i$, $\hat{\boldsymbol{y}}_i$ is the complementary set of $\boldsymbol{y}_i$ and $\boldsymbol{y}_i$ is the known label set of the i-th instance. The smaller the value, the better the performance.

6) COVERAGE

evaluates how many steps are needed, on average, to go down the label ranking list to cover all the ground-truth labels of the instance.

$$\mathrm{Coverage} = \frac{1}{m}\sum_{i=1}^{m}\max_{y\in\boldsymbol{y}_i} rank_f(\boldsymbol{x}_i, y) - 1. \tag{26}$$

The smaller the value, the better the performance.

7) MACRO AUC

(AUC) evaluates the average of AUC of all the class labels.

$$\mathrm{AUC} = \frac{1}{l}\sum_{j=1}^{l}\frac{|\{(\boldsymbol{x}', \boldsymbol{x}'') \in \boldsymbol{Z}_j \times \overline{\boldsymbol{Z}}_j : f(\boldsymbol{x}', l_j) \geq f(\boldsymbol{x}'', l_j)\}|}{|\boldsymbol{Z}_j||\overline{\boldsymbol{Z}}_j|}, \tag{27}$$

TABLE 7. Experimental Results of All Comparing Learning Algorithm (mean) on last Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.2.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Science	HM(↓)	0.0381	0.0385	0.0411	0.0383	0.0386	0.0387	0.0437	**0.0375**	0.0376	**0.0375**	0.0385
	Mi(↑)	0.4504	0.4554	0.4323	0.4564	0.4528	0.4554	0.3154	**0.4585**	0.4494	0.4553	0.4533
	AP(↑)	0.5993	0.6032	0.5851	0.6016	0.6016	0.6134	0.5029	0.6028	0.6018	0.6020	**0.6034**
	OE(↓)	0.4954	0.4838	0.5032	0.4890	0.4870	0.4850	0.6360	**0.4812**	0.4932	0.4884	0.5034
	RL(↓)	0.1235	0.1274	0.1414	0.1083	0.1287	0.1094	0.1110	0.1194	0.1153	0.1205	**0.1006**
	CV(↓)	0.1702	0.1783	0.1957	0.1547	0.1804	0.1557	0.1461	0.1695	0.1640	0.1703	**0.1431**
	AUC(↑)	0.8429	0.8389	0.8240	0.8585	0.8361	0.8576	0.8631	0.8460	0.8513	0.8442	**0.8681**
Recreation	HM(↓)	**0.0651**	0.0675	0.0711	0.0654	0.0690	0.0663	0.0762	0.0666	0.0681	0.0657	0.0656
	Mi(↑)	**0.4682**	0.4584	0.4441	0.4644	0.4568	0.4656	0.3608	0.4619	0.4517	0.4662	0.4627
	AP(↑)	0.6407	0.6363	0.6248	0.6382	0.6314	0.6449	0.5664	0.6375	0.6384	0.6412	**0.6471**
	OE(↓)	0.4506	0.4530	0.4648	0.4534	0.4598	0.4496	0.5696	0.4554	0.4582	0.4501	**0.4470**
	RL(↓)	0.1403	0.1473	0.1572	0.1415	0.1491	0.1323	0.1408	0.1428	0.1369	0.1402	**0.1308**
	CV(↓)	0.1906	0.1984	0.2094	0.1911	0.2012	0.1807	0.1814	0.1929	0.1864	0.1905	**0.1791**
	AUC(↑)	0.8159	0.8077	0.7990	0.8137	0.8066	0.8235	0.8169	0.8131	0.8180	0.8153	**0.8246**
Languagelog	HM(↓)	0.0188	**0.0171**	0.0227	0.0186	0.0174	0.0184	0.0549	0.0178	0.0174	0.0182	0.0207
	Mi(↑)	0.2637	0.2559	0.2566	0.2685	0.2458	0.2776	0.1389	0.2710	0.2513	0.2722	**0.2798**
	AP(↑)	0.3271	0.3552	0.3269	0.3526	0.3531	0.3572	0.1998	0.3510	0.3476	0.3559	**0.3581**
	OE(↓)	0.7505	**0.7121**	0.7471	0.7327	0.7217	0.7252	0.8849	0.7245	0.7272	0.7252	0.7491
	RL(↓)	0.1517	0.1462	0.1920	0.1320	0.1486	0.1352	0.2589	0.1358	0.1516	0.1322	**0.1186**
	CV(↓)	0.1855	0.1831	0.2348	0.1661	0.1858	0.1734	0.3054	0.1686	0.1891	0.1689	**0.1523**
	AUC(↑)	0.6970	0.6968	0.6545	0.7112	0.6963	0.7066	0.5877	0.7085	0.6937	0.7089	**0.7230**
Arts	HM(↓)	0.0656	0.0672	0.0683	0.0648	0.0680	0.0656	0.0744	0.0661	0.0645	0.0669	**0.0644**
	Mi(↑)	**0.4558**	0.4474	0.4391	0.4534	0.4470	0.4527	0.3168	0.4526	0.4504	0.4497	0.4531
	AP(↑)	0.6287	0.6241	0.6133	0.6270	0.6220	0.6284	0.5390	0.6267	0.6288	0.6258	**0.6303**
	OE(↓)	0.4570	**0.4542**	0.4666	0.4558	0.4598	0.4578	0.6004	0.4606	0.4584	0.4606	0.4630
	RL(↓)	0.1318	0.1369	0.1471	0.1321	0.1415	0.1201	0.1321	0.1293	0.1236	0.1317	**0.1183**
	CV(↓)	0.2042	0.2105	0.2236	0.2047	0.2170	0.1889	0.1889	0.2015	0.1943	0.2038	**0.1861**
	AUC(↑)	0.8248	0.8195	0.8079	0.8235	0.8140	0.8370	0.8350	0.8270	0.8346	0.8247	**0.8401**
Social	HM(↓)	0.0218	0.0218	0.0229	0.0224	0.0217	0.0221	0.0317	0.0219	0.0217	**0.0214**	0.0218
	Mi(↑)	0.6346	0.6513	0.6351	0.6421	0.6480	0.6401	0.5215	0.6477	0.6506	0.6515	**0.6536**
	AP(↑)	0.7671	0.7905	0.7766	0.7894	0.7910	0.7903	0.6917	0.7922	0.7913	0.7937	**0.7942**
	OE(↓)	0.2860	**0.2570**	0.2756	0.2684	0.2594	0.2704	0.4310	0.2600	0.2572	0.2574	0.2716
	RL(↓)	0.0729	0.0657	0.0791	0.0594	0.0647	0.0549	0.0559	0.0611	0.0651	0.0608	**0.0548**
	CV(↓)	0.1052	0.1002	0.1154	0.0902	0.0971	0.0824	0.0788	0.0923	0.0988	0.0930	**0.0751**
	AUC(↑)	0.8934	0.8976	0.8847	0.9077	0.9002	0.9143	0.9162	0.9056	0.8990	0.9045	**0.9198**
Entertainment	HM(↓)	**0.0545**	0.0576	0.0607	0.0552	0.0581	0.0561	0.0814	0.0568	0.0576	0.0573	0.0568
	Mi(↑)	**0.5799**	0.5669	0.5527	0.5726	0.5656	0.5688	0.4617	0.5685	0.5628	0.5695	0.5683
	AP(↑)	0.7195	0.7155	0.7063	0.7248	0.7171	0.7229	0.6370	0.7185	0.7179	0.7194	**0.7242**
	OE(↓)	0.3609	0.3622	0.3724	0.3592	0.3592	0.3606	0.5204	0.3608	0.3608	0.3634	**0.3586**
	RL(↓)	0.1016	0.1104	0.1184	0.0941	0.1106	0.0931	0.1007	0.1051	0.1005	0.1016	**0.0896**
	CV(↓)	0.1457	0.1585	0.1674	0.1365	0.1578	0.1357	0.1350	0.1512	0.1455	0.1457	**0.1300**
	AUC(↑)	0.8607	0.8496	0.8412	0.8683	0.8491	0.8694	0.8663	0.8558	0.8607	0.8599	**0.8739**
Computers	HM(↓)	**0.0372**	0.0379	0.0396	0.0378	0.0380	0.0384	0.0441	0.0374	0.0375	0.0376	0.0384
	Mi(↑)	0.5544	0.5582	0.5499	0.5565	0.5608	0.5422	0.4926	**0.5655**	0.5548	0.5608	0.5432
	AP(↑)	0.7123	0.7170	0.7099	0.7223	0.7151	0.7188	0.6428	0.7177	0.7182	0.7173	**0.7196**
	OE(↓)	0.3392	0.3334	0.3426	0.3368	0.3396	0.3488	0.4646	**0.3300**	0.3380	0.3382	0.3508
	RL(↓)	0.0903	0.0945	0.1024	0.0762	0.0920	0.0696	0.0736	0.0861	0.0782	0.0844	**0.0705**
	CV(↓)	0.1352	0.1432	0.1514	0.1187	0.1397	0.1102	0.1102	0.1317	0.1222	0.1289	**0.1089**
	AUC(↑)	0.8777	0.8705	0.8619	0.8899	0.8729	0.8981	0.8967	0.8800	0.8871	0.8821	**0.8988**

where $\boldsymbol{Z}_j = \{\boldsymbol{x}_i | l_j \in \boldsymbol{y}_i, 1 \leq i \leq l\}$, $\overline{\boldsymbol{Z}}_j = \{\boldsymbol{x}_i | l_j \notin \boldsymbol{y}_i, 1 \leq i \leq l\}$ indicates the set of test instances with(without) label l_j. The larger the value, the better the performance.

C. COMPARISON METHODS

To confirm the effectiveness of the CLSML approach, we compare it to the ten top performance-related algorithms listed below.

GLOCAL [28]: In order to deal with both the full-label and the missing-label cases, GLOCAL employs both global and local correlations in the development of a manifold regularization term for learning a latent label representation. The value of model parameter $\lambda = 1$, λ_1 to λ_5 are tuned in $\{2^{-5}, 2^{-4}, \ldots, 2^{1}\}$.

JLCLS [38]: JLCLS provides a multi-label learning algorithm for joint label completion and label-specific features to address the issue of less consideration in the impact of label marking errors or defaults in datasets. By combining joint label completion with label-specific features, it develops a new multi-label learning algorithm framework. The alternating iteration method is used to obtain the completion matrix and label-specific features. Model parameters α, β and θ are tuned in $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$.

LLSF [18]: During the binary training stage, inconsistency occurs when an instance belongs to more than one class label and can be treated as both a positive and negative instance simultaneously. LLSF attempts to overcome the inconsistency problem by learning label-specific features for

TABLE 8. Experimental Results of All Comparing Learning Algorithm (mean) on first Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.3.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Rcv1subset5	HM(↓)	0.0262	0.0268	0.0270	0.0263	0.0266	0.0263	0.0338	0.0262	**0.0257**	0.0264	0.0263
	Mi(↑)	0.4455	0.4738	0.4663	0.4758	0.4735	0.4725	0.3627	**0.4780**	0.4651	0.4758	0.4711
	AP(↑)	0.5640	0.6147	0.5981	0.6153	0.6110	0.6148	0.4662	**0.6163**	0.6121	0.6159	0.6150
	OE(↓)	0.4212	0.4077	0.4217	0.4127	0.4158	0.4187	0.5640	0.4163	0.4080	0.4143	**0.4077**
	RL(↓)	0.0778	0.0768	0.0918	0.0673	0.0774	0.0613	0.0846	0.0704	0.0674	0.0666	**0.0617**
	CV(↓)	0.1715	0.1764	0.2034	0.1571	0.1769	0.1441	0.1773	0.1633	0.1567	0.1560	**0.1446**
	AUC(↑)	0.8962	0.8995	0.8824	0.9097	0.8984	0.9174	0.8940	0.9066	0.9098	0.9103	**0.9168**
Rcv1subset4	HM(↓)	0.0214	0.0219	0.0216	0.0216	0.0223	0.0220	0.0281	0.0219	**0.0211**	0.0218	0.0221
	Mi(↑)	0.4463	0.4980	0.4943	0.4915	0.4990	0.4901	0.3424	**0.5053**	0.4972	0.4972	0.4901
	AP(↑)	0.6165	0.6793	0.6698	0.6793	0.6777	0.6751	0.5148	0.6841	0.6823	0.6827	**0.6867**
	OE(↓)	0.3908	0.3645	**0.3630**	0.3738	0.3632	0.3885	0.6048	0.3658	0.3592	0.3672	0.3868
	RL(↓)	0.0643	0.0541	0.0672	0.0487	0.0581	0.0450	0.0620	0.0493	0.0479	0.0485	**0.0431**
	CV(↓)	0.1392	0.1241	0.1491	0.1141	0.1311	0.1051	0.1274	0.1157	0.1124	0.1127	**0.1016**
	AUC(↑)	0.9060	0.9188	0.9040	0.9261	0.9147	0.9308	0.9129	0.9246	0.9272	0.9261	**0.9332**
Rcv1subset3	HM(↓)	**0.0268**	0.0279	0.0286	0.0271	0.0281	0.0272	0.0424	0.0274	0.0274	0.0278	0.0275
	Mi(↑)	0.4333	0.4634	0.4494	0.4637	0.4592	0.4654	0.3483	**0.4663**	0.4596	0.4657	0.4612
	AP(↑)	0.5627	0.6067	0.5909	0.6118	0.5985	0.6144	0.4418	0.6138	0.6078	0.6127	**0.6168**
	OE(↓)	0.4583	**0.4450**	0.4457	0.4517	0.4453	0.4527	0.6793	0.4470	0.4502	0.4518	0.4585
	RL(↓)	0.0738	0.0723	0.0851	0.0627	0.0796	0.0592	0.0833	0.0645	0.0635	0.0625	**0.0577**
	CV(↓)	0.1599	0.1632	0.1882	0.1449	0.1777	0.1378	0.1717	0.1481	0.1460	0.1442	**0.1340**
	AUC(↑)	0.9017	0.9033	0.8883	0.9132	0.8936	0.9173	0.8940	0.9109	0.9126	0.9127	**0.9190**
Rcv1subset2	HM(↓)	**0.0266**	0.0275	0.0285	0.0274	0.0280	0.0273	0.0433	0.0276	0.0276	0.0276	0.0276
	Mi(↑)	0.4259	0.4602	0.4441	0.4638	0.4508	0.4677	0.3435	**0.4705**	0.4592	0.4690	0.4682
	AP(↑)	0.5606	0.6077	0.5933	0.6154	0.6009	0.6146	0.4442	0.6155	0.6091	0.6145	**0.6175**
	OE(↓)	0.4503	0.4418	**0.4383**	0.4397	0.4387	0.4495	0.6667	0.4392	0.4410	0.4418	0.4493
	RL(↓)	0.0733	0.0715	0.0857	0.0629	0.0799	0.0566	0.0811	0.0634	0.0615	0.0626	**0.0572**
	CV(↓)	0.1578	0.1636	0.1904	0.1464	0.1791	0.1348	0.1680	0.1466	0.1439	0.1451	**0.1341**
	AUC(↑)	0.9017	0.9021	0.8857	0.9122	0.8922	0.9185	0.8954	0.9118	0.9139	0.9128	**0.9191**
Rcv1subset1	HM(↓)	**0.0306**	0.0311	0.0318	0.0311	0.0313	0.0308	0.0422	0.0308	0.0309	0.0311	**0.0306**
	Mi(↑)	0.4695	0.4696	0.4494	0.4806	0.4732	0.4806	0.3810	0.4807	0.4675	0.4796	**0.4814**
	AP(↑)	0.5594	0.5801	0.5601	0.5868	0.5814	0.5950	0.4369	0.5939	0.5846	0.5907	**0.5953**
	OE(↓)	0.4695	0.4613	0.4758	0.4667	0.4520	0.4687	0.6565	**0.4487**	0.4657	0.4592	0.4620
	RL(↓)	0.0684	0.0752	0.0896	0.0661	0.0754	0.0554	0.0837	0.0649	0.0652	0.0651	**0.0592**
	CV(↓)	0.1570	0.1773	0.2051	0.1575	0.1777	0.1358	0.1848	0.1544	0.1544	0.1545	**0.1434**
	AUC(↑)	0.9116	0.9046	0.8903	0.9150	0.9053	0.9270	0.8975	0.9165	0.9168	0.9163	**0.9230**
Stackex_cs	HM(↓)	0.0109	0.0112	0.0110	0.0111	0.0110	0.0109	0.0147	0.0109	**0.0107**	0.0108	0.0109
	Mi(↑)	0.2569	0.4492	0.4456	0.4522	0.4499	0.4536	0.3553	0.4515	0.4470	0.4533	**0.4553**
	AP(↑)	0.2846	0.5161	0.5090	0.5179	0.5143	0.5178	0.3730	0.5157	0.5143	0.5153	**0.5179**
	OE(↓)	0.6333	0.4456	0.4522	0.4439	0.4436	0.4410	0.6806	0.4416	0.4483	0.4400	**0.4399**
	RL(↓)	0.1591	0.0809	0.0856	0.0768	0.0834	0.0781	0.0783	0.0777	0.0751	0.0777	**0.0723**
	CV(↓)	0.2663	0.1757	0.1852	0.1665	0.1798	0.1692	0.1581	0.1695	0.1591	0.1689	**0.1572**
	AUC(↑)	0.8445	0.9116	0.9072	0.9165	0.9098	0.9147	0.9166	0.9149	0.9195	0.9151	**0.9207**
Stackex_chess	HM(↓)	0.0135	0.0117	0.0133	0.0117	0.0117	0.0123	0.0164	0.0115	**0.0114**	0.0161	0.0135
	Mi(↑)	0.2912	0.4041	0.3560	0.4110	0.3962	0.4092	0.2913	**0.4116**	0.4096	0.3560	0.4069
	AP(↑)	0.3476	0.4770	0.4256	0.4841	0.4671	0.4911	0.3603	0.4840	0.4758	0.4480	**0.4921**
	OE(↓)	0.5660	0.4418	0.5104	0.4442	0.4639	0.4360	0.6376	0.4478	0.4519	0.5600	**0.4352**
	RL(↓)	0.1343	0.1337	0.1641	0.1185	0.1361	0.1104	0.1271	0.1239	0.1349	0.1242	**0.1039**
	CV(↓)	0.2445	0.2638	0.3165	0.2292	0.2700	0.2237	0.2503	0.2439	0.2662	0.2256	**0.2114**
	AUC(↑)	0.8580	0.8584	0.8276	0.8755	0.8564	0.8816	0.8627	0.8681	0.8586	0.8658	**0.8874**

each label. It then assumes that each label is only associated with a fraction of the original feature set and that any two strongly correlated class labels can share more information than two uncorrelated or weakly correlated ones. The value of parameters α and β are tuned in $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$, and the value of threshold τ is set to 0.5.

CIMML [34]: It proposes Class Imbalance aware Missing labels Multi-label Learning, which handles class imbalance issue by constructing a label weight matrix with weight estimation guided by how frequently a label is present, absent, and unobserved. Model parameters λ_C,λ_R,λ_W,λ_L,λ_I and β are searched in $\{10^{-5}, 10^{-4}, \ldots, 10^{3}\}$.

SGMML [36]: To deal with the issues emerging from incomplete labels and high-dimensional input space, SGMML propose a multi-label learning approach based on identifying the label-specific features and constraining them with a sparse global structure. The sparse structural constraint helps maintain the typical characteristics of the multi-label learning data. Besides, it also constructs supplementary label correlations to assist missing label recovery as part of the optimization problem,and learns label-specific features and constrain them with sparse global structure for multi-label learning with missing labels. The value of parameters λ_s are searched in $\{10^{-5}, 10^{-4}, \ldots, 10^{3}\}$.

LRMML [35]: It proposed a unified framework to capture the label correlations, which utilize both auxiliary label matrix and the low rank constraints on estimated labels. Besides, it also enforces maximal separation among different

TABLE 9. Experimental Results of All Comparing Learning Algorithm (mean) on last Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.3.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Science	HM(↓)	**0.0372**	0.0389	0.0412	0.0394	0.0388	0.0384	0.0453	0.0380	0.0376	0.0388	0.0400
	Mi(↑)	0.4377	0.4407	0.4189	0.4398	0.4389	0.4404	0.3193	0.4420	0.4340	**0.4459**	0.4290
	AP(↑)	0.5943	0.5963	0.5740	0.6023	0.5944	0.6050	0.5017	0.5968	0.5959	0.5978	**0.6037**
	OE(↓)	0.4970	0.4876	0.5146	0.4944	0.4892	0.4900	0.6444	**0.4862**	0.5016	0.4940	0.5062
	RL(↓)	0.1234	0.1343	0.1531	0.1136	0.1389	0.1145	0.1128	0.1273	0.1208	0.1260	**0.1055**
	CV(↓)	0.1720	0.1904	0.2123	0.1631	0.1973	0.1657	0.1595	0.1812	0.1725	0.1794	**0.1507**
	AUC(↑)	0.8412	0.8276	0.8092	0.8498	0.8219	0.8490	0.8582	0.8356	0.8432	0.8360	**0.8601**
Recreation	HM(↓)	0.0669	0.0666	0.0692	0.0660	0.0666	0.0665	0.0733	0.0679	0.0668	0.0678	**0.0645**
	Mi(↑)	**0.4544**	0.4493	0.4369	0.4493	0.4498	0.4526	0.3678	0.4490	0.4410	0.4513	0.4523
	AP(↑)	0.6352	0.6279	0.6182	0.6327	0.6255	0.6364	0.5654	0.6301	0.6315	0.6335	**0.6383**
	OE(↓)	0.4578	0.4628	0.4754	0.4594	0.4642	0.4584	0.5706	0.4646	0.4656	0.4588	**0.4572**
	RL(↓)	0.1411	0.1556	0.1621	0.1475	0.1543	0.1374	0.1443	0.1494	0.1410	0.1464	**0.1360**
	CV(↓)	0.1913	0.2094	0.2179	0.1990	0.2086	0.1863	0.1882	0.2025	0.1912	0.1980	**0.1852**
	AUC(↑)	0.8120	0.7956	0.7889	0.8044	0.7976	0.8148	0.8080	0.8031	0.8094	0.8053	**0.8156**
Languagelog	HM(↓)	0.0185	**0.0164**	0.0235	0.0179	0.0169	0.0178	0.0569	0.0180	0.0166	0.0176	0.0216
	Mi(↑)	0.2449	0.2363	0.2402	0.2515	0.2278	0.2539	0.1299	0.2459	0.2332	0.2418	**0.2528**
	AP(↑)	0.3191	0.3444	0.3099	0.3443	0.3445	0.3487	0.1953	0.3439	0.3415	**0.3485**	0.3353
	OE(↓)	0.7533	**0.7313**	0.7601	0.7388	0.7316	0.7334	0.8890	0.7443	0.7334	0.7327	0.7622
	RL(↓)	0.1527	0.1506	0.1990	0.1401	0.1553	0.1352	0.2690	0.1414	0.1580	0.1350	**0.1228**
	CV(↓)	0.1875	0.1888	0.2469	0.1750	0.1969	0.1716	0.3168	0.1775	0.1979	0.1711	**0.1575**
	AUC(↑)	0.6933	0.6911	0.6400	0.7016	0.6855	0.7067	0.5783	0.7001	0.6869	0.7061	**0.7174**
Arts	HM(↓)	0.0647	0.0658	0.0669	0.0666	0.0662	0.0670	0.0747	0.0663	0.0659	**0.0643**	0.0644
	Mi(↑)	**0.4545**	0.4362	0.4271	0.4424	0.4336	0.4419	0.2962	0.4413	0.4394	0.4395	0.4402
	AP(↑)	0.6255	0.6184	0.6084	0.6220	0.6158	0.6275	0.5314	0.6211	0.6265	0.6209	**0.6281**
	OE(↓)	**0.4552**	0.4656	0.4712	0.4612	0.4682	0.4672	0.6206	0.4660	0.4588	0.4664	0.4706
	RL(↓)	0.1413	0.1426	0.1542	0.1374	0.1465	0.1248	0.1331	0.1356	0.1291	0.1367	**0.1208**
	CV(↓)	0.2173	0.2184	0.2328	0.2116	0.2236	0.1959	0.1895	0.2098	0.2020	0.2113	**0.1889**
	AUC(↑)	0.8134	0.8111	0.7998	0.8153	0.8060	0.8302	0.8317	0.8180	0.8269	0.8172	**0.8357**
Social	HM(↓)	**0.0211**	0.0223	0.0234	0.0222	0.0220	0.0222	0.0279	0.0222	0.0219	0.0218	0.0225
	Mi(↑)	0.6317	0.6377	0.6209	0.6350	0.6366	0.6348	0.5234	0.6367	0.6392	0.6376	**0.6403**
	AP(↑)	0.7533	0.7833	0.7666	0.7803	0.7815	0.7818	0.6964	0.7844	0.7810	0.7835	**0.7869**
	OE(↓)	0.2974	**0.2624**	0.2838	0.2756	0.2686	0.2764	0.4166	0.2652	0.2700	0.2700	0.2790
	RL(↓)	0.0786	0.0710	0.0836	0.0672	0.0713	0.0612	0.0615	0.0677	0.0720	0.0652	**0.0605**
	CV(↓)	0.1150	0.1078	0.1239	0.1018	0.1071	0.0936	0.0898	0.1032	0.1077	0.0998	**0.0823**
	AUC(↑)	0.8805	0.8900	0.8752	0.8951	0.8884	0.9010	0.9044	0.8943	0.8879	0.8963	**0.9083**
Entertainment	HM(↓)	0.0579	0.0579	0.0599	0.0548	0.0571	0.0605	0.0743	0.0572	0.0588	0.0575	**0.0567**
	Mi(↑)	**0.5659**	0.5512	0.5410	0.5556	0.5539	0.5451	0.4262	0.5519	0.5485	0.5489	0.5480
	AP(↑)	0.7134	0.7063	0.6966	0.7123	0.7098	0.7128	0.6339	0.7112	0.7090	0.7130	**0.7134**
	OE(↓)	**0.3622**	0.3718	0.3794	0.3654	0.3700	0.3720	0.5178	0.3714	0.3738	0.3674	0.3732
	RL(↓)	0.1055	0.1188	0.1279	0.0993	0.1145	0.0991	0.1037	0.1097	0.1062	0.1083	**0.0943**
	CV(↓)	0.1516	0.1697	0.1793	0.1441	0.1634	0.1432	0.1382	0.1574	0.1527	0.1550	**0.1368**
	AUC(↑)	0.8545	0.8370	0.8272	0.8596	0.8429	0.8596	0.8597	0.8477	0.8520	0.8500	**0.8661**
Computers	HM(↓)	**0.0376**	0.0397	0.0403	0.0384	0.0388	0.0391	0.0446	0.0379	0.0385	0.0386	0.0392
	Mi(↑)	0.5416	0.5423	0.5326	0.5374	0.5442	0.5312	0.4892	**0.5477**	0.5385	0.5401	0.5316
	AP(↑)	0.7094	0.7086	0.6984	0.7145	0.7124	0.7148	0.6408	0.7123	0.7126	0.7108	**0.7148**
	OE(↓)	0.3400	0.3404	0.3528	0.3462	**0.3370**	0.3486	0.4630	0.3376	0.3414	0.3406	0.3538
	RL(↓)	0.0929	0.0977	0.1106	0.0792	0.0966	0.0728	0.0737	0.0899	0.0812	0.0890	**0.0720**
	CV(↓)	0.1405	0.1484	0.1635	0.1250	0.1493	0.1172	0.1123	0.1397	0.1308	0.1381	**0.1103**
	AUC(↑)	0.8731	0.8650	0.8519	0.8842	0.8648	0.8920	0.8943	0.8721	0.8811	0.8743	**0.8954**

label subspaces for better label differentiation and recovers missing labels and trains the classifier by building an auxiliary label matrix and imposing low and high rank constraints on label subspace. δ is set as 0.005 and the hyperparameters λ_R,λ_L and λ_T are searched in $\{10^{-10}, 10^{-9}, \ldots, 10^{5}\}$.

SMILE [25]: Semi-supervised multi-label classification using incomplete label information for the whole name, estimating label correlations from partially labeled instances, and replenishing missing labels. Then it constructs a neighborhood graph using labeled and unlabeled instances. Finally, the known labels and renewed labels of labeled instances are combined to train a graph-based semi-supervised linear classifier. Based on the recommendations in the publication, the parameters α, s and the number of nearest neighbors k are set to 0.35, 0.5 and 5, respectively.

CLML [11]: It suggests an innovative method for learning common and label-specific features for multi-label classification using correlation information obtained from labels and instances. First, it introduces the regularizers $l_{2,1}$-norm and l_1-norm to simultaneously learn common and label-specific features. Second, it constrains label correlations on label outputs rather than the coefficient matrix by employing a regularizer. Finally, the k-nearest neighbor approach is adopted to take instance correlations into account. Values of all parameters α, γ, λ_1 and λ_2 are tuned in $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$, respectively.

TABLE 10. Experimental Results of All Comparing Learning Algorithm (mean) on first Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.4.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Rcv1subset5	HM(↓)	0.0267	0.0274	0.0270	0.0271	0.0270	0.0268	0.0328	0.0267	**0.0262**	0.0272	0.0270
	Mi(↑)	0.4376	0.4702	0.4542	0.4741	0.4678	0.4755	0.3593	0.4730	0.4618	**0.4762**	0.4715
	AP(↑)	0.5584	0.6062	0.5839	0.6141	0.6007	0.6163	0.4694	0.6131	0.6101	**0.6127**	0.6126
	OE(↓)	0.4228	0.4152	0.4312	0.4217	0.4233	0.4185	0.5442	0.4163	0.4129	0.4152	**0.4113**
	RL(↓)	0.0790	0.0792	0.0987	0.0673	0.0817	0.0603	0.0834	0.0691	0.0683	0.0677	**0.0616**
	CV(↓)	0.1750	0.1823	0.2200	0.1576	0.1871	0.1466	0.1792	0.1629	0.1607	0.1600	**0.1464**
	AUC(↑)	0.8947	0.8957	0.8739	0.9096	0.8933	0.9157	0.8948	0.9065	0.9083	0.9089	**0.9162**
Rcv1subset4	HM(↓)	0.0215	0.0220	0.0221	0.0217	0.0216	0.0217	0.0283	**0.0212**	0.0219	0.0217	0.0221
	Mi(↑)	0.4313	0.4825	0.4758	0.4832	0.4811	0.4754	0.3357	0.4877	**0.4889**	0.4801	0.4742
	AP(↑)	0.6058	0.6649	0.6565	0.6717	0.6663	0.6644	0.5066	0.6741	0.6729	0.6673	**0.6756**
	OE(↓)	0.3985	0.3720	0.3695	0.3798	**0.3630**	0.4008	0.6302	0.3732	0.3695	0.3847	0.3950
	RL(↓)	0.0671	0.0622	0.0749	0.0545	0.0650	0.0488	0.0650	0.0556	0.0523	0.0545	**0.0480**
	CV(↓)	0.1437	0.1410	0.1670	0.1247	0.1474	0.1130	0.1340	0.1292	0.1215	0.1257	**0.1118**
	AUC(↑)	0.9028	0.9090	0.8939	0.9192	0.9060	0.9267	0.9094	0.9171	0.9219	0.9184	**0.9274**
Rcv1subset3	HM(↓)	**0.0271**	0.0283	0.0291	0.0284	0.0286	0.0282	0.0418	0.0279	0.0282	0.0287	0.0276
	Mi(↑)	0.4130	0.4464	0.4228	0.4504	0.4419	0.4568	0.3420	**0.4556**	0.4510	0.4499	0.4517
	AP(↑)	0.5484	0.5906	0.5758	0.5981	0.5840	0.6021	0.4362	0.6006	0.5974	0.5982	**0.6044**
	OE(↓)	0.4683	0.4573	0.4602	0.4622	0.4540	0.4602	0.6895	**0.4538**	0.4555	0.4618	0.4657
	RL(↓)	0.0808	0.0818	0.0998	0.0715	0.0924	0.0675	0.0897	0.0742	0.0695	0.0714	**0.0657**
	CV(↓)	0.1725	0.1830	0.2157	0.1613	0.2035	0.1543	0.1841	0.1663	0.1581	0.1632	**0.1489**
	AUC(↑)	0.8940	0.8913	0.8714	0.9029	0.8791	0.9075	0.8863	0.8995	0.9066	0.9020	**0.9098**
Rcv1subset2	HM(↓)	**0.0274**	0.0292	0.0289	0.0289	0.0297	0.0284	0.0426	0.0291	0.0294	0.0291	0.0282
	Mi(↑)	0.4149	0.4463	0.4266	0.4537	0.4352	0.4531	0.3306	**0.4540**	0.4469	0.4527	0.4509
	AP(↑)	0.5477	0.5934	0.5760	0.6012	0.5841	0.6037	0.4374	0.6030	0.5978	0.6032	**0.6067**
	OE(↓)	0.4590	0.4483	0.4532	0.4500	0.4505	0.4562	0.6802	**0.4423**	0.4450	0.4430	0.4580
	RL(↓)	0.0801	0.0826	0.0969	0.0714	0.0915	0.0625	0.0872	0.0714	0.0693	0.0704	**0.0639**
	CV(↓)	0.1693	0.1854	0.2136	0.1631	0.2007	0.1490	0.1803	0.1633	0.1600	0.1613	**0.1481**
	AUC(↑)	0.8941	0.8893	0.8736	0.9021	0.8784	0.9103	0.8878	0.9021	0.9048	0.9033	**0.9105**
Rcv1subset1	HM(↓)	**0.0301**	0.0314	0.0323	0.0313	0.0316	0.0312	0.0415	0.0312	0.0310	0.0311	0.0312
	Mi(↑)	0.4604	0.4641	0.4389	0.4723	0.4615	0.4809	0.3807	0.4795	0.4652	0.4753	**0.4848**
	AP(↑)	0.5472	0.5713	0.5468	0.5803	0.5707	0.5921	0.4345	0.5874	0.5775	0.5848	**0.5922**
	OE(↓)	0.4717	0.4657	0.4798	0.4697	0.4657	0.4625	0.6615	**0.4542**	0.4660	0.4548	0.4560
	RL(↓)	0.0753	0.0811	0.1012	0.0708	0.0824	0.0632	0.0852	0.0707	0.0685	0.0702	**0.0618**
	CV(↓)	0.1696	0.1901	0.2287	0.1694	0.1933	0.1514	0.1884	0.1686	0.1633	0.1690	**0.1499**
	AUC(↑)	0.9050	0.8985	0.8777	0.9096	0.8978	0.9197	0.8958	0.9099	0.9129	0.9098	**0.9204**
Stackex_cs	HM(↓)	0.0112	0.0111	0.0112	0.0113	0.0110	0.0109	0.0148	0.0112	**0.0105**	0.0111	0.0108
	Mi(↑)	0.2214	0.4431	0.4370	0.4450	0.4424	0.4464	0.3536	0.4436	0.4441	0.4466	**0.4471**
	AP(↑)	0.2663	0.5062	0.4988	0.5080	0.5040	0.5093	0.3712	0.5073	0.5080	0.5100	**0.5119**
	OE(↓)	0.6655	0.4574	0.4604	0.4538	0.4558	0.4548	0.6804	0.4584	0.4536	0.4534	**0.4531**
	RL(↓)	0.1638	0.0880	0.0906	0.0831	0.0902	0.0830	0.0790	0.0828	0.0790	0.0816	**0.0774**
	CV(↓)	0.2745	0.1896	0.1954	0.1801	0.1931	0.1800	0.1609	0.1795	0.1678	0.1785	**0.1607**
	AUC(↑)	0.8395	0.9047	0.9016	0.9098	0.9028	0.9093	0.9152	0.9092	0.9156	0.9106	**0.9160**
Stackex_chess	HM(↓)	0.0117	0.0114	0.0132	0.0115	0.0113	0.0121	0.0161	**0.0113**	**0.0113**	0.0165	0.0129
	Mi(↑)	0.2598	0.3860	0.3429	0.4009	0.3850	0.4004	0.2825	0.4048	0.3934	0.3488	**0.4059**
	AP(↑)	0.3193	0.4594	0.4119	0.4728	0.4512	0.4774	0.3508	0.4701	0.4613	0.4381	**0.4840**
	OE(↓)	0.6263	0.4740	0.5260	0.4496	0.4830	0.4460	0.6430	0.4609	0.4657	0.5693	**0.4334**
	RL(↓)	0.1458	0.1375	0.1830	0.1173	0.1455	0.1193	0.1406	0.1327	0.1450	0.1299	**0.1143**
	CV(↓)	0.2634	0.2715	0.3482	0.2379	0.2862	0.2368	0.2714	0.2647	0.2835	0.2346	**0.2292**
	AUC(↑)	0.8459	0.8527	0.8088	0.8725	0.8459	0.8724	0.8501	0.8590	0.8473	0.8601	**0.8781**

LSF-CI [39]: With both correlation information in label space and correlation information in feature space taken into account, it learns label-specific features for each label. The k neighborhood graph model is used in LSF-CI to compute instance correlations in feature space, and cosine similarity is used to compute label correlations in label space. Label correlations are then included under the assumption that the similarity between coefficient vectors W_i and W_j may be high if label y_i is strongly correlated with label y_j, and an instance correlation regular term is also included under the assumption that their corresponding predicted labels generated through the coefficient matrix may be similar if the two instances are strongly correlated in the original feature space. Following that, the classification coefficient matrix W is learned. Values of parameters α, γ and η are seached in $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$ respectively, and the value of threshold τ is set to 0.5. Besides, its additional parameter β is seached in $\{2^{-12}, 2^{-11}, \ldots, 2^{12}\}$, and the number of nearest neighbors k is 10.

LSML [8]: It presents a novel framework for incorporating both label correlations and classification coefficient matrix into learning models. While filling the missing labels, the framework can extract label-specific features. The value for All parameters are turned in $\{10^{-5}, 10^{-4}, \ldots, 10^{3}\}$.

D. COMPARATIVE EXPERIMENTS AND ANALYSIS

In our experiments, we evaluate the performance of each algorithm using five-fold cross-validations, i.e., 80% of each

TABLE 11. Experimental Results of All Comparing Learning Algorithm (mean) on last Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.4.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Science	HM(↓)	0.0392	0.0393	0.0413	0.0385	0.0398	**0.0383**	0.0475	0.0385	0.0395	0.0396	0.0393
	Mi(↑)	0.4161	0.4161	0.4005	0.4190	0.4109	0.4176	0.3329	**0.4205**	0.4132	0.4190	0.4069
	AP(↑)	0.5798	0.5840	0.5674	0.5925	0.5827	0.5936	0.4971	0.5838	0.5822	0.5834	**0.5848**
	OE(↓)	0.5126	0.5028	0.5200	0.5068	0.5024	0.5048	0.6562	0.5000	0.5120	**0.4984**	0.5274
	RL(↓)	0.1320	0.1443	0.1600	0.1217	0.1463	0.1221	0.1185	0.1371	0.1265	0.1324	**0.1123**
	CV(↓)	0.1847	0.2062	0.2247	0.1771	0.2090	0.1787	0.1639	0.1999	0.1821	0.1913	**0.1622**
	AUC(↑)	0.8281	0.8115	0.7975	0.8368	0.8089	0.8360	0.8446	0.8182	0.8323	0.8250	**0.8479**
Recreation	HM(↓)	**0.0652**	0.0682	0.0692	0.0674	0.0671	0.0668	0.0723	0.0658	0.0666	0.0683	**0.0652**
	Mi(↑)	**0.4451**	0.4318	0.4175	0.4361	0.4280	0.4403	0.3680	0.4376	0.4311	0.4352	0.4349
	AP(↑)	0.6279	0.6205	0.6054	0.6222	0.6171	0.6282	0.5657	0.6269	0.6209	0.6272	**0.6279**
	OE(↓)	0.4630	0.4730	0.4934	0.4742	0.4782	0.4720	0.5626	0.4680	0.4790	0.4656	**0.4630**
	RL(↓)	0.1501	0.1614	0.1728	0.1559	0.1607	0.1460	0.1526	0.1558	0.1496	0.1528	**0.1405**
	CV(↓)	0.2023	0.2173	0.2294	0.2095	0.2172	0.1981	0.1969	0.2114	0.2002	0.2071	**0.1903**
	AUC(↑)	0.7981	0.7839	0.7737	0.7899	0.7850	0.8007	0.7910	0.7914	0.7967	0.7935	**0.8039**
Languagelog	HM(↓)	0.0193	**0.0165**	0.0229	0.0170	0.0168	0.0177	0.0581	0.0170	0.0166	0.0172	0.0209
	Mi(↑)	0.2187	0.2214	0.2348	0.2263	0.2126	0.2383	0.1280	0.2272	0.2085	0.2288	**0.2464**
	AP(↑)	0.3077	0.3328	0.3085	0.3416	0.3292	0.3409	0.1916	0.3382	0.3267	**0.3458**	0.3310
	OE(↓)	0.7697	0.7423	0.7615	0.7347	0.7519	0.7382	0.8910	0.7409	0.7573	**0.7361**	0.7629
	RL(↓)	0.1571	0.1615	0.2044	0.1438	0.1640	0.1435	0.2793	0.1493	0.1629	0.1454	**0.1277**
	CV(↓)	0.1949	0.2031	0.2531	0.1844	0.2053	0.1828	0.3278	0.1903	0.2048	0.1833	**0.1653**
	AUC(↑)	0.6855	0.6797	0.6354	0.6947	0.6750	0.6962	0.5666	0.6889	0.6775	0.6924	**0.7086**
Arts	HM(↓)	**0.0638**	0.0677	0.0683	0.0675	0.0679	0.0679	0.0743	0.0661	0.0668	0.0668	0.0670
	Mi(↑)	**0.4375**	0.4216	0.4128	0.4265	0.4169	0.4283	0.2863	0.4287	0.4253	0.4210	0.4208
	AP(↑)	0.6128	0.6075	0.6010	0.6139	0.6074	0.6138	0.5252	0.6143	0.6144	0.6147	**0.6163**
	OE(↓)	**0.4672**	0.4764	0.4806	0.4736	0.4758	0.4758	0.6348	0.4716	0.4764	0.4706	0.4880
	RL(↓)	0.1341	0.1535	0.1620	0.1448	0.1535	0.1306	0.1342	0.1437	0.1358	0.1454	**0.1279**
	CV(↓)	0.2060	0.2331	0.2455	0.2232	0.2335	0.2052	0.1997	0.2234	0.2114	0.2249	**0.1993**
	AUC(↑)	0.8212	0.7964	0.7877	0.8056	0.7962	0.8208	0.8233	0.8056	0.8173	0.8043	**0.8259**
Social	HM(↓)	**0.0213**	0.0226	0.0242	0.0221	0.0223	0.0225	0.0274	0.0223	0.0225	0.0220	0.0220
	Mi(↑)	0.6291	0.6298	0.6033	0.6323	0.6294	0.6307	0.5304	0.6330	0.6294	0.6352	**0.6372**
	AP(↑)	0.7478	0.7752	0.7501	0.7770	0.7758	0.7788	0.6987	0.7764	0.7757	0.7771	**0.7791**
	OE(↓)	0.3034	0.2752	0.3022	0.2782	0.2746	0.2804	0.4104	**0.2724**	0.2726	0.2736	0.2844
	RL(↓)	0.0830	0.0776	0.0986	0.0706	0.0778	0.0646	0.0643	0.0695	0.0773	0.0685	**0.0643**
	CV(↓)	0.1204	0.1162	0.1407	0.1072	0.1157	0.0975	0.0906	0.1056	0.1159	0.1038	**0.0889**
	AUC(↑)	0.8755	0.8814	0.8593	0.8902	0.8816	0.8974	0.9001	0.8918	0.8815	0.8929	**0.9039**
Entertainment	HM(↓)	**0.0597**	0.0600	0.0632	0.0610	0.0624	0.0610	0.0661	0.0615	0.0603	0.0618	0.0628
	Mi(↑)	**0.5263**	0.5109	0.5003	0.5144	0.5096	0.5107	0.4108	0.5102	0.5066	0.5115	0.5065
	AP(↑)	0.7060	0.7008	0.6905	0.7055	0.7017	0.7052	0.6317	0.7013	0.7010	0.7035	**0.7063**
	OE(↓)	**0.3714**	0.3766	0.3888	0.3766	0.3776	0.3752	0.5094	0.3816	0.3830	0.3770	0.3832
	RL(↓)	0.1131	0.1277	0.1354	0.1081	0.1231	0.1059	0.1090	0.1174	0.1130	0.1170	**0.1017**
	CV(↓)	0.1653	0.1827	0.1917	0.1586	0.1778	0.1547	0.1474	0.1705	0.1631	0.1705	**0.1438**
	AUC(↑)	0.8403	0.8211	0.8126	0.8432	0.8247	0.8463	0.8476	0.8319	0.8384	0.8323	**0.8513**
Computers	HM(↓)	**0.0380**	0.0402	0.0410	0.0392	0.0392	0.0387	0.0450	0.0396	0.0391	0.0393	0.0390
	Mi(↑)	0.5245	0.5224	0.5084	0.5235	0.5171	0.5169	0.4913	**0.5268**	0.5240	0.5212	0.5186
	AP(↑)	0.6966	0.7000	0.6830	0.7044	0.6981	0.7033	0.6398	0.7024	0.7025	0.7019	**0.7045**
	OE(↓)	0.3466	0.3440	0.3674	**0.3426**	0.3524	0.3544	0.4562	0.3476	0.3474	0.3518	0.3582
	RL(↓)	0.0987	0.1072	0.1199	0.0851	0.1039	0.0792	0.0771	0.0987	0.0888	0.0958	**0.0757**
	CV(↓)	0.1504	0.1660	0.1802	0.1380	0.1628	0.1293	0.1221	0.1551	0.1432	0.1519	**0.1208**
	AUC(↑)	0.8590	0.8475	0.8346	0.8718	0.8495	0.8785	0.8834	0.8570	0.8673	0.8603	**0.8846**

randomly generated dataset as the training component and the remaining 20% as the testing part, repeated five times. The missing rate of class labels is set from 10% to 60% for each data set, with a step size of 10%. According to the pre-set missing rate, we randomly remove the observed labels from the training data. To avoid an empty class or instance with no positive labels, at least one instance is preserved for each class label, and each instance requires at least one positive class label.

Tables 4-15 report the mean values for each competing algorithm under different evaluations over 14 multi-label datasets with varying missing rates. Meanwhile, to conduct an achievements investigation into the relative performance of the comparing algorithms from a statistical perspective, we adopt the Friedman test [41]. At the significance level $\alpha = 0.05$, the null hypothesis of indistinguishable performance among all comparison methods is rejected for each evaluation metric. The Friedman statistics F_F for each evaluation measure and critical value are shown in Table 3, illustrating that values of F_F greater than the critical value produce significant differences between competing algorithms.

As a consequence, CLSML is considered to be the control technique, and the Bonferroni-Dunn test is employed as a post-hoc test for assessing if the average significant difference between any pair of algorithms exceeds the critical difference CD (CD $= q_\alpha \sqrt{\frac{k(k+1)}{6N}}$, where $k = 11$, $N = 14$, $q_\alpha = 3.3536$, CD $= 4.2040$). Fig. 1 depicts the CD diagrams

TABLE 12. Experimental Results of All Comparing Learning Algorithm (mean) on first Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.5.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Rcv1subset5	HM(↓)	**0.0263**	0.0276	0.0282	0.0269	0.0280	0.0285	0.0320	0.0273	0.0279	0.0271	0.0288
	Mi(↑)	0.4279	0.4535	0.4443	0.4682	0.4547	0.4571	0.3459	0.4624	0.4549	**0.4625**	0.4608
	AP(↑)	0.5430	0.5904	0.5700	0.6025	0.5875	0.6016	0.4718	0.6014	0.5953	**0.6024**	0.6005
	OE(↓)	0.4308	0.4337	0.4472	0.4287	0.4305	0.4227	0.5317	0.4245	0.4197	0.4218	**0.4188**
	RL(↓)	0.0841	0.0822	0.1043	0.0702	0.0850	0.0624	0.0832	0.0735	0.0702	0.0716	**0.0621**
	CV(↓)	0.1832	0.1917	0.2331	0.1670	0.1940	0.1539	0.1805	0.1741	0.1659	0.1708	**0.1501**
	AUC(↑)	0.8906	0.8916	0.8672	0.9062	0.8897	0.9146	0.8946	0.9017	0.9061	0.9043	**0.9154**
Rcv1subset4	HM(↓)	**0.0220**	0.0227	0.0236	0.0227	0.0225	0.0227	0.0280	**0.0220**	0.0230	0.0221	0.0238
	Mi(↑)	0.4328	0.4756	0.4658	0.4739	0.4698	0.4712	0.3261	**0.4807**	0.4737	0.4789	0.4675
	AP(↑)	0.5909	0.6527	0.6400	0.6568	0.6496	0.6564	0.4999	0.6627	0.6589	0.6620	**0.6661**
	OE(↓)	0.4197	0.3835	0.3873	0.3863	0.3782	0.4060	0.6620	**0.3778**	0.3802	0.3802	0.4030
	RL(↓)	0.0719	0.0661	0.0807	0.0573	0.0691	0.0504	0.0641	0.0581	0.0535	0.0567	**0.0476**
	CV(↓)	0.1497	0.1516	0.1789	0.1344	0.1579	0.1201	0.1352	0.1367	0.1257	0.1333	**0.1137**
	AUC(↑)	0.8988	0.9033	0.8876	0.9150	0.9001	0.9245	0.9100	0.9140	0.9206	0.9161	**0.9276**
Rcv1subset3	HM(↓)	**0.0264**	0.0275	0.0297	0.0293	0.0279	0.0285	0.0393	0.0288	0.0287	0.0290	0.0271
	Mi(↑)	0.4034	0.4349	0.4224	0.4541	0.4296	0.4534	0.3358	0.4527	0.4478	**0.4546**	0.4422
	AP(↑)	0.5320	0.5807	0.5624	0.5932	0.5703	0.5938	0.4372	0.5916	0.5887	0.5920	**0.5943**
	OE(↓)	0.4808	0.4575	0.4660	0.4542	0.4600	0.4585	0.6997	**0.4515**	0.4528	0.4542	0.4708
	RL(↓)	0.0877	0.0840	0.1026	0.0736	0.0961	0.0686	0.0861	0.0740	0.0718	0.0743	**0.0641**
	CV(↓)	0.1843	0.1899	0.2275	0.1714	0.2152	0.1596	0.1816	0.1704	0.1665	0.1723	**0.1496**
	AUC(↑)	0.8876	0.8888	0.8679	0.9009	0.8740	0.9066	0.8903	0.8993	0.9040	0.8993	**0.9110**
Rcv1subset2	HM(↓)	**0.0276**	0.0277	0.0304	0.0284	0.0288	0.0286	0.0363	0.0291	0.0280	0.0291	0.0288
	Mi(↑)	0.4166	0.4323	0.4251	0.4463	0.4273	0.4487	0.3326	0.4490	0.4348	**0.4493**	0.4469
	AP(↑)	0.5413	0.5818	0.5636	0.5915	0.5696	0.5932	0.4401	0.5907	0.5894	0.5907	**0.5940**
	OE(↓)	0.4593	0.4573	0.4595	0.4558	0.4533	0.4613	0.6925	0.4563	**0.4457**	0.4562	0.4578
	RL(↓)	0.0796	0.0870	0.1027	0.0731	0.1001	0.0652	0.0845	0.0736	0.0702	0.0726	**0.0643**
	CV(↓)	0.1725	0.1961	0.2275	0.1720	0.2211	0.1546	0.1813	0.1724	0.1644	0.1705	**0.1523**
	AUC(↑)	0.8946	0.8839	0.8655	0.9001	0.8697	0.9088	0.8900	0.8991	0.9034	0.9003	**0.9109**
Rcv1subset1	HM(↓)	**0.0307**	0.0321	0.0332	0.0313	0.0318	0.0325	0.0397	0.0314	0.0320	0.0313	0.0312
	Mi(↑)	0.4579	0.4534	0.4307	0.4706	0.4561	0.4849	0.3816	0.4699	0.4621	0.4700	**0.4792**
	AP(↑)	0.5418	0.5587	0.5333	0.5751	0.5578	0.5843	0.4453	0.5767	0.5721	0.5755	**0.5851**
	OE(↓)	0.4588	0.4707	0.4905	0.4617	0.4710	0.4602	0.6345	0.4582	0.4677	0.4625	**0.4562**
	RL(↓)	0.0781	0.0856	0.1064	0.0720	0.0872	0.0598	0.0854	0.0730	0.0707	0.0726	**0.0617**
	CV(↓)	0.1763	0.2007	0.2443	0.1737	0.2048	0.1552	0.1915	0.1761	0.1728	0.1756	**0.1526**
	AUC(↑)	0.9017	0.8932	0.8713	0.9083	0.8919	0.9188	0.8957	0.9070	0.9097	0.9072	**0.9200**
Stackex_cs	HM(↓)	**0.0106**	0.0110	0.0111	0.0112	0.0110	0.0110	0.0150	0.0109	**0.0106**	0.0109	0.0109
	Mi(↑)	0.1951	0.4358	0.4305	0.4405	0.4356	0.4390	0.3529	0.4410	0.4380	0.4406	**0.4427**
	AP(↑)	0.2424	0.4963	0.4894	0.5005	0.4943	0.4993	0.3733	0.5017	0.4995	0.5001	**0.5028**
	OE(↓)	0.7042	0.4600	0.4670	0.4604	0.4665	0.4595	0.6687	0.4513	0.4627	0.4591	**0.4503**
	RL(↓)	0.1746	0.0955	0.0983	0.0889	0.0972	0.0915	0.0850	0.0894	0.0849	0.0899	**0.0841**
	CV(↓)	0.2920	0.2041	0.2102	0.1915	0.2072	0.1965	0.1732	0.1942	0.1786	0.1947	**0.1728**
	AUC(↑)	0.8285	0.8971	0.8937	0.9035	0.8957	0.9008	0.9094	0.9023	0.9093	0.9027	**0.9098**
Stackex_chess	HM(↓)	0.0120	0.0110	0.0134	0.0114	0.0113	0.0114	0.0148	**0.0109**	0.0110	0.0163	0.0123
	Mi(↑)	0.2419	0.3694	0.3120	0.3858	0.3670	0.3750	0.2743	0.3757	0.3699	0.3397	**0.3891**
	AP(↑)	0.2938	0.4473	0.3882	0.4588	0.4404	0.4606	0.3444	0.4586	0.4494	0.4322	**0.4634**
	OE(↓)	0.6478	0.4669	0.5606	0.4681	0.4848	0.4637	0.6394	0.4627	0.4818	0.5677	**0.4621**
	RL(↓)	0.1663	0.1578	0.1893	0.1370	0.1551	0.1287	0.1533	0.1446	0.1571	0.1390	**0.1221**
	CV(↓)	0.2947	0.3042	0.3578	0.2688	0.3029	0.2560	0.2933	0.2843	0.3059	0.2505	**0.2441**
	AUC(↑)	0.8256	0.8346	0.8016	0.8556	0.8369	0.8641	0.8364	0.8456	0.8338	0.8502	**0.8706**

for each evaluation metric. The comparison methods with red lines connecting to CLSML in each subgraph have a distance less than CD from CLSML, showing statistical similarity with CLSML under the respective evaluation criteria. The performance of our proposed approach CLSML is described in depth below.

- Tables 4–15 present the experimental findings from 11 methods on 14 different datasets. The best experimental findings are shown in bold in the results. According to these findings, CLSML is optimal in more than five evaluation metrics on five datasets and best in more than four evaluation metrics on fourteen datasets with missing rates ranging from 0.1 to 0.6. CLSML surpasses all comparison algorithms in terms of Ranking Loss, Coverage, and Macro AUC, but Average Precison is perfect on all datasets except the ''Rcv1subset5'' and ''Languagelog'' datasets. The exceptional performance of our suggested method demonstrates the effectiveness of learning common and label-specific features, as well as the superiority of label space recovery via sparse high-order label correlations. Simultaneously, we found that on all 14 datasets, under HL evaluation, our method was only slightly lower than the optimal method by 0.001-0.002. When the missing label rate is in the range from 0.3 to 0.6, the GLOCAL method has the best HL evaluation, which is partly attributed to the GLOCAL method considering both global and local label correlations. When the missing rate is between 0.1 and 0.2, HL better

TABLE 13. Experimental Results of All Comparing Learning Algorithm (mean) on last Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.5.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Science	HM(↓)	**0.0383**	0.0394	0.0408	0.0399	0.0399	0.0390	0.0471	0.0407	0.0401	0.0399	0.0410
	Mi(↑)	0.4026	0.4096	0.3894	0.4119	0.3999	**0.4131**	0.3309	0.4093	0.3992	0.4100	0.3999
	AP(↑)	0.5654	0.5766	0.5582	0.5752	0.5707	0.5764	0.4973	0.5746	0.5739	0.5711	**0.5768**
	OE(↓)	0.5240	0.5116	0.5332	0.5106	0.5180	**0.5078**	0.6526	0.5136	0.5216	0.5110	0.5324
	RL(↓)	0.1373	0.1469	0.1700	0.1297	0.1534	0.1298	0.1218	0.1421	0.1310	0.1380	**0.1139**
	CV(↓)	0.1902	0.2105	0.2372	0.1888	0.2178	0.1892	0.1675	0.2052	0.1866	0.2003	**0.1646**
	AUC(↑)	0.8239	0.8087	0.7840	0.8267	0.7988	0.8261	0.8387	0.8133	0.8271	0.8162	**0.8457**
Recreation	HM(↓)	**0.0658**	0.0698	0.0715	0.0659	0.0699	0.0685	0.0748	0.0690	0.0661	0.0670	0.0659
	Mi(↑)	**0.4371**	0.4223	0.4100	0.4355	0.4244	0.4314	0.3592	0.4327	0.4237	0.4302	0.4316
	AP(↑)	0.6208	0.6111	0.6007	0.6169	0.6099	0.6221	0.5597	0.6177	0.6170	0.6200	**0.6229**
	OE(↓)	0.4736	0.4816	0.4962	0.4764	0.4832	0.4776	0.5726	0.4768	0.4798	0.4746	**0.4735**
	RL(↓)	0.1531	0.1696	0.1766	0.1621	0.1672	0.1530	0.1551	0.1612	0.1521	0.1583	**0.1429**
	CV(↓)	0.2067	0.2262	0.2330	0.2175	0.2253	0.2071	0.1978	0.2166	0.2025	0.2147	**0.1918**
	AUC(↑)	0.7933	0.7765	0.7707	0.7833	0.7789	0.7928	0.7880	0.7859	0.7953	0.7877	**0.8015**
Languagelog	HM(↓)	0.0202	**0.0163**	0.0241	0.0165	0.0165	0.0169	0.0583	**0.0163**	0.0165	0.0165	0.0204
	Mi(↑)	0.2035	0.1939	0.2166	0.1942	0.1792	0.2077	0.1227	0.2081	0.1870	0.2048	**0.2192**
	AP(↑)	0.2895	0.3170	0.2943	0.3276	0.3200	0.3251	0.1838	0.3287	0.3200	**0.3313**	0.3206
	OE(↓)	0.7889	0.7669	0.7766	0.7498	0.7560	0.7594	0.8951	**0.7457**	0.7560	0.7526	0.7718
	RL(↓)	0.1559	0.1765	0.2153	0.1593	0.1739	0.1584	0.2988	0.1604	0.1722	0.1551	**0.1339**
	CV(↓)	0.1954	0.2176	0.2607	0.1986	0.2189	0.1985	0.3447	0.2001	0.2149	0.1962	**0.1723**
	AUC(↑)	0.6836	0.6678	0.6289	0.6846	0.6677	0.6840	0.5474	0.6808	0.6687	0.6848	**0.7033**
Arts	HM(↓)	**0.0660**	0.0682	0.0687	0.0681	0.0688	0.0701	0.0749	0.0671	0.0670	0.0686	0.0681
	Mi(↑)	**0.4255**	0.4002	0.3965	0.4079	0.3980	0.4080	0.2837	0.4074	0.4006	0.4051	0.3904
	AP(↑)	0.6016	0.5936	0.5918	0.6022	0.5938	0.6072	0.5195	0.6016	0.6028	0.6041	**0.6083**
	OE(↓)	**0.4764**	0.4898	0.4958	0.4826	0.4930	0.4876	0.6352	0.4894	0.4920	0.4844	0.5018
	RL(↓)	0.1437	0.1644	0.1674	0.1551	0.1618	0.1400	0.1400	0.1540	0.1439	0.1529	**0.1321**
	CV(↓)	0.2227	0.2498	0.2546	0.2392	0.2470	0.2184	0.2083	0.2377	0.2222	0.2349	**0.2041**
	AUC(↑)	0.8069	0.7806	0.7798	0.7905	0.7847	0.8081	0.8150	0.7926	0.8063	0.7937	**0.8187**
Social	HM(↓)	**0.0219**	0.0231	0.0254	0.0223	0.0232	0.0224	0.0290	0.0229	0.0230	0.0228	0.0225
	Mi(↑)	0.6226	0.6176	0.5851	0.6245	0.6169	0.6255	0.5258	0.6209	0.6153	0.6244	**0.6270**
	AP(↑)	0.7411	0.7670	0.7394	0.7682	0.7645	0.7738	0.6916	0.7681	0.7639	0.7710	**0.7744**
	OE(↓)	0.3058	0.2844	0.3162	0.2902	0.2872	**0.2838**	0.4230	0.2882	0.2912	0.2858	0.2896
	RL(↓)	0.0936	0.0831	0.1052	0.0745	0.0844	0.0692	0.0669	0.0776	0.0836	0.0734	**0.0660**
	CV(↓)	0.1322	0.1229	0.1463	0.1113	0.1234	0.1042	0.0942	0.1153	0.1210	0.1098	**0.0912**
	AUC(↑)	0.8611	0.8752	0.8533	0.8860	0.8737	0.8919	0.8975	0.8824	0.8763	0.8864	**0.8998**
Entertainment	HM(↓)	**0.0592**	0.0633	0.0640	0.0610	0.0609	0.0612	0.0669	0.0606	0.0606	0.0629	0.0629
	Mi(↑)	**0.5213**	0.4998	0.4912	0.5089	0.5024	0.5054	0.4146	0.5071	0.5019	0.5067	0.5005
	AP(↑)	0.7019	0.6928	0.6825	0.7026	0.6906	0.7010	0.6332	0.6974	0.6965	0.6989	**0.7047**
	OE(↓)	**0.3728**	0.3850	0.3874	0.3778	0.3846	0.3812	0.5054	0.3810	0.3872	0.3770	0.3842
	RL(↓)	0.1162	0.1334	0.1462	0.1133	0.1367	0.1151	0.1103	0.1246	0.1159	0.1241	**0.1036**
	CV(↓)	0.1695	0.1884	0.2037	0.1651	0.1932	0.1669	0.1496	0.1791	0.1667	0.1775	**0.1479**
	AUC(↑)	0.8357	0.8142	0.8001	0.8375	0.8082	0.8343	0.8455	0.8227	0.8359	0.8248	**0.8492**
Computers	HM(↓)	**0.0382**	0.0408	0.042	0.0397	0.0410	0.0397	0.0453	0.0394	0.0397	0.0401	0.0388
	Mi(↑)	**0.5220**	0.5082	0.4954	0.5180	0.5104	0.5116	0.4848	0.5175	0.5108	0.5130	0.5104
	AP(↑)	0.6930	0.6911	0.6757	0.6962	0.6870	0.6977	0.6370	0.6977	0.6935	0.6959	**0.6983**
	OE(↓)	0.3480	0.3558	0.3732	**0.3468**	0.3614	0.3568	0.4564	0.3490	0.3596	0.3550	0.3636
	RL(↓)	0.1038	0.1155	0.1283	0.0911	0.1174	0.0860	0.0833	0.1058	0.0968	0.1019	**0.0799**
	CV(↓)	0.1558	0.1761	0.1911	0.1460	0.1799	0.1388	0.1330	0.1647	0.1523	0.1578	**0.1284**
	AUC(↑)	0.8519	0.8374	0.8222	0.8630	0.8322	0.8678	0.8724	0.8467	0.8570	0.8529	**0.8776**

methods such as CLML, CIMML and JLCLS all consider example correlation, indicating the important role of local and example correlation in feature selection. This is also an aspect that we need to consider in our further work.

- According to the Bonferroni-Dunn posterior results for 7 evaluation metrics on 14 datasets (Fig. 1), CLSML is statistically ranked first in four metrics, including average precision, ranking loss, coverage, and AUC, fifth in Micro F1 and One Error, and seventh in Hamming loss, respectively. Besides, we found that LLSF ranks last among the seven indicators, indicating that considering only label correlation constraints on matrix W and label specific features is far from enough. LSFCI is inferior to our proposed method in all metrics except HM, possibly due to only considering instance correlation and label specific features, while ignoring high-order label correlation and common features. Our approach is more competitive compared to LSML in four metrics, but slightly inferior in three metrics with little difference, indicating that it is useful to consider both second-order label correlation constraints on the output space and common features simultaneously. We can see that CLML performs exceptionally well in metrics such as HL, Mi, and OE, and is also highly competitive in other indices. The fundamental explanation for this is that HL and Micro F1 are example-based metrics, whereas CLML includes not just instance correlations but also

TABLE 14. Experimental Results of All Comparing Learning Algorithm (mean) on first Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.6.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Rcv1subset5	HM(↓)	**0.0262**	0.0288	0.0291	0.0282	0.0287	0.0295	0.0335	0.0280	0.0291	0.0280	0.0294
	Mi(↑)	0.4189	0.4331	0.4104	0.4397	0.4300	**0.4448**	0.3241	0.4389	0.4330	0.4439	0.4374
	AP(↑)	0.5240	0.5646	0.5417	0.5771	0.5593	**0.5825**	0.4550	0.5793	0.5711	0.5781	0.5775
	OE(↓)	0.4423	0.4518	0.4633	0.4415	0.4498	0.4452	0.6127	0.4443	0.4513	0.4447	**0.4393**
	RL(↓)	0.0939	0.0956	0.1205	0.0827	0.1001	0.0698	0.0857	0.0824	0.0798	0.0819	**0.0679**
	CV(↓)	0.1985	0.2212	0.2645	0.1937	0.2292	0.1681	0.1876	0.1949	0.1869	0.1923	**0.1632**
	AUC(↑)	0.8809	0.8755	0.8485	0.8908	0.8697	0.9052	0.8904	0.8909	0.8954	0.8913	**0.9082**
Rcv1subset4	HM(↓)	**0.0227**	0.0228	0.0230	0.0242	**0.0227**	0.0247	0.0276	0.0241	0.0219	0.0244	0.0242
	Mi(↑)	0.4148	0.4353	0.4213	0.4555	**0.4278**	0.4483	0.3028	0.4546	0.4347	0.4522	0.4326
	AP(↑)	0.5744	0.6271	0.6110	0.6335	0.6207	0.6332	0.4927	0.6343	0.6360	0.6325	**0.6382**
	OE(↓)	0.4480	0.4010	0.4082	0.4095	0.4043	0.4177	0.6680	0.4005	**0.3957**	0.4117	0.4283
	RL(↓)	0.0774	0.0811	0.0991	0.0702	0.0862	0.0593	0.0705	0.0723	0.0656	0.0687	**0.0585**
	CV(↓)	0.1633	0.1837	0.2194	0.1648	0.1930	0.1411	0.1510	0.1660	0.1546	0.1566	**0.1372**
	AUC(↑)	0.8909	0.8825	0.8610	0.8968	0.8776	0.9107	0.8999	0.8936	0.9026	0.8999	**0.9125**
Rcv1subset3	HM(↓)	**0.0267**	0.0277	0.0300	0.0282	0.0284	0.0296	0.0361	0.0287	0.0294	0.0279	0.0305
	Mi(↑)	0.4049	0.4147	0.3981	0.4271	0.4103	0.4380	0.3314	0.4248	0.4196	**0.4271**	0.4205
	AP(↑)	0.5154	0.5539	0.5338	0.5688	0.5415	0.5687	0.4344	0.5641	0.5657	0.5681	**0.5693**
	OE(↓)	0.4805	**0.4605**	0.4757	0.4610	0.4778	0.4647	0.6853	0.4617	0.4618	0.4615	0.4670
	RL(↓)	0.0958	0.1017	0.1210	0.0852	0.1159	0.0762	0.0919	0.0876	0.0794	0.0845	**0.0739**
	CV(↓)	0.2007	0.2285	0.2675	0.1966	0.2537	0.1784	0.1966	0.2007	0.1844	0.1932	**0.1745**
	AUC(↑)	0.8790	0.8700	0.8483	0.8881	0.8537	0.8982	0.8840	0.8855	0.8958	0.8887	**0.9015**
Rcv1subset2	HM(↓)	**0.0270**	0.0280	0.0291	0.0277	0.0288	0.0293	0.0408	0.0291	0.0277	0.0291	0.0321
	Mi(↑)	0.3906	0.4111	0.3981	0.4170	0.3997	**0.4350**	0.3416	0.4304	0.4167	0.4309	0.4216
	AP(↑)	0.4992	0.5527	0.5354	0.5702	0.5388	0.5711	0.4452	0.5723	0.5675	0.5733	**0.5738**
	OE(↓)	0.4905	0.4732	0.4770	0.4625	0.4733	0.4620	0.6440	0.4567	**0.4547**	0.4558	0.4612
	RL(↓)	0.0958	0.1008	0.1226	0.0848	0.1180	0.0726	0.0898	0.0829	0.0794	0.0817	**0.0718**
	CV(↓)	0.2042	0.2280	0.2682	0.1960	0.2586	0.1653	0.1947	0.1930	0.1866	0.1905	**0.1703**
	AUC(↑)	0.8778	0.8687	0.8467	0.8882	0.8508	0.9050	0.8846	0.8893	0.8941	0.8908	**0.9023**
Rcv1subset1	HM(↓)	**0.0320**	0.0323	0.0352	0.0324	0.0323	0.0334	0.0411	0.0328	0.0340	0.0323	0.0347
	Mi(↑)	0.4342	0.4340	0.4057	0.4508	0.4298	0.4549	0.3734	0.4558	0.4474	0.4562	**0.4587**
	AP(↑)	0.5246	0.5327	0.5044	0.5539	0.5279	0.5525	0.4519	0.5534	0.5521	0.5566	**0.5581**
	OE(↓)	0.4685	0.4985	0.5085	0.4720	0.4993	**0.4672**	0.6067	0.4753	0.4757	0.4705	0.4730
	RL(↓)	0.0810	0.0983	0.1221	0.0813	0.1024	0.0699	0.0881	0.0830	0.0780	0.0817	**0.0682**
	CV(↓)	0.1822	0.2298	0.2768	0.1939	0.2389	0.1662	0.1995	0.2004	0.1874	0.1985	**0.1655**
	AUC(↑)	0.8990	0.8797	0.8552	0.8979	0.8761	0.9157	0.8924	0.8960	0.9024	0.8978	**0.9131**
Stackex_cs	HM(↓)	**0.0101**	0.0114	0.0113	0.0111	0.0109	0.0109	0.0144	0.0110	0.0111	0.0111	0.0110
	Mi(↑)	0.1424	0.4200	0.4175	0.4282	0.4220	0.4269	0.3472	0.4285	0.4262	0.4294	**0.4298**
	AP(↑)	0.1953	0.4796	0.4761	0.4863	0.4781	0.4833	0.3696	0.4865	0.4870	0.4855	**0.4896**
	OE(↓)	0.7535	0.4712	0.4784	0.4689	0.4739	0.4710	0.6626	0.4723	0.4683	0.4725	**0.4674**
	RL(↓)	0.2257	0.1042	0.1058	0.0991	0.1076	0.1009	0.0885	0.0981	0.0942	0.0983	**0.0856**
	CV(↓)	0.3677	0.2220	0.2257	0.2130	0.2302	0.2172	0.1815	0.2115	0.1961	0.2102	**0.1805**
	AUC(↑)	0.7738	0.8883	0.8859	0.8933	0.8840	0.8908	0.9025	0.8940	0.9005	0.8942	**0.9031**
Stackex_chess	HM(↓)	0.0122	0.0112	0.0135	0.0112	0.0115	0.0115	0.0137	**0.0109**	0.0114	0.0158	0.0120
	Mi(↑)	0.2110	0.3305	0.2959	0.3594	0.3395	0.3607	0.2464	0.3551	0.3338	0.3286	**0.3732**
	AP(↑)	0.2630	0.4124	0.3651	0.4396	0.4175	0.4434	0.3285	0.4381	0.4143	0.4160	**0.4489**
	OE(↓)	0.6961	0.5093	0.5761	0.4848	0.5093	0.4800	0.6484	0.4782	0.5248	0.5831	**0.4728**
	RL(↓)	0.1794	0.1761	0.2167	0.1525	0.1762	0.1441	0.1721	0.1508	0.1708	0.1506	**0.1308**
	CV(↓)	0.3157	0.3317	0.3948	0.2957	0.3334	0.2827	0.3202	0.2931	0.3218	0.2708	**0.2593**
	AUC(↑)	0.8107	0.8149	0.7810	0.8392	0.8171	0.8474	0.8230	0.8388	0.8251	0.8371	**0.8627**

common and label-specific features. CIMML ranks first in the Mi metric and overall high in other metrics, possibly because CIMML considers instance relevance on the basis of LSML. LRMML is highly competitive in metrics other than HM, indicating that its use of auxiliary label matrices and forced maximum separation between different label subspaces help to recover missing labels and better distinguish labels. Although SGMML considers both global sparsity constraints and high-order label correlations, its overall ranking on seven metrics is lower, possibly because SGMML only uses row sparsity $l_{2,1}$-norm constraints and ignores the overall constraints of l_1-norm. GLOCAL is far ahead in HM, but relatively weak in other indicators, indicating that considering only global and local correlations is no longer sufficient to meet most needs. JLCLS considers both high-order label correlation and label specific features, but fail to consider common features and second-order label correlation constraints on the output space. Therefore, the overall ranking of the seven indicators is weak.

- Extensive investigations show that the incompleteness of class labels has a considerable impact on the performance of multi-label classifiers, and techniques dealing with missing labels usually outperform unprocessed ones in most scenarios. For example, the performance of LLSF drops rapidly when the missing rate increases, because LLSF fails to rebuild the missing labels. While

TABLE 15. Experimental Results of All Comparing Learning Algorithm (mean) on last Seven Datasets in terms of Seven Evaluation Metrics, while misRate=0.6.

DataSet	Eval	GLOCAL	JLCLS	LLSF	CIMML	SGMML	LRMML	SMILE	CLML	LSF-CI	LSML	CLSML
Science	HM(↓)	**0.0391**	0.0414	0.0432	0.0416	0.0416	0.0407	0.0484	0.0403	0.0416	0.0398	0.0420
	Mi(↑)	0.3855	0.3902	0.3670	0.4000	0.3840	**0.4013**	0.3304	0.3891	0.3853	0.3928	0.3863
	AP(↑)	0.5473	0.5602	0.5335	0.5633	0.5509	0.5612	0.4902	0.5612	0.5604	0.5624	**0.5642**
	OE(↓)	0.5460	0.5276	0.5598	0.5262	0.5422	0.5282	0.6572	0.5330	0.5448	**0.5246**	0.5380
	RL(↓)	0.1416	0.1652	0.1829	0.1397	0.1685	0.1401	0.1310	0.1514	0.1418	0.1521	**0.1210**
	CV(↓)	0.1943	0.2312	0.2526	0.2000	0.2360	0.2013	0.1819	0.2149	0.2007	0.2160	**0.1729**
	AUC(↑)	0.8193	0.7889	0.7709	0.8163	0.7850	0.8150	0.8273	0.8047	0.8155	0.8031	**0.8366**
Recreation	HM(↓)	**0.0669**	0.0698	0.0674	0.0687	0.0694	0.0687	0.0744	0.0714	0.0678	0.0686	0.0682
	Mi(↑)	**0.4305**	0.4152	0.3963	0.4230	0.4108	0.4269	0.3564	0.4176	0.4123	0.4209	0.4123
	AP(↑)	0.6055	0.5967	0.5824	0.6034	0.5937	0.6130	0.5538	0.6066	0.6048	0.6050	**0.6096**
	OE(↓)	0.4842	0.4964	0.5152	0.4908	0.5020	0.4848	0.5800	0.4858	0.4980	0.4902	**0.4837**
	RL(↓)	0.1590	0.1798	0.1921	0.1726	0.1798	0.1607	0.1585	0.1716	0.1601	0.1701	**0.1486**
	CV(↓)	0.2112	0.2355	0.2501	0.2289	0.2373	0.2150	0.1984	0.2276	0.2116	0.2260	**0.1962**
	AUC(↑)	0.7890	0.7671	0.7543	0.7739	0.7670	0.7854	0.7855	0.7760	0.7857	0.7763	**0.7942**
Languagelog	HM(↓)	0.0206	0.0166	0.0286	0.0167	0.0168	**0.0166**	0.0611	0.0165	0.0171	0.0168	0.0208
	Mi(↑)	0.1903	0.1887	0.2096	0.1946	0.1782	0.1985	0.1216	0.1930	0.1785	0.1840	**0.2120**
	AP(↑)	0.2768	0.3091	0.2763	0.3140	0.3034	0.3187	0.1791	0.3181	0.3032	**0.3187**	0.3118
	OE(↓)	0.8047	0.7615	0.7971	0.7642	0.7711	0.7587	0.9013	**0.7457**	0.7806	0.7622	0.7841
	RL(↓)	0.1618	0.1976	0.2357	0.1750	0.1999	0.1761	0.3161	0.1793	0.1903	0.1710	**0.1490**
	CV(↓)	0.1994	0.2423	0.2838	0.2153	0.2466	0.2168	0.3616	0.2214	0.2319	0.2087	**0.1873**
	AUC(↑)	0.6784	0.6464	0.6090	0.6687	0.6445	0.6679	0.5346	0.6611	0.6534	0.6735	**0.6894**
Arts	HM(↓)	0.0674	0.0708	0.0696	0.0682	0.0694	**0.0666**	0.0776	0.0681	0.0679	0.0682	0.0720
	Mi(↑)	**0.4135**	0.3984	0.3781	0.3951	0.3889	0.3941	0.2847	0.3973	0.3874	0.3948	0.3888
	AP(↑)	0.5977	0.5907	0.5782	0.5914	0.5837	0.5962	0.5174	0.5945	0.5943	0.5964	**0.5978**
	OE(↓)	0.4942	**0.4920**	0.5138	0.4990	0.5060	0.4984	0.6378	0.4928	0.5072	**0.4920**	0.5158
	RL(↓)	0.1518	0.1688	0.1795	0.1619	0.1713	0.1447	0.1429	0.1629	0.1472	0.1611	**0.1359**
	CV(↓)	0.2325	0.2564	0.2690	0.2477	0.2590	0.2235	0.2104	0.2481	0.2240	0.2483	**0.2095**
	AUC(↑)	0.7986	0.7762	0.7658	0.7821	0.7741	0.8022	0.8108	0.7830	0.8045	0.7844	**0.8146**
Social	HM(↓)	**0.0215**	0.0239	0.0266	0.0234	0.0242	0.0233	0.0304	0.0236	0.0246	0.0234	0.0230
	Mi(↑)	0.6187	0.6054	0.5700	0.6135	0.6023	0.6170	0.5218	0.6129	0.6005	0.6145	**0.6199**
	AP(↑)	0.7399	0.7526	0.7209	0.7579	0.7521	0.7666	0.6894	0.7616	0.7537	0.7605	**0.7666**
	OE(↓)	0.3106	0.2976	0.3392	0.2970	0.2972	0.2934	0.4246	0.2954	0.2972	0.2950	**0.2926**
	RL(↓)	0.0928	0.0945	0.1202	0.0838	0.0966	0.0749	0.0714	0.0840	0.0923	0.0820	**0.0698**
	CV(↓)	0.1301	0.1346	0.1617	0.1220	0.1366	0.1101	0.1045	0.1226	0.1315	0.1203	**0.1008**
	AUC(↑)	0.8627	0.8643	0.8394	0.8755	0.8624	0.8862	0.8929	0.8757	0.8662	0.8766	**0.8950**
Entertainment	HM(↓)	0.0622	0.0632	0.0644	**0.0610**	0.0618	0.0620	0.0684	0.0631	0.0614	0.0630	0.0622
	Mi(↑)	**0.5100**	0.4866	0.4787	0.5033	0.4936	0.4923	0.4105	0.4919	0.4930	0.4935	0.4941
	AP(↑)	0.6968	0.6773	0.6693	0.6961	0.6803	0.6920	0.6299	0.6869	0.6881	0.6876	**0.6979**
	OE(↓)	**0.3816**	0.4040	0.4084	0.3858	0.3984	0.3922	0.5076	0.3946	0.3996	0.3948	0.3936
	RL(↓)	0.1196	0.1447	0.1588	0.1200	0.1445	0.1217	0.1139	0.1331	0.1232	0.1299	**0.1043**
	CV(↓)	0.1715	0.2023	0.2162	0.1740	0.2021	0.1745	0.1544	0.1876	0.1739	0.1845	**0.1512**
	AUC(↑)	0.8326	0.8016	0.7890	0.8283	0.8031	0.8270	0.8407	0.8149	0.8284	0.8183	**0.8489**
Computers	HM(↓)	**0.0391**	0.0423	0.0428	0.0403	0.0408	0.0405	0.0447	0.0399	0.0405	0.0418	0.0397
	Mi(↑)	**0.5167**	0.5006	0.4841	0.5106	0.4966	0.5089	0.4812	0.5069	0.5090	0.5001	0.5102
	AP(↑)	0.6790	0.6774	0.6610	0.6921	0.6751	0.6956	0.6335	0.6879	0.6894	0.6841	**0.6933**
	OE(↓)	0.3574	0.3660	0.3898	**0.3548**	0.3732	0.3596	0.4578	0.3570	0.3612	0.3650	0.3654
	RL(↓)	0.1145	0.1293	0.1421	0.0985	0.1291	0.0904	0.0873	0.1185	0.1033	0.1125	**0.0802**
	CV(↓)	0.1712	0.1932	0.2089	0.1553	0.1947	0.1451	0.1385	0.1809	0.1608	0.1737	**0.1298**
	AUC(↑)	0.8388	0.8215	0.8088	0.8541	0.8192	0.8630	0.8678	0.8337	0.8503	0.8392	**0.8744**

the efficacy of these techniques for modeling missing labels, such as LSML, SMILE, JLCLS, CIMML, LRMML and GLOCAL, declines rather slowly as the missing rate increases. In contrast, the overall performance of our suggested method greatly surpasses all contrasting algorithms. This conclusion is attributed to the modeling of missing label recovery by using label correlations and learning common as well as label-specific features between class labels. When the missing rate is low, our proposed methods in Mi and OE cannot compete with most algorithms. However, as the missing rate rises, our method gradually closes the gap with other algorithms, and it has achieved transcendence on multiple datasets, including Stackex_chess and Social. According to the experimental results and analyses, we can conclude that our proposed approach outperforms other well-established multi-label classification algorithms in addressing multi-label learning with missing labels.

E. ANALYSIS OF ABLATION EXPERIMENTS

1) ANALYSIS OF ABLATION EXPERIMENT ON L_1-NORM AND $L_{2,1}$-NORM

Although the $l_{2,1}$-norm can be understood as a combination of the l_1-norm and the l_2-norm, and it can select differentiating features for all instances with joint sparsity, its disadvantages are equally clear. It disregards the labels' distinguishing characteristics or the redundant correlation of features [11].

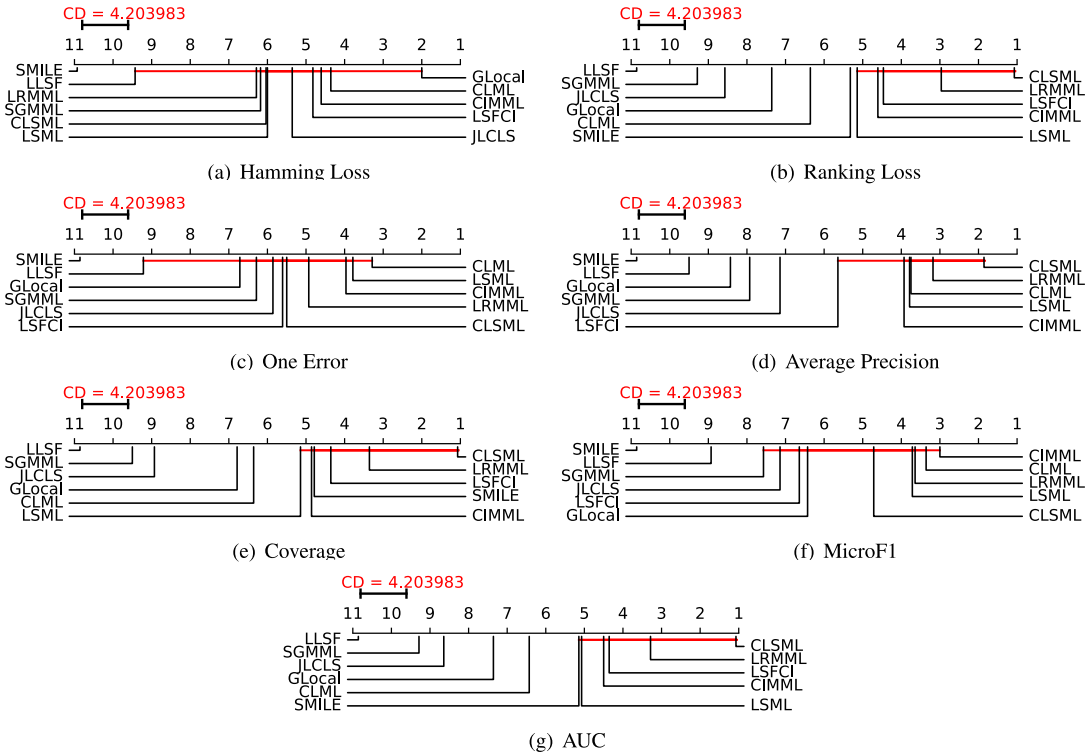


FIGURE 1. Comparing CLSML against other competing algorithms with Bonferroni-Dunn test.

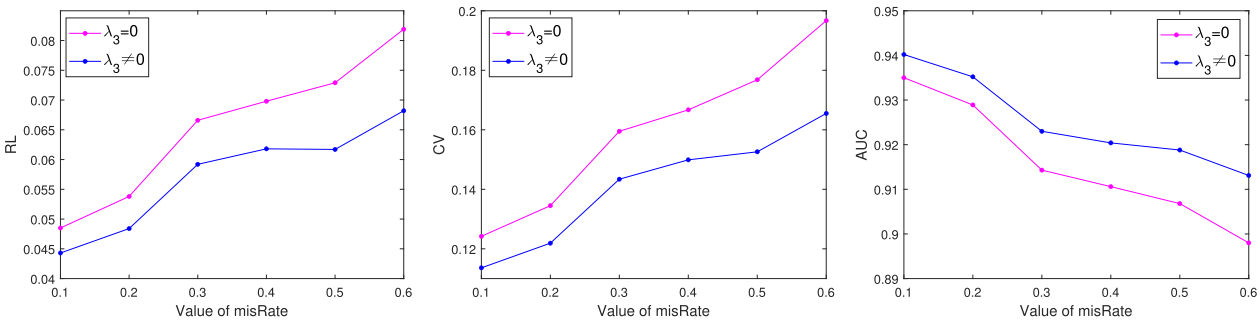


FIGURE 2. Analysis of ablation experiments for label correlation on Rcv1subset1 dataset while missing rate from 0.1 to 0.6.

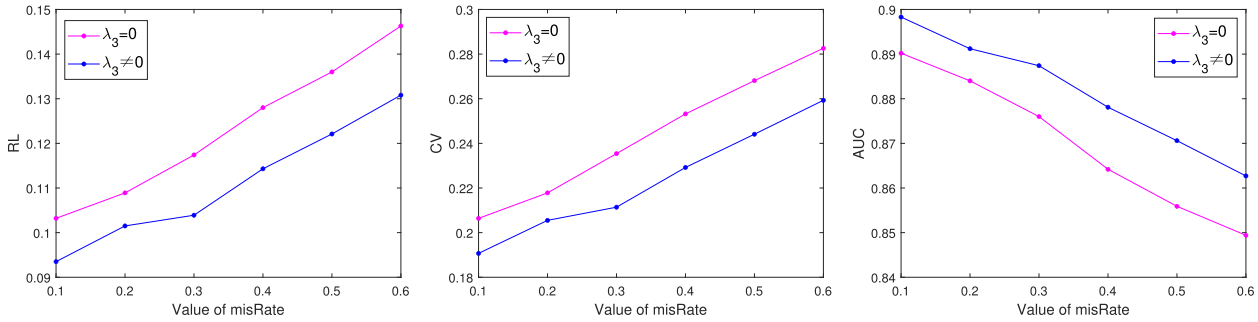


FIGURE 3. Analysis of ablation experiments for label correlation on Stackex_chess dataset while missing rate from 0.1 to 0.6.

The l_1 norm allows the coefficients of some of the features to be lowered to zero, resulting in indirect feature selection, and it is appropriate for situations when the features are correlated. So, we integrate the l_1- and l_2-norms to learn label-specific and common properties. To demonstrate that the combination of these two norms may yield the best

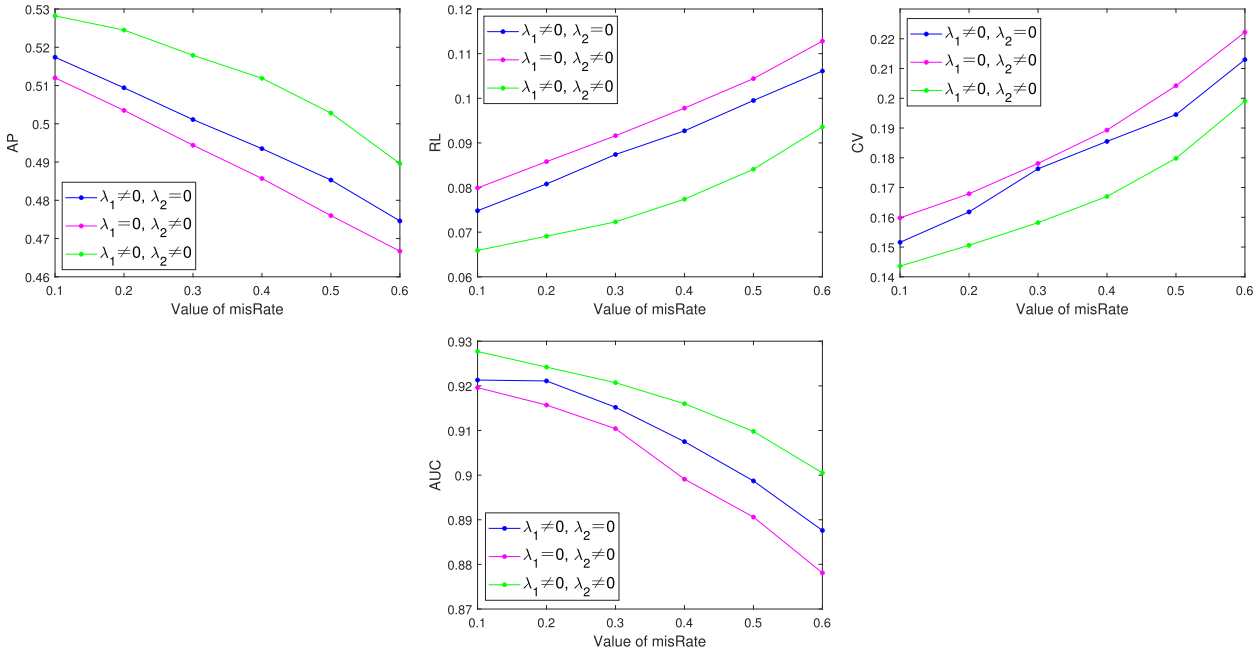


FIGURE 4. Analysis of ablation experiments for l_1-norm and $l_{2,1}$-norm on Stackex_cs dataset while missing rate from 0.1 to 0.6.

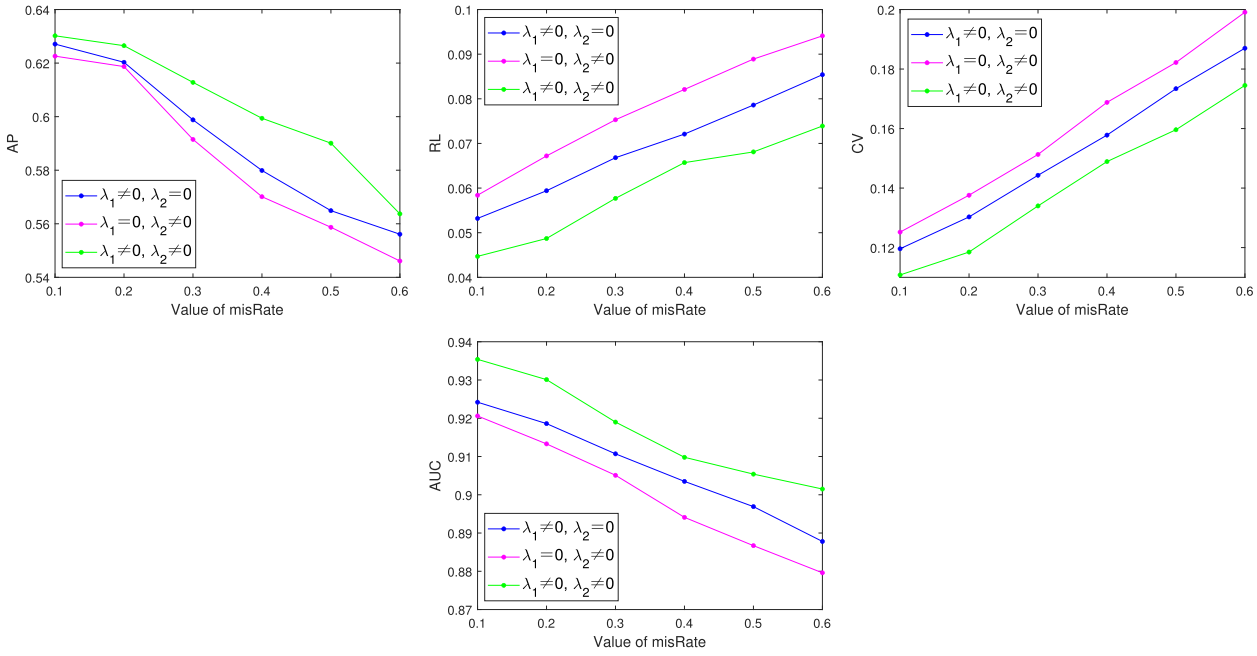


FIGURE 5. Analysis of ablation experiments for l_1-norm and $l_{2,1}$-norm on Rcv1subset3 dataset while missing rate from 0.1 to 0.6.

classification model performance, we present the $l_{2,1}$-norm and l_1-norm ablation experiments.

By conducting experiments on three different scenarios as shown in Figs. 4–6: $\lambda_1 \neq 0$ and $\lambda_2 = 0$, $\lambda_1 = 0$ and $\lambda_2 \neq 0$, and $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$, where λ_1 controls the l_1-norm and λ_2 controls the $l_{2,1}$-norm, it is obviously that only when λ_1 and λ_2 are not equal to 0, the performance is optimal. $\lambda_1 \neq 0$ and $\lambda_2 = 0$ comes second, while $\lambda_1 = 0$ and $\lambda_2 \neq 0$ is the worst. This result proves that $l_{2,1}$-norm cannot replace l_1-norm in forcing row sparsity to select labels-specific features, and for constraints on both row sparsity and column sparsity, using only l_1- norm cannot achieve the desired effect.

2) ANALYSIS OF ABLATION EXPERIMENT ON TWO-STAGE SECOND-ORDER LABEL CORRELATIONS

Learning high-order correlations in the training stage can greatly improve the performance of classification models, it is equally important to add a label correlation constraint to

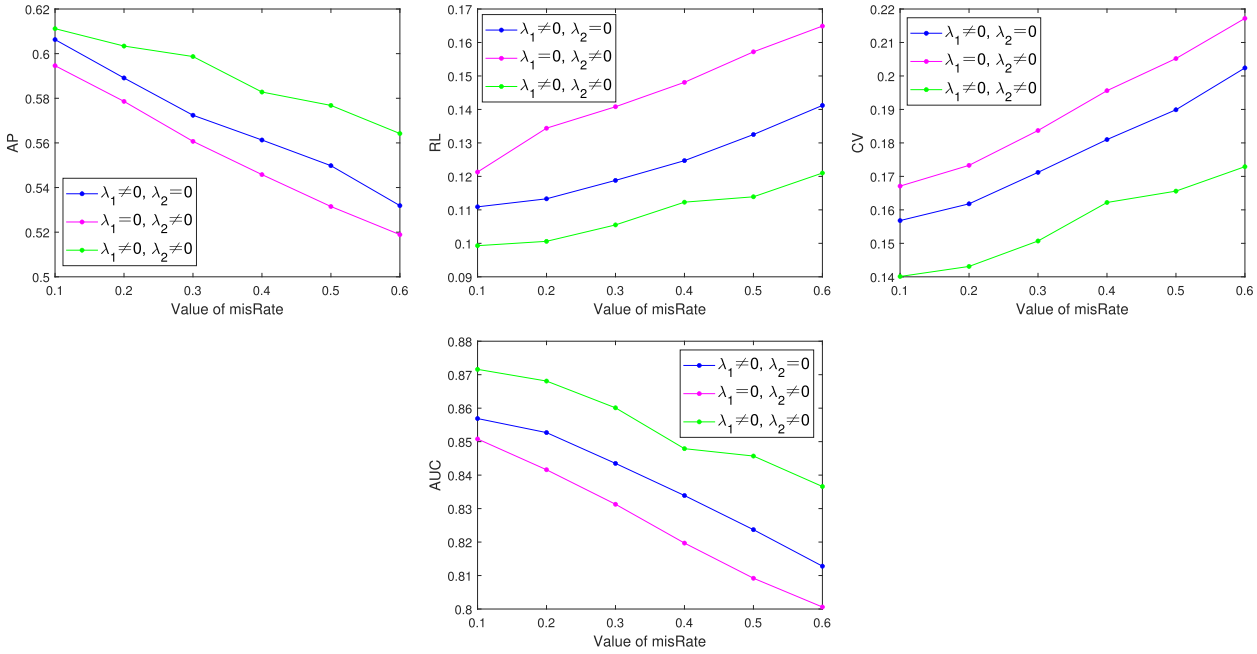


FIGURE 6. Analysis of ablation experiments for l_1-norm and $l_{2,1}$-norm on Science dataset while missing rate from 0.1 to 0.6.

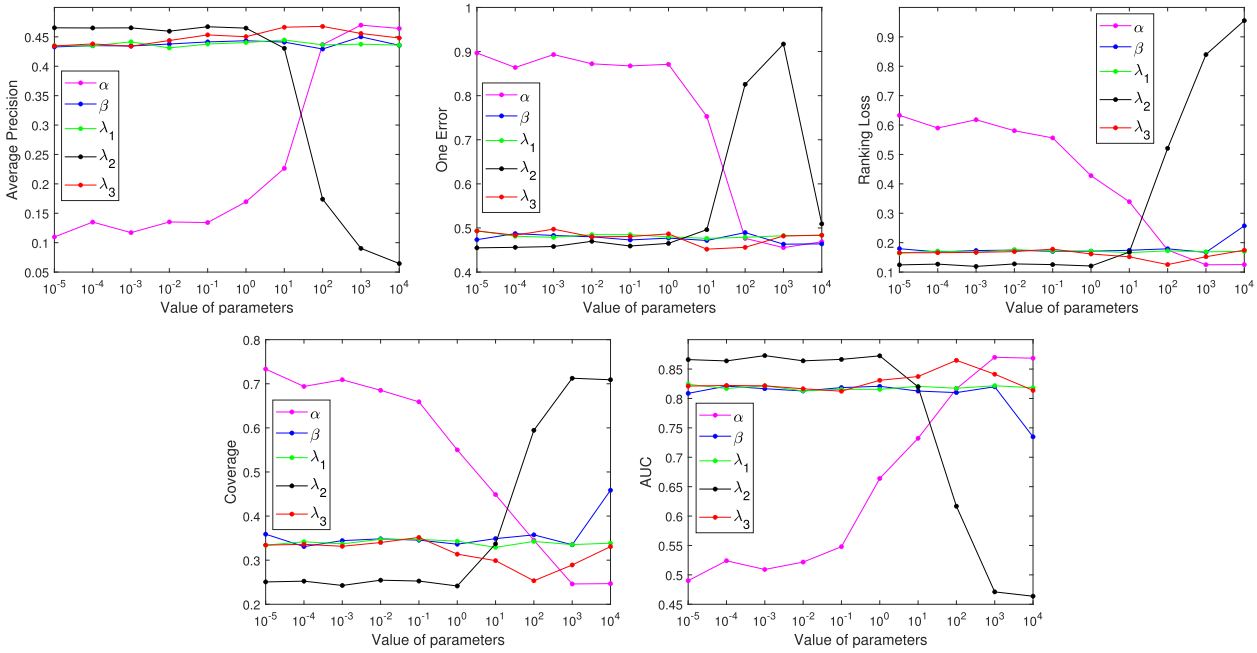


FIGURE 7. Sensitivity analysis of parameters on Stackex_chess dataset with missing rate 0.5.

the output of the model. In order to improve the consistency of the feature-label space, a two-stage second-order label correlation learning technique based on cosine similarity is developed to further constraint the label output, where it is controlled by λ_3. To demonstrate its effectiveness, we introduce ablation experiments on two-stage second-order label correlations and its results are shown in Figs. 2–3, where $\lambda_3 = 0$ represents that the model has only learned high-order label correlations without considering the two-stage second-order label correlations. It is obvious that learning the labels correlations of both high-order and second-order simultaneously can improve the model performance.

F. SENSITIVITY ANALYSIS OF PARAMETERS

We have five parameters in the proposed approach: α, β, λ_1, λ_2, and λ_3. Control the loss between the recovered

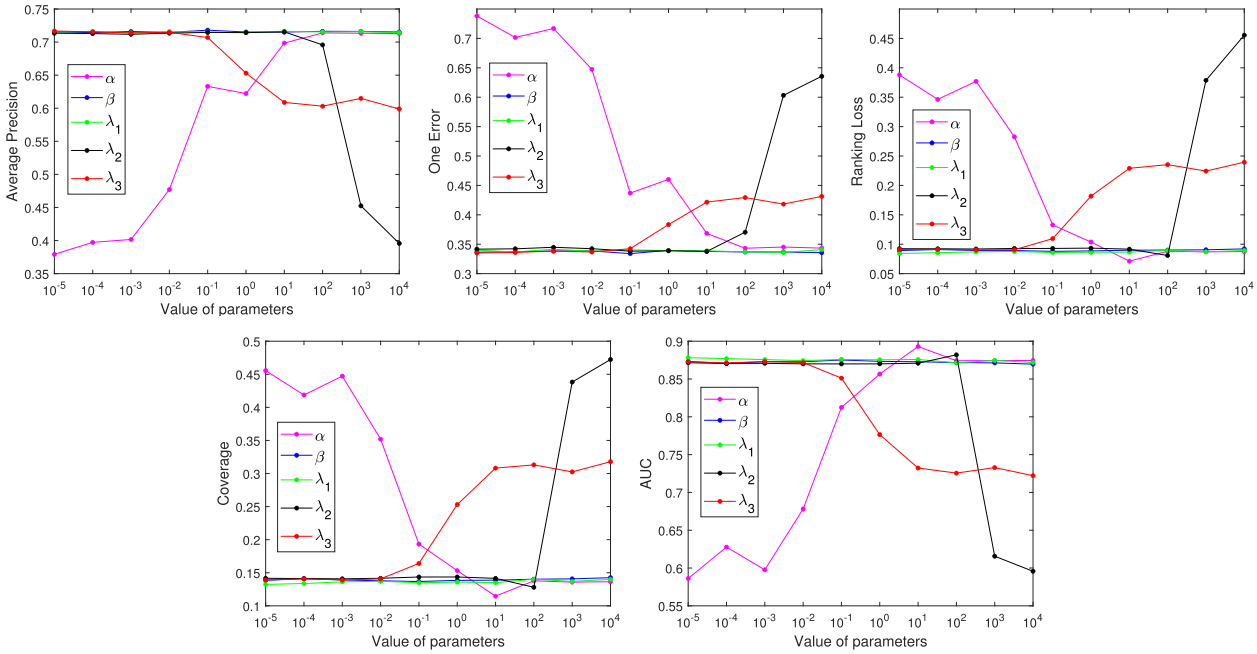


FIGURE 8. Sensitivity analysis of parameters on computers dataset while missing rate 0.3.

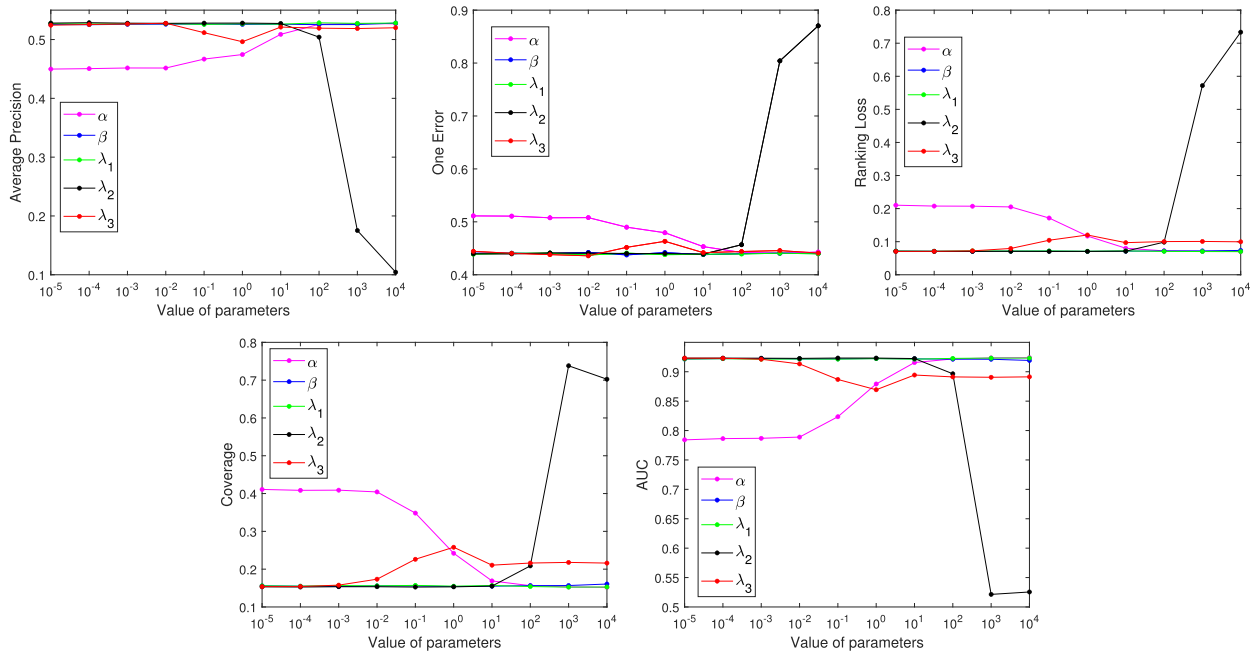


FIGURE 9. Sensitivity analysis of parameters on Stackex_cs dataset while missing rate 0.1.

label matrix and the original incomplete label matrix with parameter α. β governs the sparsity of label correlations between different class labels. The weights of label-specific features and common features components in the coefficient matrix are represented by λ_1 and λ_2, respectively. λ_3 regulates the contributions of label correlations, and all parameter values are specified from $[10^{-5}, 10^{-4}, \ldots, 10^{2}, 10^{3}]$. Next, we conduct the sensitivity analysis of CLSML parameters on three datasets with varying label missing rates. We tune one parameter by a step 10^1, while keeping other parameters at their optimal settings.

The experimental results shown in Figs. 7–9 demonstrate that the parameters α, λ_2, and λ_3 have the greatest influence on CLSML performance. Among these, α and λ_2 have a considerable impact on model performance, however, the impact of λ_3 on the model is primarily dependent

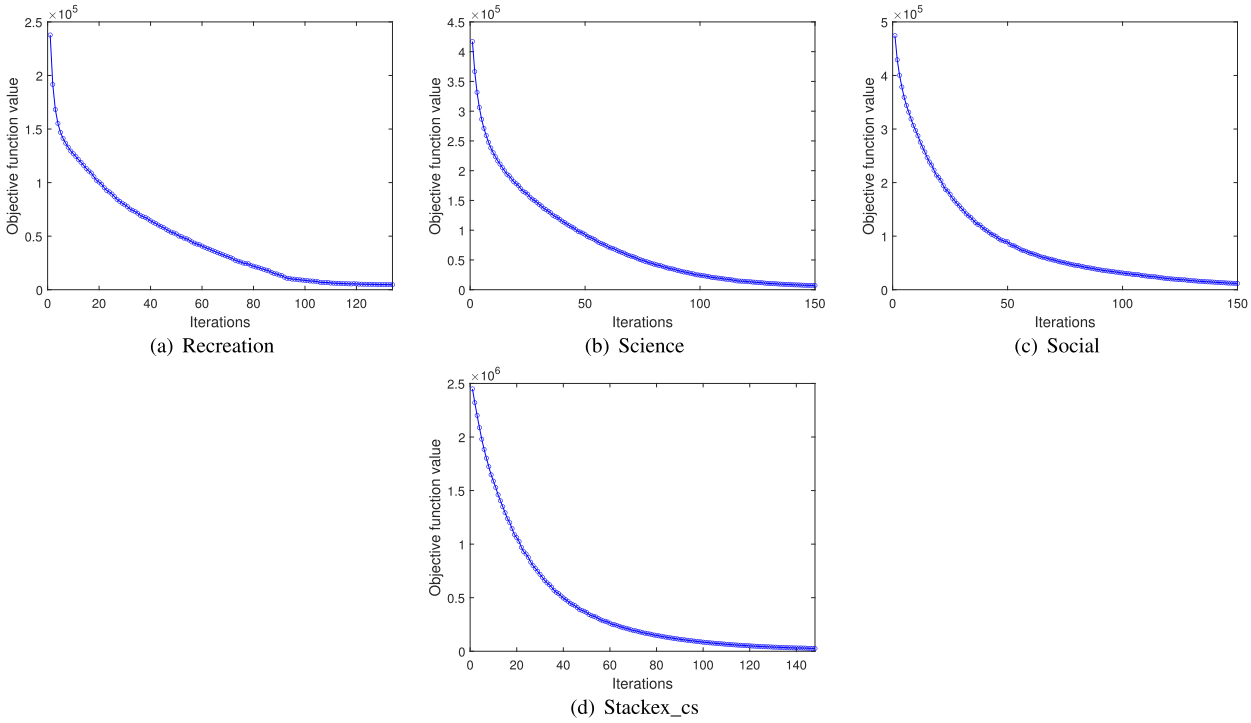


FIGURE 10. CLSML's objective function values of CLSML with respect to the number of iterations over four datasets.

on datasets. The best results are usually obtained when α is more than approximately 10^2 and λ_2 is less than about 10^0.

Obviously, decreasing the value of α or increasing the value of λ_2 reduces the algorithm's performance. This is due to the fact that if the value of λ_2 is large enough, the $l_{2,1}$-norm regularization term dominates in the model, increasing the row sparsity of the coefficient matrix, causing the coefficient matrix to come close to the zero matrix and so losing classification performance. If the value of α is small enough, it indicates that the higher-order label correlation matrix obtained through learning is unable to successfully rebuild the label matrix, resulting in poor performance. The other two parameters, on the other hand, are quite robust to the model's performance over the range of values $[10^{-5}, 10^{-4}, \ldots, 10^2, 10^4]$.

G. TIME COMPLEXITY ANALYSIS

Our proposed algorithm's time complexity is related to two primary components: initialization and iterations. The time-consuming complexity of initializing W during the initialization phase is $O(nd^2 + d^3 + ndl + d^2l)$, where n is the number of instances, d is the number of features, and l is the number of class labels. Calculating label correlations by cosine similarity takes $O(nl^2)$. Furthermore, the time-dependent complexity of computing L_f is $O(d^3 + l^3)$. The time cost is dominated during the iteration phase by calculating the gradient of $f(W)$ and $f(C)$, which can be expressed as $O(d^2n + ndl + d^2l + l^2d)$ and $O(nl^2 + l^3 + ndl + dl^2)$. CLSML's overall time complexity is thus $O((1+t)(l^3 + d^2l + nd^2 + ndl) + (n+d)tl^2 + d^3)$.

Fig. 10 shows that the objective function values on the four datasets decline rapidly as the number of iterations increases. On the Recreation, Science, Social, and Stackex_cs datasets, the changes in the objective function values stabilize after 120, 150, 160, and 150 iterations, respectively.

V. CONCLUSION

The multi-label classification matrix's columns and rows determine the specific and common feature weights corresponding to the class labels, while the l_1-norm and $l_{2,1}$-norm constrain the global sparsity and row sparsity of the matrix, respectively, determining their important roles in multi-label feature selection and extraction techniques. In this paper, we first use the l_1-norm and $l_{2,1}$-norm regular terms to constrain the classification weight matrix. We then incorporate the higher-order label correlation matrix learning constraints into the framework of the constructed classifier training model to learn the label complement matrix through the training process. Furthermore, we develop a label-completion strategy to estimate missing labels and build a second-order label correlation matrix with cosine similarity to further constrain the label outputs, resulting in two label correlation constraints embedded in the model. This effectively improves and corrects the inconsistency in the feature-label space caused by a single label correlation constraint used in most current multi-label learning methods with missing labels.

Experiments are carried out on fourteen benchmark datasets to compare our method to ten well-established multi-label classification algorithms, and the results show that for different missing label rates, our proposed method outperforms the other comparative methods in terms of overall performance, with the exception of Hamming Loss. When the missing label rate for Hamming Loss is between 0.3 and 0.6, the GLOCAL approach has a large advantage, which can be explained by the fact that Glocal takes into account both the global and local label correlations. This also drives us to conduct further research in which we will investigate local label correlation and instance correlation in order to improve the model's performance on example-based metrics.

REFERENCES

[1] N. Ueda and K. Saito, "Parametric mixture models for multi-labeled text," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Jan. 2002, pp. 737–744.

[2] I. Chalkidis, E. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos, "Large-scale multi-label text classification on EU legislation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 6314–6322, doi: 10.18653/v1/p19-1636.

[3] 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, doi: 10.1016/j.patcog.2004.03.009.

[4] J. Lee, W. Seo, J.-H. Park, and D.-W. Kim, "Compact feature subset-based multi-label music categorization for mobile devices," *Multimedia Tools Appl.*, vol. 78, no. 4, pp. 4869–4883, Feb. 2019, doi: 10.1007/s11042-018-6100-8.

[5] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, Apr. 2006, doi: 10.1093/bioinformatics/btk048.

[6] X. Kong, Z. Wu, L.-J. Li, R. Zhang, P. S. Yu, H. Wu, and W. Fan, "Large-scale multi-label learning with incomplete label assignments," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2014, pp. 920–928.

[7] B. Wu, Z. Liu, S. Wang, B.-G. Hu, and Q. Ji, "Multi-label learning with missing labels," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 1964–1968.

[8] J. Huang, F. Qin, X. Zheng, Z. Cheng, Z. Yuan, W. Zhang, and Q. Huang, "Improving multi-label classification with missing labels by learning label-specific features," *Inf. Sci.*, vol. 492, pp. 124–146, Aug. 2019, doi: 10.1016/j.ins.2019.04.021.

[9] L. Jiang, G. Yu, M. Guo, and J. Wang, "Feature selection with missing labels based on label compression and local feature correlation," *Neurocomputing*, vol. 395, pp. 95–106, Jun. 2020, doi: 10.1016/j.neucom.2019.12.059.

[10] R. Li, X. Zhao, Z. Shang, and L. Jia, "Semi-supervised multi-label learning with missing labels by exploiting feature-label correlations," *Stat. Anal. Data Mining, ASA Data Sci. J.*, vol. 16, no. 2, pp. 187–209, Dec. 2022, doi: 10.1002/sam.11607.

[11] J. Li, P. Li, X. Hu, and K. Yu, "Learning common and label-specific features for multi-label classification with correlation information," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108259, doi: 10.1016/j.patcog.2021.108259.

[12] T. Ren, X. Jia, W. Li, L. Chen, and Z. Li, "Label distribution learning with label-specific features," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 10–16.

[13] 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, doi: 10.1109/TPAMI.2014.2339815.

[14] S. Xu, X. Yang, H. Yu, D.-J. Yu, J. Yang, and E. C. C. Tsang, "Multi-label learning with label-specific feature reduction," *Knowl.-Based Syst.*, vol. 104, pp. 52–61, Jul. 2016, doi: 10.1016/j.knosys.2016.04.012.

[15] J.-J. Zhang, M. Fang, and X. Li, "Multi-label learning with discriminative features for each label," *Neurocomputing*, vol. 154, pp. 305–316, Apr. 2015, doi: 10.1016/j.neucom.2014.11.062.

[16] W. Zhan and M.-L. Zhang, "Multi-label learning with label-specific features via clustering ensemble," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2017, pp. 129–136.

[17] 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, doi: 10.1109/TCYB.2017.2663838.

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

[19] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label-specific features and class-dependent labels for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3309–3323, Dec. 2016, doi: 10.1109/TKDE.2016.2608339.

[20] L. Hu, Y. Li, W. Gao, P. Zhang, and J. Hu, "Multi-label feature selection with shared common mode," *Pattern Recognit.*, vol. 104, Aug. 2020, Art. no. 107344, doi: 10.1016/j.patcog.2020.107344.

[21] L. Jian, J. Li, K. Shu, and H. Liu, "Multi-label informed feature selection," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1627–1633.

[22] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, vol. 2, 2013, pp. 2301–2309.

[23] B. Wu, S. Lyu, and B. Ghanem, "ML-MG: Multi-label learning with missing labels using a mixed graph," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4157–4165.

[24] M. Chen, A. Zheng, and K. Weinberger, "Fast image tagging," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1274–1282.

[25] Q. Tan, Y. Yu, G. Yu, and J. Wang, "Semi-supervised multi-label classification using incomplete label information," *Neurocomputing*, vol. 260, pp. 192–202, Oct. 2017, doi: 10.1016/j.neucom.2017.04.033.

[26] Y. Y. Sun, Y. Zhang, and Z. H. Zhou, "Multi-label learning with weak label," in *Proc. 24th AAAI Conf. Artif. Intell.*, Atlanta, GA, USA, Jul. 2010, pp. 593–598.

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

[28] Y. Zhu, J. T. Kwok, and Z.-H. Zhou, "Multi-label learning with global and local label correlation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1081–1094, Jun. 2018, doi: 10.1109/TKDE.2017.2785795.

[29] L. Xu, Z. Wang, Z. Shen, Y. Wang, and E. Chen, "Learning low-rank label correlations for multi-label classification with missing labels," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 1067–1072.

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

[31] 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, doi: 10.1109/TCYB.2019.2894985.

[32] J. Ma, Z. Tian, H. Zhang, and T. W. S. Chow, "Multi-label low-dimensional embedding with missing labels," *Knowl.-Based Syst.*, vol. 137, pp. 65–82, Dec. 2017, doi: 10.1016/j.knosys.2017.09.005.

[33] R. Rastogi and S. Mortaza, "Multi-label classification with missing labels using label correlation and robust structural learning," *Knowl.-Based Syst.*, vol. 229, Oct. 2021, Art. no. 107336, doi: 10.1016/j.knosys.2021.107336.

[34] R. Rastogi and S. Kumar, "Discriminatory label-specific weights for multi-label learning with missing labels," *Neural Process. Lett.*, vol. 55, no. 2, pp. 1397–1431, Apr. 2023.

[35] S. Kumar and R. Rastogi, "Low rank label subspace transformation for multi-label learning with missing labels," *Inf. Sci.*, vol. 596, pp. 53–72, Jun. 2022, doi: 10.1016/j.ins.2022.03.015.

[36] S. Kumar, N. Ahmadi, and R. Rastogi, "Multi-label learning with missing labels using sparse global structure for label-specific features," *Int. J. Speech Technol.*, vol. 53, no. 15, pp. 18155–18170, Jan. 2023, doi: 10.1007/s10489-022-04439-7.

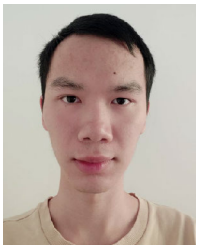
[37] F. Nie, H. Huang, C. Xiao, and C. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. NIPS*, 2010, pp. 1813–1821.

[38] Y. Wang, W. Zheng, Y. Cheng, and D. Zhao, "Joint label completion and label-specific features for multi-label learning algorithm," *Soft Comput.*, vol. 24, no. 9, pp. 6553–6569, May 2020, doi: 10.1007/s00500-020-04775-1.

[39] H. Han, M. Huang, Y. Zhang, X. Yang, and W. Feng, "Multi-label learning with label specific features using correlation information," *IEEE Access*, vol. 7, pp. 11474–11484, 2019, doi: 10.1109/ACCESS.2019.2891611.

[40] Y. Cheng, C. Zhang, and S. Pang, "Multi-label space reshape for semantic-rich label-specific features learning," *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 4, pp. 1005–1019, Apr. 2022, doi: 10.1007/s13042-021-01432-3.
[41] K. M. Deliparaschos, F. I. Nenedakis, and S. G. Tzafestas, "Design and implementation of a fast digital fuzzy logic controller using FPGA technology," *J. Intell. Robotic Syst.*, vol. 45, no. 1, pp. 77–96, Jan. 2006, doi: 10.1007/s10846-005-9016-2.
[42] 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, doi: 10.1016/j.patcog.2006.12.019.
[43] Y. Zhang and Z.-H. Zhou, "Multilabel dimensionality reduction via dependence maximization," *ACM Trans. Knowl. Discovery from Data*, vol. 4, no. 3, pp. 1–21, Oct. 2010, doi: 10.1145/1839490.1839495.
[44] 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, doi: 10.1109/TKDE.2013.39.

RUNXIN LI received the B.Sc. and Ph.D. degrees in applied mathematics from Yunnan University, China, in 2010 and 2013, respectively. He is currently a Lecturer with Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming, China. His main research interests include data mining and machine learning.



ZEXIAN OUYANG received the bachelor's degree in electronic information science and technology from Hunan University of Arts and Science, Changde, Hunan, in 2020. He is currently pursuing the M.S. degree in computer technology with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan, China. His research interests include data mining and machine learning.

ZHENHONG SHANG received the B.Sc. degree in electronic science and engineering from Nankai University, Tianjing, China, in 1998, the M.Sc. degree in computer science from Kunming University of Science and Technology, Kunming, China, in 2001, and the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004. He is currently a Professor with the Department of Computer Science and Technology, Kunming University of Science and Technology. His current research interests include computer vision and image processing.

LIANYIN JIA received the Ph.D. degree from the School of Computer Science, South China University of Technology, in 2012. He is currently a Professor with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. His current research interests include database, data mining, information retrieval, and parallel computing.

XIAOWU LI received the M.S. degree in computer application technology from Kunming University of Science and Technology, Kunming, China, in 2007, and the Ph.D. degree in computer application technology from Southwest Jiaotong University, Chengdu, China, in 2015. He is currently a Lecturer with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology. His research interests include intelligent systems, edge computing, artificial intelligence, identification technology of RFID tags, and location systems based on RFID and Lora technology.