Soft Computing (2022) 26:2225–2239
https://doi.org/10.1007/s00500-021-06645-w

DATA ANALYTICS AND MACHINE LEARNING

Learning multi-label label-specific features via global and local label correlations

Dawei Zhao[1,2] · Qingwei Gao[1,2] · Yixiang Lu[1] · Dong Sun[1]

Accepted: 3 December 2021 / Published online: 27 January 2022

Abstract
Label-specific features learning is a multi-label learning framework that utilizes label feature extraction to solve a single example where multiple class labels exist simultaneously. As an essential multi-label learning method, label correlation learning has been widely used in multi-label classification learning. However, in the existing label-specific features learning, the label correlation measurement often assumes that the label correlations are a global structure or that the label correlations only have a local smoothness. In actual application scenarios, the two situations may occur together. This paper proposes a multi-label classification method by joint **L**abel-**S**pecific features and **G**lobal and **L**ocal label correlation learning, named LSGL. Firstly, we obtain the weight of the label-specific features of each class label utilizing the l_1-norm and then learn high-order global label correlation and label local smoothness. By adding manifold regularization terms, we fully utilize the structural relationship between features and labels and mine global and local label association information. These processes are carried out in a unified optimization model, and each part learns and promotes each other. Finally, in the low-dimensional label-specific features representation learning is to carry out multi-label classification learning through the support vector machine and the extreme learning machine, respectively. A comparative study with state-of-the-art approaches and statistical hypothesis testing manifests the validity of the LSGL method and the features learned from label-specific features learning.

Keywords Multi-label label-specific features learning · Label correlations · Proximal gradient descent · Support vector machine · Extreme learning machine

1 Introduction

Multi-label learning (MLL), as one of the hotspots in machine learning research, has been widely used in many domains, for example, image annotation, text classification, and gene annotation (Wang et al. 2016; Zhang et al. 2018; Liu et al. 2018) (Al-Salemi et al. 2018; Gargiulo et al. 2019) (Guan et al. (2018)), respectively. In a framework of MLL, a single instance is associated with multiple class labels simultaneously, and its main challenge is how to learn an efficient classification model that predicts a set of labels that may exist for a new instance (Gibaja and Ventura 2015; Zhang and Zhou 2013). Existing MLL approaches can be divided into problem transformation (PT) and algorithm adaptive (AA) approaches Tsoumakas et al. (2009). The main idea of the problem transformation approach is to convert one or more single-label classification learning algorithms into MLL approaches. Typical examples include the binary classification method (BR) Boutell et al. (2004) and the chain classification method (CC) Read et al. (2011). The adaptive algorithm method improves the traditional single-label classification algorithm to realize the classification of MLL directly. Algorithm adaptation is currently the primary way to solve MLL problems. Representative algorithms include the lazy learning algorithm ML-kNN Zhang and Zhou (2007) and the kernel tricks learning algorithm RankSVM Elisseeff and Weston (2002). RMLDM Rezaei-Ravari et al. (2021) leverages dual-manifold regularization to construct a neural network and simultaneously combines feature and label local geometric structure mining for multi-label learning. In the past research, MLL has made progress, but some problems still need further study. Learning and utilizing the correlation among labels are one of the critical issues currently recog-

✉ Qingwei Gao
qingweigao@ahu.edu.cn

[1] School of Electrical Engineering and Automation, Anhui University, Hefei 230601, People's Republic of China

[2] School of Computer Science and Technology, Anhui University, Hefei 230601, People's Republic of China

nized and concerned. The label correlation (LC) theory holds that there is a specific correlation among labels Zhang et al. (2019). For example, in a picture with the sky and sea labels, in the MLL, it can be considered that the sky label may be associated with the white cloud label and the sea and fish labels are relatively large, and the correlation information among labels can improve the performance of the classifier.

At present, there are a large number of methods to explore the feasibility of using LC to improve the performance of MLL, which is based on a probability model or an optimization model to solve the problem. The existing MLL methods can be systematically divided into the following three categories according to the LC considered Zhang and Zhou (2013). The strategy of the first-order approach is to solve MLL problems without considering LC, such as BPMLL Zhang and Zhou (2006), CLR Fürnkranz et al. (2008), and MLRL Zhang and Yeung (2013); the second-order strategy approach considers that the correlations among the labels appear in pairs, such as Huang et al. (2016); Weng et al. (2018); the higher-order strategy approach considers the correlations among all class labels or a subset of class labels, such as Charte et al. (2014), Cheng et al. (2018), Jun Xie et al. (2019), He et al. (2019), Xu et al. (2014). These studies show that LC can effectively improve the performance of multi-label classification algorithms.

However, most of the existing MLL methods on the hypothesis of LC are considered global label correlation. In other words, they utilize globally consistent LC during the learning process. In the real world, the differences among instances can lead to different correlations among labels. In other words, due to the similarity among instances, the correlation among different labels may lead to local LC. ML-LOC Huang and Zhou (2012) first proposed the concept of local LC when solving MLL problems, which believed that there would be a local correlation among labels due to different instances. Therefore, a large number of scholars have considered the global and local LC together.

To this end, this paper proposes a label-specific features algorithm that combines global and local LC (LSGL). The main contributions of this paper are as follows:

1. Consider learning label-specific features and global and local label correlations under a unified framework, which promotes and influences each other.
2. Through a reliable hypothesis, there is often a specific relationship between the local characteristics of the examples and the local LC, and this connection is more evident in learning label-specific features.
3. The experimental results on 15 benchmark multi-label data sets verify that our algorithm is more competitive than the state-of-the-art algorithms and converges with fewer iterations.

The remainder of this paper is organized as follows: In Sect. 2, we reviewed the related work. In Sect. 3, we introduce the specific framework of algorithm learning proposed in this paper. In Sect. 4, we introduce the optimization process of the algorithm. The results of comparative experiments and specific analyses are illustrated in Sect. 5. Finally, Sect. 6 concludes this paper.

2 Related work

2.1 Label-specific features learning

Label-specific features learning is a novel multi-label classification direction in only a few years, which assumes each label has its unique feature subset representation. LIFT Zhang and Lei (2014) solves the multi-label classification problem from the perspective that each class label has its unique attributes. For example, we utilize the color and texture features to distinguish the blue sky and white clouds in pictures. LIFT performs cluster analysis on the specific feature' positive and negative instances of the labels and utilizes the distance between the original instance and the positive instance and the counterexample center instance to represent the new feature. Based on LIFT, many scholars have made improvements. For example, FRS-LIFT Suping et al. (2016) utilizes fuzzy rough sets to reduce the label-specific features further. FRS-SS-LIFT Suping et al. (2016) reduces the label-specific features based on sample selection. The above algorithms do not consider the label correlations information. MLC-LFLC Ma et al. (2021) extracts two-level label-specific features in a unified model framework. MLC-LFLC constructs a classifier with a certain distinguishability between instances with the same label and a certain degree of connectivity between instances with different labels. LF-LPLC Weng et al. (2018) has developed MLL based on label-specific features and local pairwise label correlation. SLEF Qiao et al. (2017) finds second-order label correlations and adds sparse regular learning label-specific features to the model parameters. MLSF Sun et al. (2016) combines meta-label learning and label-specific feature learning through a two-step learning method. The first step is tantamount to constructing a meta-label space by using spectral clustering for accounting label correlations. The second step uses the l_1-norm to learn label-specific features. LLSF Huang et al. (2015) has learned the feature extraction method based on label-specific features learning, but its label correlation is pre-computed. LLSF can be regarded as a method of feature space extraction. MLFC Zhang et al. (2018) jointly learns label correlations and label-specific features, which considers that similar samples share similar label correlation-based features concerning the label space, where additional features represent the label correlations. LSML Huang et al. (2019)

jointly learns the high-order label correlation and specific-label features of a unified system to solve the problem of missing labels in MLL. The method of this paper is the same as LLSF and LSML in terms of learning label-specific features, but LLSF considers label correlation by directly adding prior knowledge, while LSML only explores the global label correlation of label-specific features. The research on the above-mentioned label-specific features learning found that most of the existing methods ignore the problem of local label structure.

2.2 Label correlations learning

On the aforementioned, there is equally the problem of global and local LC in label-specific features learning. The local smoothness of labels assumes that instances close to each other usually share the same set of label subsets. MDFS Zhang et al. (2019) performs low-dimensional embedding of the original feature space to fit the natural label distribution and captures the local LC information. The motivation for constructing global LC is to observe that the correlation among labels is usually shared among instances Huang and Zhou (2012). At present, some algorithms have constructed global and local LC into a unified learning framework for MLL. For example, in GLOCAL Zhu et al. (2017), global and local label correlations are implemented to solve the full and missing multi-label classification problems. GLMVML Zhu et al. (2020) utilizes global and local label correlations of the entire data set and each view at the same time when dealing with multi-view and MLL problems. CLSF Che et al. (2020) assigns labels with strong relationships to the corresponding label group and computes local and global label correlations simultaneously. According to the LC correlations theory, if the labels of the instances are similar, then the spatial structure of the instances is likely that similar instances may share some instance-related information, which is often reflected in the local space among instances. Similarly, local label correlations assume that the label correlations between different instances are different, but instances with similar features share their common label relationships.

The research, as mentioned above, has proved theoretically or practically that the performance of multi-label classifiers can be substantially improved by using global and local LC. The existing methods of mining global and local LC mainly use the entire feature representation data to distinguish different labels. According to the label-specific feature research, it can be found that this strategy may obtain sub-optimal results. This paper proposes a multi-label classification method that utilizes global and local label correlation with label-specific features learning. First, we design a linear optimization framework to model label-specific feature learning problems through global and local LC. In this framework, the label-specific feature weights and LC weight coefficients are updated in alternating iterations. We build the non-negligible difference relationship between the original label set and the ground truth based on label propagation dependence and explore the global and local correlation structure of labels based on the feature-label and sample-label manifold regularization. Secondly, these optimization features can be utilized to distinguish the corresponding label-specific features from other labels. Finally, we utilize a linear model to predict the set of labels for unknown instances. At the same time, the LSGL algorithm is invoked as a feature extraction method combined with ELM Huang et al. (2015), Huang (2014), Cheng et al. (2019) and BSVM models. By analyzing experimental results and statistical hypothesis tests, we can draw intuitive conclusions that LSGL has achieved a certain degree of competitiveness compared to the state-of-the-art multi-label method on various evaluation metrics.

3 The proposed method

3.1 Multi-label learning

Let $\boldsymbol{X} = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{n\times d}$ be the input instance space of d-dimensional features, where n denotes the number of samples in the input space, $x_i \in \mathbb{R}^d$ denotes the feature vector corresponding to the i-th instances. $\boldsymbol{Y} = [y_1, y_2, \cdots, y_n]^{\mathrm{T}} \in \mathbb{R}^{n\times l}$ denotes the label space matrix corresponding to the instance space, where l denotes the number of labels of the instance; $y_i \in \{0, 1\}^i$ denotes the corresponding label vector. Therefore, an MLL training data set containing n instances can be defined as:

$$\boldsymbol{D} = \{x_i, Y_i | 1 \le i \le n\} \subset \mathbb{R}^d \times \{0, 1\}^i$$

3.2 Label-specific features learning

In general, regarding label-specific features, we assume that a known instance space gives the own attributes of each class label. In dealing with this problem, a linear model with l_1-norm regularization can generally be utilized for modeling Huang et al. (2015). Nonzero entries for each $\boldsymbol{w}^i \in \mathbb{R}^d$ can be utilized to determine the specific characteristics of the label and can also effectively distinguish the corresponding class labels.

$$\min_{\boldsymbol{w}^i} \frac{1}{2} \left\| \boldsymbol{X}\boldsymbol{w}^i - \boldsymbol{y}^i \right\|_2^2 + \lambda_4 \left\| \boldsymbol{w}^i \right\|_1 \tag{1}$$

Further, the optimization problem can be rewritten as:

$$\min_{\boldsymbol{W}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y}\|_F^2 + \lambda_4 \|\boldsymbol{W}\|_1 \tag{2}$$

where $\boldsymbol{W} = \left[\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^l\right] \in \mathbb{R}^{d\times l}$ denotes the weight parameter, and $\lambda_4 \geq 0$ denotes a penalty parameter. Intuitively, samples with similar labels have similar similarities in their features. Assuming that there are highly similar label-specific features, their corresponding class labels $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are also strongly correlated. Therefore, there is a similarity between the corresponding label-specific feature model parameters $\boldsymbol{w}^i$ and $\boldsymbol{w}^j$. Otherwise, $\boldsymbol{w}^i$ and $\boldsymbol{w}^j$ are different. This paper utilizes a common Euclidean distance metric to measure the similarity between $\boldsymbol{w}^i$ and $\boldsymbol{w}^j$. Then, the optimization problem for Eq.2 can be rewritten as:

$$\begin{aligned} &\min_{\boldsymbol{W},\boldsymbol{S}} \frac{1}{2}\left\|\boldsymbol{X}\boldsymbol{W}-\boldsymbol{Y}\right\|_F^2 + \frac{\lambda_2}{2}\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\right) + \lambda_4\|\boldsymbol{W}\|_1 \\ &\text{s.t.}\ \ \boldsymbol{S} \geq 0 \end{aligned} \tag{3}$$

where $\boldsymbol{S} \in \mathbb{R}^{l\times l}$ denotes the label correlation matrix, $\boldsymbol{S}_{ij}$ denotes the degree of correlation between labels $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$. Because the degree of correlation between labels is different from each other, $\boldsymbol{S}$ is a matrix that is not strictly symmetric. To avoid that the matrix $\boldsymbol{S}$ is not positive definite or semi-positive definite, which adversely affects the second regular term in Eq.3, we give $\boldsymbol{S} = \frac{\boldsymbol{S}+\boldsymbol{S}^{\mathrm{T}}}{2}$ to solve this problem.

3.3 Global and Local label correlations learning

Inspired by the idea of label propagation dependence Fu et al. (2013), Xu Xu et al. (2014) et al. consider that there is an individual dependency among labels, and it is found that $\boldsymbol{Y}\times\boldsymbol{S}$ can be utilized to complement the original labels space, where $\boldsymbol{S}$ is a global label correlation matrix. The original label space information can be propagated. To ensure the addition of related regular terms to the difference between the label completion matrix and the original label matrix, Eq.3 can be written as:

$$\begin{aligned} &\min_{\boldsymbol{W},\boldsymbol{S}} \frac{1}{2}\left\|\boldsymbol{X}\boldsymbol{W}-\boldsymbol{Y}\right\|_F^2 + \frac{\lambda_1}{2}\left\|\boldsymbol{Y}\boldsymbol{S}-\boldsymbol{Y}\right\|_F^2 \\ &\quad + \frac{\lambda_2}{2}\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\right) + \lambda_4\|\boldsymbol{W}\|_1 \\ &\text{s.t.}\ \ \boldsymbol{S} \geq 0 \end{aligned} \tag{4}$$

We assume that the LC matrix can enrich the original label space information to improve the classification performance, but to avoid the unreliable measurement of label correlation affecting the effect of multi-label learning, the second term in Eq.4 is defined to constrain the label correlation.

Further, based on the exploration of label correlation from a global perspective, we consider the local structure information of the label to ensure that the resulting label correlation matrix is more robust than the "ground truth". The smoothness hypothesis is usually used when exploring the local label correlation. It is believed that the distance between any two examples in the feature space can measure the similarity of their corresponding class labels. Furthermore, it can be expressed that if the initial label vectors $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are very similar in inherent geometric space, then the real labels $\widehat{\boldsymbol{Y}}_i$ and $\widehat{\boldsymbol{Y}}_j$ should also have similar structural features Zhang et al. (2019), Ren et al. (2017). The manifold regularization term of the local smooth structure hypothesis can be expressed as:

$$\begin{aligned} \Omega\left(\boldsymbol{S}\right) &= \sum_{i,j=1}^{n} \left\|\boldsymbol{Y}\boldsymbol{S}_i - \boldsymbol{Y}\boldsymbol{S}_j\right\|^2 \boldsymbol{E}_{i,j} \\ &= \mathrm{Tr}\left((\boldsymbol{Y}\boldsymbol{S})^{\mathrm{T}}\boldsymbol{L}_x\boldsymbol{Y}\boldsymbol{S}\right) \end{aligned} \tag{5}$$

where $\boldsymbol{L}_x$ is the graph Laplacian matrix of $\boldsymbol{E}$. $\boldsymbol{E}$ is the weight matrix of the instance, used for the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ of the instance. $\boldsymbol{E}$ can be obtained as follows:

$$\boldsymbol{E}_{ij} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{2\sigma^2}\right) & \boldsymbol{x}_i \in N_k\left(\boldsymbol{x}_j\right)\ \ or\ \ \boldsymbol{x}_j \in N_k\left(\boldsymbol{x}_i\right) \\ 0 & otherwise \end{cases} \tag{6}$$

The LC matrix $\boldsymbol{S}$ preserves the local smoothness of the sample. Note that the correlation among class labels may only be related to a subset of the class label, so we add the l_1-norm regular item on $\boldsymbol{S}$ to learn the sparse label dependency; then, the final optimization function is expressed as follows:

$$\begin{aligned} &\min_{\boldsymbol{W},\boldsymbol{S}} \frac{1}{2}\left\|\boldsymbol{X}\boldsymbol{W}-\boldsymbol{Y}\right\|_F^2 \\ &\quad + \frac{\lambda_1}{2}\left\|\boldsymbol{Y}\boldsymbol{S}-\boldsymbol{Y}\right\|_F^2 + \frac{\lambda_2}{2}\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\right) \\ &\quad + \frac{\lambda_3}{2}\Omega\left(\boldsymbol{S}\right) + \lambda_4\|\boldsymbol{W}\|_1 + \lambda_5\|\boldsymbol{S}\|_1 \\ &\text{s.t.}\ \ \boldsymbol{S} \geq 0 \end{aligned} \tag{7}$$

4 Optimization

Equation7 is a convex optimization problem, but it is not smooth due to the existence of the l_1-norm regularization term. In convex optimization problems, there are some cases where the objective function is not differentiable. The general solution is to solve the optimal solution by introducing subgradients iteratively. However, the speed of the subgradient method is slower than that of the gradient descent method. For this reason, for some cases where the overall non-differentiable but the non-smooth convex function can be decomposed into differentiable and non-differentiable, the approximation model can be utilized to optimize the solution. This method of solving the non-smooth convex optimization problem is named accelerated proximal gradient descent

method (PGD) Combettes and Wajs (2005). Specifically, two model parameters (i.e., $\boldsymbol{W}$ and $\boldsymbol{S}$) in Eq.7 are represented by $\boldsymbol{\Phi}$. According to the representation of the convex optimization problem in the general PGD algorithm, we abbreviate Eq.7 as follows:

$$\min_{\boldsymbol{\Phi}\in\mathcal{H}} \{F(\boldsymbol{\Phi}) := f(\boldsymbol{\Phi}) + g(\boldsymbol{\Phi})\} \tag{8}$$

where $\mathcal{H}$ is a real Hilbert space, $f(\cdot)$ is a convex differentiable function, and $g(\cdot)$ is a convex non-differentiable function. $f(\boldsymbol{\Phi})$ and $g(\boldsymbol{\Phi})$ are expressed as:

$$\begin{aligned} f(\boldsymbol{\Phi}) &= \frac{1}{2}\|\boldsymbol{X}\boldsymbol{W}-\boldsymbol{Y}\|_F^2 + \frac{\lambda_1}{2}\|\boldsymbol{Y}\boldsymbol{S}-\boldsymbol{Y}\|_F^2 \\ &+\frac{\lambda_2}{2}\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\right) + \frac{\lambda_3}{2}\Omega(\boldsymbol{S}) \\ \text{s.t. } & \boldsymbol{S}\geq 0 \end{aligned} \tag{9}$$

$$g(\boldsymbol{\Phi}) = \lambda_4\|\boldsymbol{W}\|_1 + \lambda_5\|\boldsymbol{S}\|_1 \tag{10}$$

For any $L > 0$, we can perform a second-order approximation to the function $f(\cdot)$. The second approximation of $F(\boldsymbol{\Phi}) := f(\boldsymbol{\Phi}) + g(\boldsymbol{\Phi})$ is defined as follows:

$$\begin{aligned} \mathcal{Q}_L\left(\boldsymbol{\Phi},\boldsymbol{\Phi}^t\right) &= f(\boldsymbol{\Phi}) + \left\langle \nabla f\left(\boldsymbol{\Phi}^t\right), \boldsymbol{\Phi}-\boldsymbol{\Phi}^t\right\rangle \\ &+\tfrac{1}{2}\left\|\boldsymbol{\Phi}-\boldsymbol{\Phi}^t\right\|_F^2 + g(\boldsymbol{\Phi}) \end{aligned} \tag{11}$$

For any $L \geq L_f$, where L_f denotes the Lipschitz constant, here it has $\mathcal{Q}_L\left(\boldsymbol{\Phi},\boldsymbol{\Phi}^t\right) \geq F(\boldsymbol{\Phi})$. Afterward, the PGD algorithm minimizes a sequence of separable quadratic approximations to $F(\boldsymbol{\Phi})$. Finally, the solution of $\boldsymbol{\Phi}$ can be obtained by minimizing $\mathcal{Q}_L\left(\boldsymbol{\Phi},\boldsymbol{\Phi}^t\right)$:

$$\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}} g(\boldsymbol{\Phi}) + \frac{L}{2}\left\|\boldsymbol{\Phi}-\boldsymbol{G}^t\right\|_F^2 \tag{12}$$

where $\boldsymbol{G}^t = \boldsymbol{\Phi}^t - \frac{1}{L}\nabla f\left(\boldsymbol{\Phi}^t\right)$, $\boldsymbol{\Phi}^t = \boldsymbol{\Phi}_t + \frac{\alpha_{t-1}-1}{\alpha_t}\left(\boldsymbol{\Phi}_t - \boldsymbol{\Phi}_{t-1}\right)$, and $\boldsymbol{\Phi}^t = \boldsymbol{\Phi}_t + \frac{\alpha_{t-1}-1}{\alpha_t}\left(\boldsymbol{\Phi}_t - \boldsymbol{\Phi}_{t-1}\right)$ for a sequence α_t by satisfying $\alpha_{t-1}^2 - \alpha_{t+1} \leq \alpha_t^2$ can improve the convergence rate to $\mathcal{O}\left(\frac{1}{t^2}\right)$. Here $\boldsymbol{\Phi}_t$ is the result of $\boldsymbol{\Phi}$ at the t-th iteration. The model coefficients $\boldsymbol{W}$ and $\boldsymbol{S}$ are unknown parameters for the problem Eq.7, and they can be updated alternatively. In this paper, one is fixed in each iteration of the two model coefficients, and the other is updated.

4.1 Fix S updating W

The gradient of the problem Eq.9 w.r.t $\boldsymbol{W}$ can be obtained by:

$$\begin{aligned} \nabla f(\boldsymbol{W}) &= \frac{\partial f_{\boldsymbol{W}}(\boldsymbol{\Phi})}{\partial \boldsymbol{W}} \\ &= \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W} - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} + \lambda_2\boldsymbol{W}\boldsymbol{S} \end{aligned} \tag{13}$$

According to Eq.12, $\boldsymbol{W}$ can be computed as:

$$\boldsymbol{W}^t = \boldsymbol{W}_t + \frac{\alpha_{t-1}-1}{\alpha_t}\left(\boldsymbol{W}_t - \boldsymbol{W}_{t-1}\right) \tag{14}$$

$$\boldsymbol{W}^{t+1} = \mathrm{prox}_{\varepsilon}\left(\boldsymbol{W}^t - \frac{1}{L}\nabla f\left(\boldsymbol{W}^t\right)\right) \tag{15}$$

where $\uparrow$ denotes the step size. The function $g(\boldsymbol{\Phi})$ with respect to $\boldsymbol{W}$ corresponds to the l_1-norm, which can be solved by the element-wise soft-threshold operator defined as:

$$\mathrm{prox}_{\varepsilon}\left(\boldsymbol{W}_{ij}\right) = \left(\left|\boldsymbol{W}_{ij}\right| - \varepsilon\right)_+ \mathrm{sign}\left(\boldsymbol{W}_{ij}\right) \tag{16}$$

where $(\cdot)_+ = \max(\cdot, 0)$.

4.2 Fix W updating S

The gradient of the problem Eq.9 w.r.t $\boldsymbol{S}$ can be obtained by:

$$\begin{aligned} \nabla f(\boldsymbol{S}) &= \frac{\partial f_{\boldsymbol{S}}(\boldsymbol{\Phi})}{\partial \boldsymbol{S}} = \lambda_1\left(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{S} - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right) \\ &+\frac{\lambda_2}{2}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} + \lambda_3\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}_x\boldsymbol{Y}\boldsymbol{S} \end{aligned} \tag{17}$$

We utilize the parameter $\lambda_1 > 0$ to control the loss between the label completion matrix and the original label matrix and $\boldsymbol{L}_x \in \mathbb{R}^{n\times n}$ is the graph Laplacian matrix of the coder labels matrix $\boldsymbol{E}$ ($\boldsymbol{E}$ is utilized to calculate the similarity between instances). In the accelerated PGD method, $\boldsymbol{S}$ can be updated by:

$$\boldsymbol{S}^t = \boldsymbol{S}_t + \frac{\alpha_{t-1}-1}{\alpha_t}\left(\boldsymbol{S}_t - \boldsymbol{S}_{t-1}\right) \tag{18}$$

$$\boldsymbol{S}^{t+1} = \mathrm{prox}_{\varepsilon}\left(\boldsymbol{S}^t - \frac{1}{L}\nabla f\left(\boldsymbol{S}^t\right)\right) \tag{19}$$

The function $g(\boldsymbol{\Phi})$ with regard to $\boldsymbol{S}$ corresponds to the l_1-norm with the qualified non-negative constraint, which can be solved by the element-wise soft-threshold operator defined as:

$$\mathrm{prox}_{\varepsilon}\left(\boldsymbol{S}_{ij}\right) = \left(\left|\boldsymbol{S}_{ij}\right| - \varepsilon\right)_+ \mathrm{sign}\left(\boldsymbol{S}_{ij}\right) \tag{20}$$

where $(\cdot)_+ = \max(\cdot, 0)$.

In order to prove the *Lipschitz* continuity of Eq.7, given $\boldsymbol{\Phi}_1 = (\boldsymbol{W}_1, \boldsymbol{S}_1)$, and $\boldsymbol{\Phi}_2 = (\boldsymbol{W}_2, \boldsymbol{S}_2)$. According to Eq.13, Eq.17, and the *Frobenius* norm inequality, the following reasoning can be obtained:

$$\begin{aligned} &\|\nabla f(\boldsymbol{\Phi}_1) - \nabla f(\boldsymbol{\Phi}_2)\|_F^2 \\ &= \left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\Delta\boldsymbol{W} + \lambda_2\Delta\boldsymbol{W}\boldsymbol{S} + \lambda_1\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\Delta\boldsymbol{S}\right. \\ &\left.+\lambda_3\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}_x\boldsymbol{Y}\Delta\boldsymbol{S}\right\|_F^2 \end{aligned}$$

$$\leq 2\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right\|_2^2\|\Delta\boldsymbol{W}\|_F^2+2\|\Delta\boldsymbol{W}\|_F^2\|\lambda_2\boldsymbol{S}\|_2^2 +2\left\|\lambda_1\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right\|_2^2\|\Delta\boldsymbol{S}\|_F^2+2\left\|\lambda_3\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}_x\boldsymbol{Y}\right\|_2^2\|\Delta\boldsymbol{S}\|_F^2 \tag{21}$$

where $\nabla f(\boldsymbol{\Phi}_1)=\nabla f(\boldsymbol{W}_1)+\nabla f(\boldsymbol{S}_1)$, $\nabla f(\boldsymbol{\Phi}_2)=\nabla f(\boldsymbol{W}_2)+\nabla f(\boldsymbol{S}_2)$, $\Delta\boldsymbol{W}=\boldsymbol{W}_1-\boldsymbol{W}_2$, and $\Delta\boldsymbol{S}=\boldsymbol{S}_1-\boldsymbol{S}_2$.

Then, the $Lipschitz$ constant L_f is expressed as:

$$\begin{aligned}&L_f=\sqrt{2(\boldsymbol{A}+\boldsymbol{B})};\\&\boldsymbol{A}=\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right\|_2^2+\left\|\lambda_1\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right\|_2^2;\\&\boldsymbol{B}=\|\lambda_2\boldsymbol{S}\|_2^2+\left\|\lambda_3\boldsymbol{Y}^{T}\boldsymbol{L}_x\boldsymbol{Y}\right\|_2^2\end{aligned} \tag{22}$$

In summary, we introduce a linear model to generate the predicted labels vector $\boldsymbol{Y}_t$:

$$\boldsymbol{Y}_t=\operatorname{sign}(\boldsymbol{P}_t-\eta) \tag{23}$$

where $\boldsymbol{P}_t=\boldsymbol{X}_t\boldsymbol{W}^*$, η is the given threshold set to be 0.5.

Algorithm 1 **L**abel-**S**pecific features learning via **G**lobal and **L**ocal label correlations(**LSGL**).

Require:
The training data set: $\boldsymbol{X}\in\mathbb{R}^{n\times d}$;
The label data set: $\boldsymbol{Y}\in\mathbb{R}^{n\times l}$;
The non-negative trade-off parameters: λ_1,λ_2,λ_3,λ_4,λ_5;
Instance similarity matrix: $\boldsymbol{L}_x$;
Ensure:
Model parameters: $\boldsymbol{W}^*$ and $\boldsymbol{S}^*$;
1: Initialization: $\boldsymbol{W}_0$, $\boldsymbol{W}_1=rand(n,l)$; $\boldsymbol{S}_0$, $\boldsymbol{S}_1=zeros(n,l)$; α_0, $\alpha_1=1$; $t=1$.
2: **repeat**
3: $\boldsymbol{W}^t=\boldsymbol{W}_t+\frac{\alpha_{t-1}-1}{\alpha_t}(\boldsymbol{W}_t-\boldsymbol{W}_{t-1})$;
4: $\boldsymbol{G}_w^t=\boldsymbol{W}^t-\frac{1}{L_f}\nabla_{\boldsymbol{W}}f(\boldsymbol{W}^t)$;
5: $\boldsymbol{W}_{t+1}=\text{prox}_{\frac{\lambda_4}{L_f}}(\boldsymbol{G}_{\boldsymbol{w}}^t)$ by Eq.15;
6: $\boldsymbol{S}^t=\boldsymbol{S}_t+\frac{\alpha_{t-1}-1}{\alpha_t}(\boldsymbol{S}_t-\boldsymbol{S}_{t-1})$;
7: $\boldsymbol{G}_s^t=\boldsymbol{S}^t-\frac{1}{L_f}\nabla_{\boldsymbol{S}}f(\boldsymbol{S}^t)$;
8: $\boldsymbol{S}_{t+1}=\text{prox}_{\frac{\lambda_5}{L_f}}(\boldsymbol{G}_s^t)$ by Eq.19;
9: $\boldsymbol{S}_{t+1}=\boldsymbol{S}_t$;
10: Compute LC matrix $\boldsymbol{S}$ by $\boldsymbol{S}=\frac{\boldsymbol{S}+\boldsymbol{S}^{\mathrm{T}}}{2}$;
11: $\alpha_{t+1}=\frac{1+\sqrt{4\alpha_t^2+1}}{2}$;
12: $t=t+1$;
13: **until** convergence
14: **return** $\boldsymbol{W}^*=\boldsymbol{W}_t$;$\boldsymbol{S}^*=\boldsymbol{S}_t$.

4.3 Complexity analysis

In this section, we mainly analyze the complexity of the optimization parts listed in Algorithm 1. In the LSGL algorithm, $\boldsymbol{X}\in\mathbb{R}^{n\times d}$, $\boldsymbol{Y}\in(0,1)^{n\times l}$, $\boldsymbol{S}\in\mathbb{R}^{l\times l}$, and $\boldsymbol{W}\in\mathbb{R}^{d\times l}$, where n denotes the number of samples, d denotes the dimension of the sample, and l denotes the number of class labels. In Algorithm 1, most time-consuming mainly consists of three parts. Firstly, we calculate the value of functions F (**fl**) and $\mathcal{Q}_L$ (**fl**). Secondly, the gradient of f (**fl**) w.r.t $\boldsymbol{W}$ should be calculated. Thirdly, we need to calculate the gradient of f (**fl**) w.r.t $\boldsymbol{S}$. In summary, the total time complexity of LSGL is $\mathcal{O}\left(d\left(d^2+nl+l^2\right)+l\left(l^2+d^2\right)+n\left(d^2+l^2\right)\right)$.

5 Evaluation and discussion

5.1 Datasets

To verify the effectiveness of our proposed LSGL algorithm, we performed experiments on 15 benchmark multi-label data sets, which can be downloaded from Mulan Tsoumakas et al. (2011)[1] and Huiskes and Lew (2008)[2]. The details of the data sets are summarized in Table 1.

5.2 Evaluation metrics

In this section, five common evaluation metrics were utilized for evaluation. Let $D_t=\{\boldsymbol{X}_i,\boldsymbol{Y}_i|i=1,2,\cdots,p\}$ denotes the test data set, where $\boldsymbol{Y}_i\in\boldsymbol{Y}$ denotes a set of true label vectors corresponding to the i-th instance, and $h(\boldsymbol{X}_i)$ denotes a function of a set of predicted label vectors of the i-th instance. $f(\boldsymbol{X}_i,\boldsymbol{Y}_i)$ indicates the confidence score that $\boldsymbol{X}_i$ belongs to label $\boldsymbol{Y}_i$. The $rank_f(\boldsymbol{x}_i,\boldsymbol{y})$ return the rank of $\boldsymbol{y}$ derived from $f(\boldsymbol{X}_i,\boldsymbol{Y}_i)$. The performance of the MLL algorithm can be objectively evaluated data through the five evaluation metrics Huang et al. (2019): average precision (AP), coverage (CV), Hamming loss (HL), one error (OE), and ranking loss (RL). The detailed formulation definitions of the five evaluation metrics (Huang et al. 2016, 2019; Beck and Teboulle 2009) are as follows:

– AP: Evaluate the average score of the correct labels for a particular label $\boldsymbol{y}\in\boldsymbol{Y}_i$ permutation.

$$\text{Average precision}=\frac{1}{p}\sum_{i=1}^{p}\frac{1}{|Y_i|}\sum_{y\in Y_i}\times\frac{|\{y'|rank_f(X_i,y')\leq rank_f(X_i,y'),y'\in Y_i\}|}{rank_f(X_i,y)}$$

– CV: Measures how many steps the average takes to traverse all relevant labels on the samples.

$$\text{Coverage}=\frac{1}{n_t}\sum_{i=1}^{n_t}\max_{y_i\in Y_i}rank_f(x_i,y)-1$$

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

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

Table 1 Multi-label data sets

Data set	Instance	Features	Labels	Cardinality	Density	Domain
Arts	5000	462	26	1.636	0.063	Text
Birds	645	258	19	1.014	0.053	Audio
Cal500	502	68	174	26.044	0.150	Music
Computers	5000	681	33	1.461	0.044	Text
Emotion	593	72	6	1.869	0.311	Music
Enron	1702	1001	53	3.378	0.064	Text
Flags	194	9	7	3.392	0.482	Images
Genbase	662	1186	27	1.252	0.046	Biology
Medical	978	1449	45	1.245	0.028	Text
Science	5000	743	40	1.451	0.036	Text
Social	5000	1047	39	1.283	0.033	Text
Yeast	2417	103	14	4.237	0.303	Music
Corel5k	5000	499	374	3.522	0.009	Images
Mirflickr	2780	1536	14	1.000	0.071	Images
Tmc2007	28596	500	22	2.220	0.101	Text

– HL: Measures the mismatch between the true and predicted labels of samples on a single label.

$$\text{Hamming loss} = \frac{1}{n_t}\sum_{i=1}^{n_t}\frac{1}{l}\left|h\left(\boldsymbol{X}_i\right) \neq Y_i\right|$$

– OE: Consider the case where the predicted value is top-ranked but not affiliated with the samples.

$$\text{One Error} = \frac{1}{m}\sum_{i=1}^{m}\left[Y_{i,l_i} = -1\right]$$

where $l_i = \underset{k\notin\{1,\cdots,Y_i\}}{\arg\max} f_k\left(x_i, y\right)$.

– RL: Consider the case where the rank of the irrelevant labels is lower than the rank of the relevant labels.

$$\text{Ranking Loss} = \frac{1}{n_t}\sum_{i=1}^{n_t}\frac{1}{\left|\boldsymbol{Y}_i\right|\left|\hat{\boldsymbol{Y}}_i\right|}\left|\ell_r\right|$$

where

$$\ell_r = \left\{\left(y', y''\right)\left|f\left(\boldsymbol{X_i}, y'\right)\right.\right. \leq f\left(\boldsymbol{X_i}, y''\right), \left.\left(y', y''\right) \in \boldsymbol{Y}_i \times \overline{\boldsymbol{Y}}_i\right\}.$$

5.3 Comparing algorithms

In this section, we compare our proposed method, LSGL, with the following state-of-the-art MLL methods. LSGL is a label-specific features learning method for MLL. Features learned by the LSGL algorithm can be combined with other classifiers, such as BSVM and ELM.

1. ML-kNN[3] Zhang and Zhou (2007): A lazy MLL approach is based on the classic k-nearest neighbor method. The nearest-neighbor k of ML-kNN is set to 10, and the smoothing parameter s is set to 1.
2. LIFT[4] Zhang and Lei (2014): It explores the specific features of different labels by k-means clustering of positive and negative instances. The clustering ratio r is set to 0.2.
3. LLSF[5] Huang et al. (2015): This is a MLL algorithm that learns label-specific features by adding prior knowledge about label correlations. The regularization parameters α and β are tuned in $\left\{2^{-10}, 2^{-9}, \cdots, 2^{10}\right\}$.
4. Glocal[6] Zhu et al. (2017): This is an MLL algorithm with global and local label correlation, which can handle both full and missing labels. Parameter λ is set to 1, and the basic values of other parameters are selected by fivefold cross-validation on the training set.
5. LSGL[7]: This is a MLL algorithm that jointly learns label-specific features and global and local label correlations. Trade-off parameters λ_1 are searched in $\left\{10^{-3}, 10^{-2}, \cdots, 10^{3}\right\}$, λ_2, λ_3, λ_4, and λ_5 are searched in $\left\{10^{-3}, 10^{-2}, \cdots, 10^{-1}\right\}$.
6. LSGL-BSVM: LSGL is utilized as a feature selection method. The label-specific feature of the label $\boldsymbol{y}_i$ is the feature corresponding to the nonzero element in each $\boldsymbol{W}_i (1 \leq i \leq l)$. Train the label-specific features of each binary classifier through BSVM. Trade-off parameters λ_1 to λ_5 are set to be the same as LSGL.

[3] code: https://cs.nju.edu.cn/zhouzh/.

[4] code: http://palm.seu.edu.cn/zhangml/.

[5] code: http://www.escience.cn/people/huangjun/index.html.

[6] code: http://www.lamda.nju.edu.cn.

[7] code: https://github.com/zhaodwahu/LSGL.

Table 2 Experimental results (mean ± std) in terms of average precision (↑)

Data sets	ML-*k*NN	LIFT	LLSF	GLOCAL	LSGL	LSGL-ELM	LSGL-BSVM
Arts	0.5267 ± 0.0160	0.6284 ± 0.0107	0.6165 ± 0.0097	0.6249 ± 0.0108	0.6371 ± 0.0066	**0.6446 ± 0.0165**	0.6388 ± 0.0089
Birds	0.7377 ± 0.0467	0.7350 ± 0.0285	0.7548 ± 0.0426	0.7579 ± 0.0319	**0.7739 ± 0.0494**	0.7735 ± 0.0370	0.7736 ± 0.0279
Cal500	0.4953 ± 0.0143	0.5003 ± 0.0185	0.4988 ± 0.0145	0.5076 ± 0.0173	**0.5078 ± 0.0143**	0.5064 ± 0.0133	0.4983 ± 0.0117
Computers	0.6488 ± 0.0121	0.7119 ± 0.0122	0.7090 ± 0.0153	0.6975 ± 0.0149	0.7145 ± 0.0112	0.7217 ± 0.0126	**0.7236 ± 0.0148**
Emotion	0.7909 ± 0.0283	0.8171 ± 0.0160	0.8087 ± 0.0375	0.8099 ± 0.0298	0.8242 ± 0.0198	**0.8283 ± 0.0287**	0.8238 ± 0.0316
Enron	0.6300 ± 0.0160	0.6999 ± 0.0161	0.7032 ± 0.0250	0.6553 ± 0.0119	0.7133 ± 0.0225	**0.7221 ± 0.0186**	0.7044 ± 0.0158
Flags	0.7913 ± 0.0397	0.7950 ± 0.0523	0.8070 ± 0.0460	0.7958 ± 0.0332	**0.8147 ± 0.0405**	0.8137 ± 0.0377	0.8096 ± 0.0277
Genbase	0.9871 ± 0.0143	0.9955 ± 0.0059	0.9947 ± 0.0062	0.9943 ± 0.0062	0.9966 ± 0.0043	0.9961 ± 0.0034	**0.9966 ± 0.0030**
Medical	0.8103 ± 0.0234	0.8772 ± 0.0203	0.9018 ± 0.0176	0.8668 ± 0.0301	0.9128 ± 0.0133	**0.9134 ± 0.0238**	0.9108 ± 0.0220
Science	0.5542 ± 0.0161	0.6075 ± 0.0155	0.5817 ± 0.0150	0.6016 ± 0.0138	0.6101 ± 0.0085	**0.6268 ± 0.0189**	0.6225 ± 0.0108
Social	0.7633 ± 0.0101	0.7893 ± 0.0130	0.7731 ± 0.0101	0.7733 ± 0.0106	0.7856 ± 0.0135	**0.7913 ± 0.0246**	0.7954 ± 0.0087
Yeast	0.7632 ± 0.0204	0.7689 ± 0.0132	0.7618 ± 0.0155	0.7608 ± 0.0081	**0.7691 ± 0.0092**	0.7678 ± 0.0178	0.7538 ± 0.0187
Corel5k	0.2458 ± 0.0063	0.2844 ± 0.0112	0.2846 ± 0.0081	0.2850 ± 0.0064	0.3151 ± 0.0069	**0.3294 ± 0.0042**	0.2897 ± 0.0047
Mirflickr	0.8149 ± 0.0073	0.8528 ± 0.0142	0.8637 ± 0.0103	0.8597 ± 0.0134	0.8670 ± 0.0016	**0.8671 ± 0.0049**	0.8621 ± 0.0089
Tmc2007	0.7946 ± 0.0042	0.8418 ± 0.0025	0.8398 ± 0.0021	0.8318 ± 0.0019	0.8418 ± 0.0014	**0.8583±0.0025**	0.8476 ± 0.0033

Table 3 Experimental results (mean ± std) in terms of coverage (↓)

Data Sets	ML-*k*NN	LIFT	LLSF	GLOCAL	LSGL	LSGL-ELM	LSGL-BSVM
Arts	0.2055 ± 0.0085	0.1707 ± 0.0081	0.2355 ± 0.0154	0.2105 ± 0.0075	0.1999 ± 0.0106	0.1901 ± 0.0095	**0.1692 ± 0.0125**
Birds	0.1466 ± 0.0300	0.1646 ± 0.0185	0.1626 ± 0.0252	0.1631 ± 0.0300	**0.1356 ± 0.0295**	0.1401 ± 0.0306	0.1380 ± 0.0192
Cal500	0.7446 ± 0.0232	0.7572 ± 0.0196	**0.7336 ± 0.0245**	0.7359 ± 0.0238	0.7487 ± 0.0186	0.7460 ± 0.0271	0.7446 ± 0.0143
Computers	0.1249 ± 0.0079	0.1039 ± 0.0057	0.1395 ± 0.0115	0.1452 ± 0.0152	0.1254 ± 0.0099	0.1207 ± 0.0098	**0.1002 ± 0.0650**
Emotion	0.2998 ± 0.0275	0.2864 ± 0.0238	0.2979 ± 0.0391	0.2993 ± 0.0314	0.2828 ± 0.0200	**0.2815 ± 0.0223**	0.2837 ± 0.0334
Enron	0.2457 ± 0.0189	**0.2192 ± 0.0181**	0.2229 ± 0.0184	0.3216 ± 0.0138	0.2381 ± 0.0156	0.2208 ± 0.0106	0.2306 ± 0.0155
Flags	0.5742 ± 0.0370	0.5539 ± 0.0379	0.5513 ± 0.0456	0.5601 ± 0.0332	0.5461 ± 0.0556	0.5509 ± 0.0285	**0.5448 ± 0.0363**
Genbase	0.0208 ± 0.0138	0.0149 ± 0.0107	0.0167 ± 0.0126	0.0150 ± 0.0076	**0.0132 ± 0.0062**	0.0144 ± 0.0078	**0.0148 ± 0.0088**
Medical	0.0585 ± 0.0166	0.0376 ± 0.0129	0.0313 ± 0.0120	0.0487 ± 0.0109	**0.0224 ± 0.0065**	0.0268 ± 0.0060	0.0375 ± 0.0197
Science	0.1474 ± 0.0103	**0.0961 ± 0.0074**	0.2074 ± 0.0172	0.1814 ± 0.0110	0.1749 ± 0.0105	0.1556 ± 0.0093	0.1275 ± 0.0102
Social	0.0729 ± 0.0068	0.0690 ± 0.0086	0.1077 ± 0.0078	0.1096 ± 0.0128	0.0889 ± 0.0094	0.0880 ± 0.0072	**0.0688 ± 0.0074**
Yeast	0.4483 ± 0.0177	0.4548 ± 0.0106	0.4551 ± 0.0186	0.4562 ± 0.0110	**0.4473 ± 0.0159**	0.4481 ± 0.0215	0.4680 ± 0.0105
Corel5k	0.3074 ± 0.0073	0.4548 ± 0.0106	0.3740 ± 0.0056	0.3337 ± 0.0050	0.3569 ± 0.0081	0.3856 ± 0.0073	**0.2891 ± 0.0067**
Mirflickr	0.0477 ± 0.0044	0.0372 ± 0.0042	0.0383 ± 0.0039	0.0395 ± 0.0044	0.0355 ± 0.0023	0.0391 ± 0.0037	**0.0348 ± 0.0035**
Tmc2007	0.1462 ± 0.0025	0.1233 ± 0.0014	0.1228 ± 0.0031	0.1301 ± 0.0025	0.1213 ± 0.0022	**0.1127±0.0014**	0.1207 ± 0.0016

Table 4 Experimental results (mean ± std) in terms of Hamming loss (↓)

Data sets	ML-*k*NN	LIFT	LLSF	GLOCAL	LSGL	LSGL-ELM	LSGL-BSVM
Arts	0.0596 ± 0.0018	0.0528 ± 0.0015	0.0571 ± 0.0017	0.0596 ± 0.0034	0.0521 ± 0.0018	**0.0517 ± 0.0023**	0.0525 ± 0.0027
Birds	0.0543 ± 0.0066	0.0518 ± 0.0049	0.0512 ± 0.0040	0.0511 ± 0.0076	0.0503 ± 0.0078	**0.0467 ± 0.0097**	0.0502 ± 0.0091
Cal500	0.1388 ± 0.0038	0.1379 ± 0.0040	0.1425 ± 0.0056	0.1388 ± 0.0074	0.1368 ± 0.0038	**0.1364 ± 0.0032**	0.1378 ± 0.0057
Computers	0.0394 ± 0.0014	0.0328 ± 0.0006	0.0333 ± 0.0025	0.0476 ± 0.0041	0.0346 ± 0.0018	**0.0323 ± 0.0013**	0.0329 ± 0.0010
Emotion	0.2031 ± 0.0165	0.1856 ± 0.0166	0.1898 ± 0.0298	0.2009 ± 0.0296	**0.1804 ± 0.0178**	0.1812 ± 0.0211	0.1901 ± 0.0207
Enron	0.0521 ± 0.0022	0.0456 ± 0.0027	0.0547 ± 0.0036	0.0719 ± 0.0082	0.0453 ± 0.0025	**0.0440 ± 0.0016**	0.0465 ± 0.0017
Flags	0.3259 ± 0.0364	0.3378 ± 0.0300	0.2890 ± 0.0300	0.3113 ± 0.0167	0.2857 ± 0.0398	**0.2829 ± 0.0398**	0.2963 ± 0.0343
Genbase	0.0048 ± 0.0022	0.0024 ± 0.0009	0.0009 ± 0.0010	0.0025 ± 0.0044	**0.0005 ± 0.0006**	0.0011 ± 0.0010	0.007 ± 0.0005
Medical	0.0158 ± 0.0017	0.0117 ± 0.0014	0.0102 ± 0.0012	0.0183 ± 0.0052	0.0095 ± 0.0012	**0.0091 ± 0.0014**	0.0098 ± 0.0011
Science	0.0323 ± 0.0009	0.0308 ± 0.0011	0.0344 ± 0.0009	0.0325 ± 0.0017	0.0309 ± 0.0009	**0.0297 ± 0.0011**	0.0307 ± 0.0010
Social	0.0209 ± 0.0009	0.0194 ± 0.0011	0.0207 ± 0.0008	0.0205 ± 0.0009	0.0205 ± 0.0011	**0.0187 ± 0.0011**	0.0194 ± 0.0009
Yeast	0.1946 ± 0.0114	**0.1925 ± 0.0065**	0.1993 ± 0.0089	0.1995 ± 0.0056	0.1961 ± 0.0067	0.1972 ± 0.0119	0.1994 ± 0.0074
Corel5k	0.0094 ± 0.0004	0.0094 ± 0.0004	0.0118 ± 0.0000	0.0095 ± 0.0000	0.0094 ± 0.0004	**0.0093 ± 0.0004**	**0.0093 ± 0.0004**
Mirflickr	0.0399 ± 0.0001	0.0340 ± 0.0037	0.0323 ± 0.0017	0.0343 ± 0.0023	0.0327 ± 0.0016	**0.0303 ± 0.0001**	0.0325 ± 0.0024
Tmc2007	0.0652 ± 0.0010	0.0538 ± 0.0009	0.0606 ± 0.0008	0.0609 ± 0.0004	0.0573 ± 0.0008	**0.0535 ± 0.0010**	0.0550 ± 0.0008

Table 5 Experimental results (mean ± std) in terms of one error (↓)

Data Sets	ML-*k*NN	LIFT	LLSF	GLOCAL	LSGL	LSGL-ELM	LSGL-BSVM
Arts	0.6088 ± 0.0252	0.4532 ± 0.0154	0.4508 ± 0.0171	0.4468 ± 0.0165	0.4364 ± 0.0166	**0.4328 ± 0.0198**	0.4414 ± 0.0113
Birds	0.3149 ± 0.0595	0.3037 ± 0.0347	0.2983 ± 0.0652	0.2835 ± 0.0416	0.2738 ± 0.0648	0.2754 ± 0.0464	**0.2674 ± 0.0501**
Cal500	0.1218 ± 0.0315	0.1209 ± 0.0466	0.1166 ± 0.0430	0.1164 ± 0.0421	0.1152 ± 0.0573	**0.1147 ± 0.0451**	0.1171 ± 0.0405
Computers	0.4254 ± 0.0214	0.3488 ± 0.0230	0.3446 ± 0.0211	0.3460 ± 0.0276	0.3438 ± 0.0208	**0.3332 ± 0.0156**	0.3376 ± 0.0188
Emotion	0.2971 ± 0.0402	0.2391 ± 0.0381	0.2510 ± 0.0654	0.2445 ± 0.0553	0.2358 ± 0.0457	**0.2221 ± 0.0545**	0.2274 ± 0.0419
Enron	0.3085 ± 0.0310	0.2404 ± 0.0372	0.2331 ± 0.0469	0.2675 ± 0.0246	**0.2175 ± 0.0225**	0.2175 ± 0.0305	0.2558 ± 0.0273
Flags	0.2438 ± 0.0733	0.2562 ± 0.1291	0.2261 ± 0.1099	0.2267 ± 0.0726	0.2072 ± 0.0653	0.2142 ± 0.0910	**0.2027 ± 0.0731**
Genbase	0.0138 ± 0.0174	**0.0000 ± 0.0000**	0.0030 ± 0.0060	0.0015 ± 0.0045	**0.0000 ± 0.0000**	**0.0000 ± 0.0000**	**0.0000 ± 0.0000**
Medical	0.2465 ± 0.0412	0.1596 ± 0.0271	0.1349 ± 0.0285	0.1627 ± 0.0513	0.1217 ± 0.0226	**0.1176 ± 0.0430**	0.1197 ± 0.0243
Science	0.5498 ± 0.0203	0.4860 ± 0.0244	0.4970 ± 0.0188	0.4826 ± 0.0212	0.4728 ± 0.0152	**0.4592 ± 0.0217**	0.4652 ± 0.0148
Social	0.3038 ± 0.0152	0.2638 ± 0.0145	0.2684 ± 0.0136	0.2742 ± 0.0158	0.2674 ± 0.0198	0.2594 ± 0.0062	**0.2578 ± 0.0116**
Yeast	0.2313 ± 0.0332	0.2201 ± 0.0245	0.2213 ± 0.0258	0.2251 ± 0.0182	0.2234 ± 0.0177	**0.2192 ± 0.0241**	0.2280 ± 0.0398
Corel5k	0.7358 ± 0.0124	0.6830 ± 0.0204	0.6468 ± 0.0125	0.6633 ± 0.0160	0.6264 ± 0.0122	**0.6112 ± 0.0097**	0.6488 ± 0.0152
Mirflickr	0.2971 ± 0.0117	0.2392 ± 0.0215	0.2209 ± 0.0161	0.2269 ± 0.0217	0.2173 ± 0.0058	**0.2169 ± 0.0063**	0.2252 ± 0.0120
Tmc2007	0.2296 ± 0.0063	0.1604 ± 0.0015	0.1822 ± 0.0044	0.1905 ± 0.0034	0.1810 ± 0.0024	**0.1586 ± 0.0033**	0.1660 ± 0.0067

Table 6 Experimental results (mean ± std) in terms of ranking loss (↓)

Data Sets	ML-kNN	LIFT	LLSF	GLOCAL	LSGL	LSGL-ELM	LSGL-BSVM
Arts	0.1479 ± 0.0070	0.1116 ± 0.0046	0.1615 ± 0.0084	0.1400 ± 0.0058	0.1301 ± 0.0050	0.1216 ± 0.0069	**0.1101 ± 0.0072**
Birds	0.1015 ± 0.0283	0.1131 ± 0.0145	0.1071 ± 0.0266	0.1084 ± 0.0212	**0.0859 ± 0.0236**	0.0890 ± 0.0159	0.0890 ± 0.0191
Cal500	0.1820 ± 0.0062	0.1818 ± 0.0092	0.1820 ± 0.0092	**0.1773 ± 0.0090**	0.1787 ± 0.0081	0.1785 ± 0.0073	0.1811 ± 0.0052
Computers	0.0861 ± 0.0050	0.0673 ± 0.0040	0.0978 ± 0.0083	0.1113 ± 0.0076	0.0848 ± 0.0130	0.0799 ± 0.0048	**0.0650 ± 0.0052**
Emotion	0.1685 ± 0.0275	0.1467 ± 0.0177	0.1574 ± 0.0337	0.1569 ± 0.0311	0.1442 ± 0.0210	**0.1410 ± 0.0228**	0.1426 ± 0.0341
Enron	0.0920 ± 0.0089	0.0750 ± 0.0061	0.0813 ± 0.0093	0.1242 ± 0.0115	0.0834 ± 0.0103	**0.0730 ± 0.0046**	0.0782 ± 0.0051
Flags	0.2452 ± 0.0396	0.2350 ± 0.0404	0.2200 ± 0.0446	0.2353 ± 0.0235	**0.2108 ± 0.0508**	0.2148 ± 0.0336	0.2199 ± 0.0313
Genbase	0.0063 ± 0.0068	0.0029 ± 0.0048	0.0047 ± 0.0079	0.0032 ± 0.0040	**0.0020 ± 0.0036**	0.0025 ± 0.0029	0.0028 ± 0.0033
Medical	0.0400 ± 0.0121	0.0241 ± 0.0099	0.0210 ± 0.0092	0.0339 ± 0.0088	**0.0130 ± 0.0049**	0.0159 ± 0.0052	0.0239 ± 0.0150
Science	0.1124 ± 0.0088	0.0961 ± 0.0074	0.1579 ± 0.0153	0.1339 ± 0.0082	0.1272 ± 0.0084	0.1106 ± 0.0075	**0.0918 ± 0.0073**
Social	0.0526 ± 0.0052	0.0478 ± 0.0054	0.0763 ± 0.0058	0.0768 ± 0.0065	0.0599 ± 0.0050	0.0594 ± 0.0062	**0.0476 ± 0.0054**
Yeast	0.1682 ± 0.0170	0.1650 ± 0.0095	0.1690 ± 0.0112	0.1708 ± 0.0071	**0.1648 ± 0.0100**	0.1653 ± 0.0141	0.1742 ± 0.0100
Corel5k	0.1350 ± 0.0026	**0.1231 ± 0.0047**	0.1743 ± 0.0038	0.1524 ± 0.0029	0.1503 ± 0.0052	0.1672 ± 0.0017	0.1232 ± 0.0028
Mirflickr	0.0514 ± 0.0047	0.0400 ± 0.0045	0.0413 ± 0.0042	0.0425 ± 0.0048	0.0382 ± 0.0025	0.0421 ± 0.0040	**0.0375 ± 0.0038**
Tmc2007	0.0603 ± 0.0013	0.0394 ± 0.0010	0.0419 ± 0.0012	0.0462 ± 0.0014	0.0408 ± 0.0008	**0.0357 ± 0.0008**	0.0403 ± 0.0010

7. LSGL-ELM: In addition, we utilize LSGL as a label-specific feature extraction method and then explore an extreme learning machine (ELM) model for multi-label classification. This method is typically different from other label-specific feature extraction methods. We utilize linear models for classification and consider neural network learning methods, which can be used to verify further the effectiveness of the LSGL algorithm in label-specific feature extraction. Parameters λ_1 to λ_5 are set to be the same as LSGL. The kernel function type selection RBF kernel, and the kernel parameter and the parameter C are set to 1.

LIBSVM Chang and Lin (2011) is utilized as the base binary learner for BSVM, LIFT, and LSGL-BSVM, where the kernel function is configured as a linear kernel, and the parameter C as 1.

5.4 Experimental results

The experiments are implemented using MATLAB 2016a on a standard Windows PC with an Intel 4.2-GHz CPU and 16-GB RAM. For each data set, the performance was systematically evaluated using tenfold cross-validation. In detail, ten repeated experiments were performed, and the average result (mean±standard deviation) of each comparison algorithm was recorded. At the beginning of each experiment, we randomly selected 90% of the instances for training and the remaining 10% for tests. The results of ten independent replicate experiments are reported in Table 2 to Table 6. It should be noted that the "↑" indication after the evaluation index indicates that the larger the value, the better the classification performance, and the "↓" indication indicates that the smaller the value of the evaluation index, the better the classification performance. In addition, the best results of the comparison algorithms are shown in bold.

The experimental results of the five evaluation metrics on the 15 data sets are listed in Tables 2, 3, 4, 5, 6, and we can conclude as follows:

(a). The experimental comparison results of the AP of each algorithm are given in Table 2. It can be seen that LSGL-ELM achieves better performance than all the compared algorithms. As shown in Table 3, LSGL-BSVM achieves better performance than all the compared algorithms. From Table 4 and Table 5, we can see that LSGL-ELM obtains excellent performance on HL and OE, which is better than LLSF, LIFT, GLOCAL, LSGL, and LSGL-BSVM. However, from Table 6, in the case of RL, we can see that LSGL achieves better performance compared with other algorithms.

(b). We further observe that the performance of LSGL is better than or roughly equivalent to LSGL-BSVM and LSGL-ELM on the data sets *yeast*, *genbase*, and *birds*. We can obtain the conclusion that the LSGL algorithm and the LSGL-BSVM algorithm have the same performance.

Table 7 Summary of the Friedman statistics F_F (k=7, N=15) and the critical value in terms of each evaluation metric

Metric	F_F	Critical value($\alpha = 0.05$)
Average precision	33.0651	2.2086
Coverage	6.0728	
Hamming loss	19.3019	
One error	33.7531	
Ranking loss	9.7736	

(c). LSGL-ELM and LSGL-BSVM are extensions of the LSGL algorithm, utilizing label-specific features for MLL. However, LSGL-ELM achieves better performance than LSGL-BSVM in terms of the three evaluation metrics except for CV and RL. LSGL-BSVM utilizes BSVM as a binary classifier. The BSVM would not directly perform MLL, while ELM can directly perform MLL. From the results, the multi-label classification effect of the ELM algorithm is somewhat better than BSVM.

Furthermore, a statistical hypothesis test was utilized to verify and compare the relative performance of various algorithms. The Friedman test Zhang et al. (2018) was utilized for performance analysis. Table 7 summarizes the Friedman statistics F_F and the corresponding critical values of the various evaluation metrics. As shown in Table 7, at the significance level α=0.05, each evaluation metric is rejected when the null hypothesis is that all comparison algorithms are equivalently executed. Therefore, the *Nemenyi* test (He et al. 2019; Zhang and Lei 2014; Demšar 2006) is utilized as a post hoc test to compare the performance of each algorithm and observe whether the LSGL algorithm is competitive. There is a significant difference in performance between the two classifiers if the corresponding average ranking reaches at least a critical difference (CD):

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

For Nemenyi test, q_α=2.948 at significance level α=0.05, and thus, CD=2.3254(k=7, N=15). Figure 1 indicates the CD diagrams of each algorithm under different evaluation metrics, respectively. In each subfigure, two or more algorithms are connected by colored solid lines indicating that there is no significant difference in performance between them. Otherwise, any algorithms that are not connected by a solid line is considered to have a significant difference in performance. For each approach, there are 30 comparative results (six parallel approaches and five evaluation metrics). Through the above experimental results, we can all obtain the following analytical results:

– Intuitively, if there is a colored solid line connection among the comparison algorithms and the LSGL-ELM algorithm, it means that there is no statistically significant difference between the LSGL-ELM algorithm and other comparison algorithms. Specifically, it can be observed from Fig. 1(a) that there is no significant difference in AP among the LSGL-ELM, LSGL, and LSGL-BSM; as shown in Fig.1(b), there is no significant difference in the CV term among the LSGL-ELM, LSGL, LSGL-BSVM, and LIFT; as shown in Fig.1(c), there is no significant

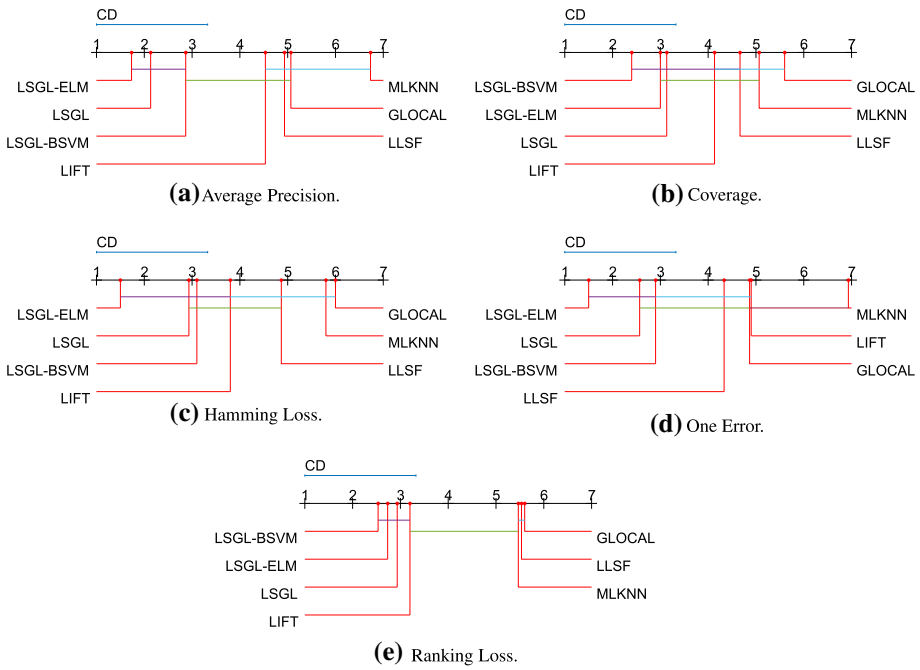

Fig. 1 Performance comparison of various algorithms

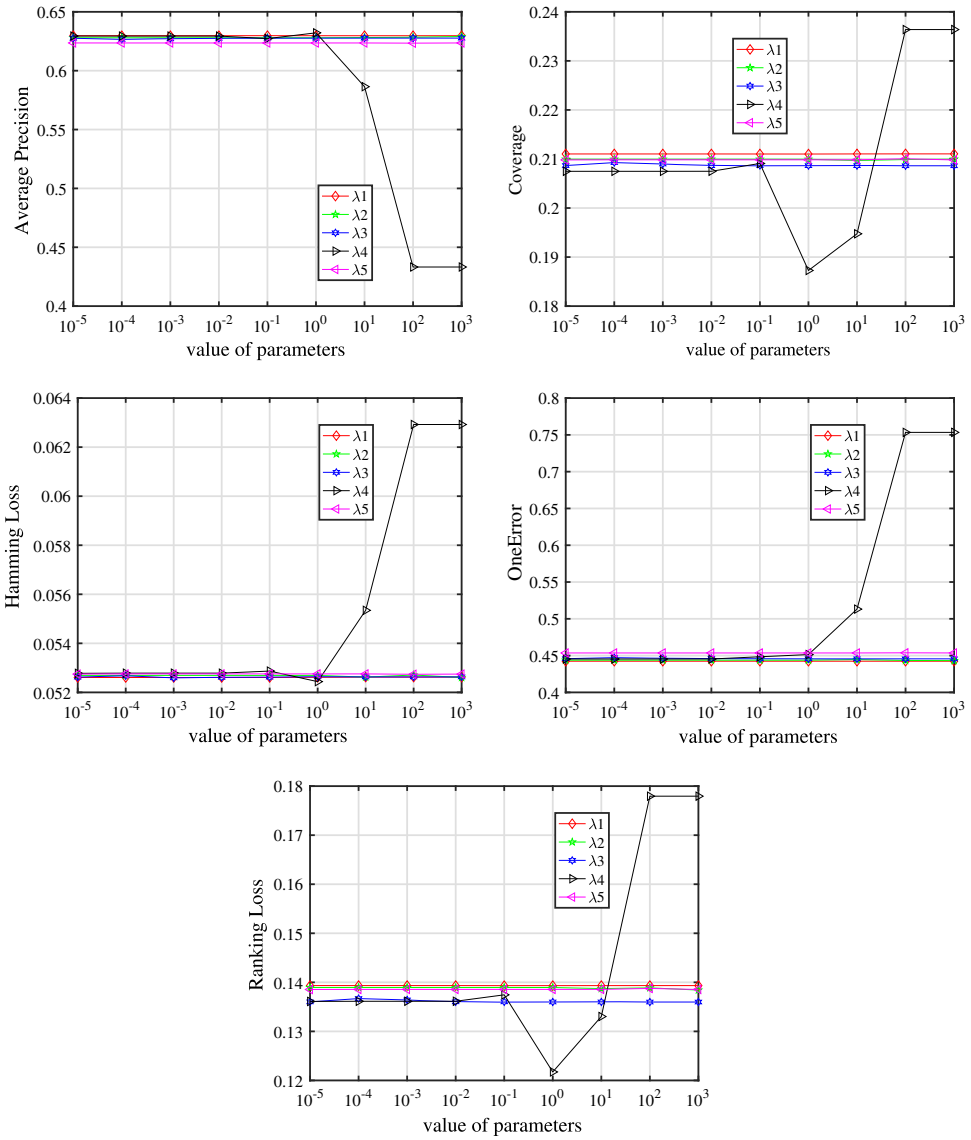

Fig. 2 Parameter sensitivity analysis of LSGL on Arts data set

difference among the LSGL-ELM, LSGL, LSGL-BSVM and LIFT algorithms on the HL term. From Fig.1(d), it can be observed that there is no significant difference among the LSGL-ELM, LSGL, and LSGL-BSVM algorithms on the OE term; as shown in Fig.1(e), there is no significant difference among LSGL-ELM, LSGL, LSGL-BSVM, and LIFT on the RL term. Based on the above analysis, in 50% of cases, the LSGL-ELM algorithm is significantly better than other comparison algorithms. In Fig.1(b) and (e), it is noted that the LSGL-BSVM algorithm obtains the optimal ranking under the coverage and ranking loss.

– From the comparative analysis of the experimental results of LSGL-BSVM and LIFT, we can note that: although both utilize BSVM as a classifier when exploring the label-specific feature learning, the observable performance of LSGL-BSM is better. A reliable understanding is that LIFT ignores the improvement of algorithm performance caused by LC. On the classifier, the solution process of ELM is faster than BSVM.
– From the comparative analysis of LSGL and LLSF experimental consequences, we can see that both add label correlation to learn label-specific features under a unified model, but the performance of LLSF is worse than LSGL. The main reason is that LLSF adds label correlation as a prior knowledge to the model, and LSGL is a method to learn global and local LC and label-specific features jointly.
– In comparison between GLOCAL and LSGL, it can be found that LSGL performs better than GLOCAL. The intuitive reason is that GLOCAL uses a linear combination of multiple-label manifold regularizes in the local LC and is not obtained by direct iterative learning. And

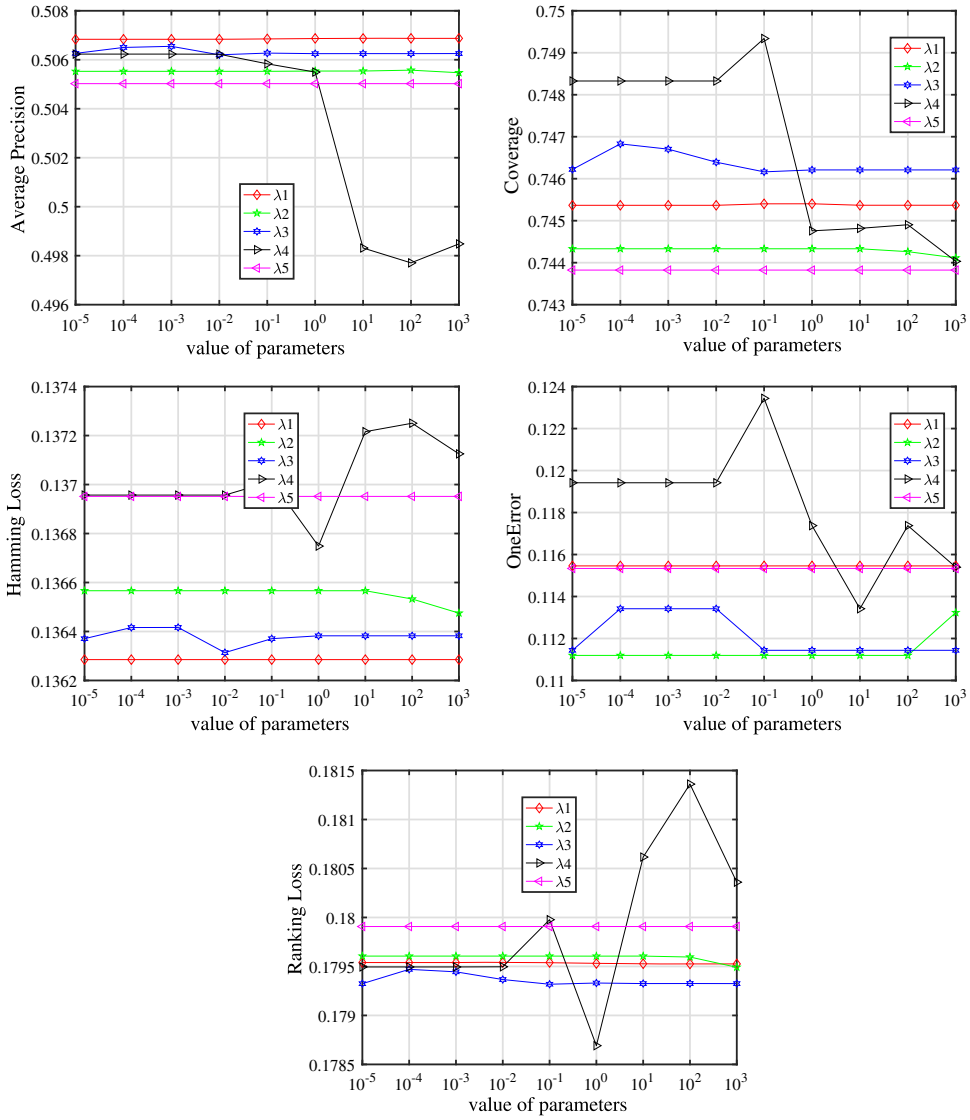

Fig. 3 Parameter sensitivity analysis of LSGL algorithm on Cal500 data set

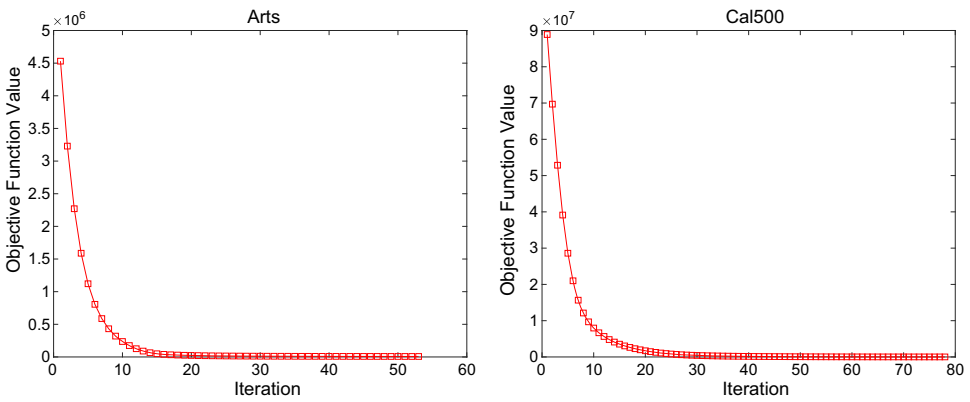

Fig. 4 Convergence trend analysis

LSGL avoids such problems, and it makes full use of the complex asymmetric relationship among labels.

– From the comparative analysis of the three approaches such as LSGL, LSGL-ELM, and LSGL-BSVM, we can see that the results of LSGL-ELM are significantly better than the other two approaches. Intuitive analysis shows that it is because the neural network-based classification algorithm has nonlinear data processing capabilities.

The comparative analysis of the performance of LSGL, LLSF, and GLOCAL further verifies the effectiveness of LSGL to learn label-specific features for global and local LC through manifold regularization. It also further supports the correctness of our hypothesis of global label consistency and local label smoothness.

5.5 Parameter sensitivity analysis

In the LSGL algorithm, there are five important parameters λ_1, λ_2, λ_3, λ_4, and λ_5. λ_1 controls the influence of LC on the prediction effect of the model. The larger the value, the smaller the impact of LC, and vice versa. λ_2 controls the similarity between models' weight coefficients for every two labels, λ_3 controls the local label smoothness term, λ_4 controls label-specific feature sparsity between any two class labels, and λ_5 controls the sparsity of global and local label correlations between different class labels. We use the $Arts$ and $Cal500$, to analyze the effect of parameters λ_1, λ_2, λ_3, λ_4, and λ_5 on the proposed method. The value of one of the parameters changes within a certain range, while the value of the other parameters is fixed to a certain value, respectively, and then observing and recording the changes in the experimental results.

The evaluation metrics were analyzed by average precision, coverage, Hamming loss, one error, and ranking loss for parameter sensitivity experiments, and the average results of LSGL with different values are depicted in Figs.2 and 3.

From Figs.2 and 3, we can find that when the value of λ_4 is too large, the performance of LSGL is poor. Because λ_4 controls the sparseness of the label-specific features, when the value of λ_4 is too large, the algorithm cannot effectively obtain the distinctive features of the label. In addition, λ_5 controls the sparseness of global and local label correlations between class labels. Some functional label correlations are filtered out because of a larger λ_5 value, so the parameter λ_5 should not be set too large. Regarding the changes in other parameters, it can be noticed that it is not sensitive to the performance of LSGL. In most cases, the highest performance is obtained when all parameters have a value of 10^{-1}.

5.6 LSGL algorithm iteration efficiency

In this section, we show the iterative convergence of the LSGL algorithm. Specifically, the convergence curves of LSGL on the $Arts$ and $Cal500$ data sets are shown in Fig.4. We can see that for the $Arts$ dataset, LSGL tends to converge 15 iterations, and for the $Cal500$ dataset, LSGL tends to converge 20 times. The convergence trends on other data sets are the same as the trends reported in Fig.4. Overall, for the datasets used in the experiments, LSGL can converge at a faster rate.

6 Conclusion

In this article, we propose an MLL method called LSGL. Based on the assumption of global label consistency and local label smoothness, LSGL conducts joint learning of global and local label correlation and label-specific features. In addition, we extract label-specific features through the nonzero entries of the coefficient matrix and then combine the feedforward neural network learning method ELM and the SVM method BSVM for multi-label classification. Compared with the state-of-the-art MLL method on 15 benchmark multi-label data sets, it verifies the effectiveness and robustness of LSGL using global and local label correlation for label-specific feature learning.

The shortcomings of our method are also prominent. First, there are two variables in LSGL that needs to be optimized through alternate iterations, which are likely to fall into the optimal local solution. Secondly, LSGL is a typical linear optimization problem, so it cannot solve where data are inseparable. In future work, we will focus on solving the above problems.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 62071001), the Anhui Natural Science Foundation of China (Nos. 2008085MF192 and 2008085MF183), the Key Science Project of Anhui Education Department of China (Nos. KJ2018A0012, KJ2019A0023, and KJ2019A0022), and the CERNET Innovation Project of China (Nos. NGII20180612, NGII20180312, and NGII20180624).

Declarations

Conflict of interest The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

Al-Salemi B, Ayob M, Noah SAM (2018) Feature ranking for enhancing boosting-based multi-label text categorization. Exp Syst Appl 113:531–543

Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imag Sci 2(1):183–202

Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Patt Recog 37(9):1757–1771

Chang C-C, Lin C-J (2011) Libsvm: A library for support vector machines. ACM Trans Intell syst technol (TIST) 2(3):27

Charte F, Rivera AJ, Del Jesus MJ, Herrera F (2014) Li-mlc: a label inference methodology for addressing high dimensionality in the label space for multilabel classification. IEEE Trans Neural Netw Learn Syst 25(10):1842–1854

Che X, Chen D, Mi J (2020) A novel approach for learning label correlation with application to feature selection of multi-label data. Inf Sci 512:795–812

Cheng Y, Zhao D, Wenfa Z, Yibin W (2018) Multi-label learning of non-equilibrium labels completion with mean shift. Neurocomputing 321:92–102

Cheng Y, Zhao D, Wang Y, Pei G (2019) Multi-label learning with kernel extreme learning machine autoencoder. Knowl Bas Syst 178:1–10

Combettes PL, Wajs VR (2005) Signal recovery by proximal forward-backward splitting. Multis Model Sim 4(4):1168–1200

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(Jan):1–30

Elisseeff A, Weston J (2002) A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, pages 681–687

Fu B, Xu G, Wang Z, Cao L (2013) Leveraging supervised label dependency propagation for multi-label learning. In *2013 IEEE 13th International Conference on Data Mining*, pages 1061–1066. IEEE

Fürnkranz J, Hüllermeier E, Mencía EL, Brinker K (2008) Multilabel classification via calibrated label ranking. Mach learn 73(2):133–153

Gargiulo F, Silvestri S, Ciampi M, De Pietro G (2019) Deep neural network for hierarchical extreme multi-label text classification. Appl Soft Comput 79:125–138

Gibaja E, Ventura S (2015) A tutorial on multilabel learning. ACM Comput Surv (CSUR) 47(3):52

Guan R, Wang X, Yang MQ, Zhang Yu, Zhou F, Yang C, Liang Y (2018) Multi-label deep learning for gene function annotation in cancer pathways. Sci Rep 8(1):267

He Z-F, Yang M, Gao Y, Liu H-D, Yin Y (2019) Joint multi-label classification and label correlations with missing labels and feature selection. Knowl Based Syst 163:145–158

Huang J, Li G, Huang Q, Wu X (2015) Learning label specific features for multi-label classification. In *2015 IEEE International Conference on Data Mining*, pages 181–190. IEEE

Huang SJ, Zhou ZH (2012) Multi-label learning by exploiting label correlations locally. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26

Huang G-B (2014) An insight into extreme learning machines: random neurons, random features and kernels. Cogn Comput 6(3):376–390

Huang G, Huang G-B, Song S, You K (2015) Trends in extreme learning machines: A review. Neural Netw 61:32–48

Huang J, Li G, Huang Q, Xindong W (2016) Learning label-specific features and class-dependent labels for multi-label classification. IEEE Trans Knowl Data Eng 28(12):3309–3323

Huang J, Qin F, Zheng X, Cheng Z, Yuan Z, Zhang W, Huang Q (2019) Improving multi-label classification with missing labels by learning label-specific features. Inf Sci 492:124–146

Huiskes MJ, Lew MS (2008) The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 39–43

Jun Xie LY, Zhu L, Duan G (2019) Conditional entropy based classifier chains for multi-label classification. Neurocomputing 335:185–194

Liu Y, Wen K, Gao Q, Gao X, Nie F (2018) Svm based multi-label learning with missing labels for image annotation. Patt Recog 78:307–317

Ma J, Zhang H, Chow TWS (2021) Multilabel classification with label-specific features and classifiers: A coarse- and fine-tuned framework. IEEE Trans Cyber 51(2):1028–1042

Qiao L, Zhang L, Sun Z, Liu X (2017) Selecting label-dependent features for multi-label classification. Neurocomputing 259:112–118

Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. Machine Learn 85(3):333

Ren W, Zhang L, Jiang B, Wang Z, Guo G, Liu G (2017) Robust mapping learning for multi-view multi-label classification with missing labels. In *International Conference on Knowledge Science, Engineering and Management*, pages 543–551. Springer,

Rezaei-Ravari M, Eftekhari M, Saberi-Movahed F (2021) Regularizing extreme learning machine by dual locally linear embedding manifold learning for training multi-label neural network classifiers. Eng Appl Artif Intell 97:104062

Sun L, Kudo M, Kimura K (2016) Multi-label classification with meta-label-specific features. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1612–1617. IEEE

Suping X, Yang X, Hualong Yu, Dong-Jun Yu, Yang J, Tsang ECC (2016) Multi-label learning with label-specific feature reduction. Knowl Bas Syst 104:52–61

Tsoumakas G, Katakis I, Vlahavas I (2009) Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer

Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I (2011) Mulan: a java library for multi-label learning. J Mach Learn Res 12(Jul):2411–2414

Wang J, Yang Y, Mao J, Huang ZHC, Xu W (2016) Cnn-rnn: a unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294

Weng W, Lin Y, Shunxiang W, Li Y, Kang Y (2018) Multi-label learning based on label-specific features and local pairwise label correlation. Neurocomputing 273:385–394

Xu L, Wang Z, Shen Z, Wang Y, Chen E (2014) Learning low-rank label correlations for multi-label classification with missing labels. In *2014 IEEE International Conference on Data Mining*, pages 1067–1072. IEEE

Zhang M-L, Lei W (2014) Lift: Multi-label learning with label-specific features. IEEE Trans Patt Anal Mach Intell 37(1):107–120

Zhang Yu, Yeung D-Y (2013) Multilabel relationship learning. ACM Trans Knowl Dis Data (TKDD) 7(2):7

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

Zhang M-L, Zhou Z-H (2007) Ml-knn: a lazy learning approach to multi-label learning. Pattern Rec 40(7):2038–2048

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

Zhang J, Qi W, Shen C, Zhang J, Jianfeng L (2018) Multilabel image classification with regional latent semantic dependencies. IEEE Trans Multim 20(10):2801–2813

Zhang J, Li C, Cao D, Lin Y, Songzhi S, Dai L, Li S (2018) Multi-label learning with label-specific features by resolving label correlations. Knowl Bas Syst 159:148–157

Zhang J, Luo Z, Li C, Zhou C, Li S (2019) Manifold regularized discriminative feature selection for multi-label learning. Patt Rec 95:136–150

Zhu Y, Kwok JT, Zhou Z-H (2017) Multi-label learning with global and local label correlation. IEEE Trans Knowl Data Eng 30(6):1081–1094

Zhu C, Miao D, Wang Z, Zhou R, Wei L, Zhang X (2020) Global and local multi-view multi-label learning. Neurocomputing 371:67–77