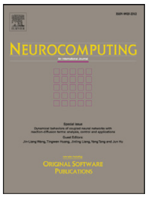
Neurocomputing 395 (2020) 95–106

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Feature selection with missing labels based on label compression and local feature correlation

Lin Jiang[a], Guoxian Yu[a], Maozu Guo[b], Jun Wang[a,*]

[a] College of Computer and Information Sciences, Southwest University, Chongqing 400715, China
[b] College of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

ARTICLE INFO

Article history:
Received 31 July 2019
Revised 26 October 2019
Accepted 12 December 2019
Available online 8 February 2020

Communicated by Dr. Swagatam Das

Keywords:
Multi-label learning
Incomplete labels
Feature selection
Label compression
Local feature correlation

ABSTRACT

Feature selection can efficiently alleviate the issue of *curse of dimensionality*, especially for multi-label data with multiple features to embody diverse semantics. Although many supervised feature selection methods have been proposed, they generally assume the labels of training data are complete, whilst we only have data with incomplete labels in many real applications. Some methods try to select features with missing labels of training data, they still can not handle feature selection with a large and sparse label space. In addition, these approaches focus on global feature correlations, but some feature correlations are local and shared by a subset of data. In this paper, we introduce an approach called Feature Selection with missing labels based on Label Compression and Local feature Correlation (FSLCLC for short). FSLCLC adopts the low-rank matrix factorization on the sparse sample-label association matrix to compress labels and recover the missing labels in the compressed label space. In addition, it utilizes sparsity regularization and local feature correlation induced manifold regularizations to select the discriminative features. To solve the joint optimization objective for label compression, recovering missing labels and feature selection, we develop an iterative algorithm with guaranteed convergence. Experimental results on benchmark datasets show that the proposed FSLCLC outperforms the state-of-the-art multi-label feature selection algorithms.

1. Introduction

Multi-label learning focuses on data annotated with a set of non-exclusive labels [1,2]. During the past decades, multi-label learning has attracted increasing attention and has been widely applied to various fields, such as text categorization [3], automatic image annotation [4], gene function prediction [5] and so on. Traditional multi-label learning approaches generally assume the training samples have complete labels. However, we cannot obtain data with complete labels in most applications. For instance, protein function prediction is an important bioinformatics task [6]. It is difficult (even impossible) to collect all functional annotations of a protein. In other words, the functional labels of protein are missing. For this reason, protein function prediction were modeled as a weakly-supervised learning problem [7,8]. Weakly-supervised learning mainly cover three scenarios, one is *insufficient* labels, the other is *incomplete* labels, and the third one is *inaccurate* labels. Insufficient labels indicates that the labeled data are not sufficient for training, incomplete labels means the labels of labeled training samples are not complete, while inaccurate labels means the labels of data are not completely accurate, but with some noisy ones. In this paper, we focus on weakly-supervised feature selection with incomplete labels and insufficient labels.

Multi-label learning with incomplete labels has been attracting increasing attention and made some achievements [9–14]. However, the dimensionality of multi-label data, such as texts, proteins and images are usually extreme large. These weak-label learning methods do not always work well under such high-dimensional data. The massive features of multi-label data bring in a heavy computational burden, due to the exponential growth of the required model parameters, which degrade the generalization performance. In other words, these methods suffer from the issue of *curse of dimensionality* [15–18]. In practice, for most tasks, we only need a small subset of original features to train the model, while the other features are irrelevant and redundant. Therefore, multi-label feature selection methods [19,20] were introduced to select the most informative features and to improve the performance of multi-label learning.

However, these feature selection methods are supervised, they ask for sufficient labeled training samples. In practice, we could

[*] Corresponding author.
E-mail addresses: guomaozu@bucea.edu.cn (M. Guo), kingjun@swu.edu.cn (J. Wang).

https://doi.org/10.1016/j.neucom.2019.12.059
0925-2312/© 2020 Elsevier B.V. All rights reserved.

only obtain a large number of unlabeled data and a few labeled ones. It is expensive and even impossible to collect complete labels for training data. To remedy this problem, some feature selection methods [21–23] adopt the idea of semi-supervised learning to leverage labeled and unlabeled data. For instance, Ma et al. [24] took advantage of labeled and unlabeled instances to capture the manifold structure between them. Semi-supervised methods usually assume that the labeled training data with complete labels. As a result, they may be easily trapped by the labeled training data with missing labels. Given that, some researchers tried to select features on multi-label data with missing labels. Zhu et al. [25] proposed multi-label feature selection with missing labels (MLMLFS), which utilizes the linear regression model to simultaneously recover the missing labels and select the most relevant features by $\ell_{2,p}$-norm regularization. MLMLFS assumes that the label space of training data tends to be sparse if the number of labels is large, it directly models the original incomplete label matrix. As a result, MLMLFS can not effectively recover the missing labels. Furthermore, most feature selection methods focus on the global feature correlation to select features. In fact, only a subset of features of a particular group of samples are related with a label. Therefore, the local feature correlations should be considered in feature selection.

To address these aforementioned issues, we introduce an approach called Feature Selection with missing labels based on Label Compression and Local feature Correlation (FSLCLC). FSLCLC uses the low-rank matrix factorization on the sparse label data matrix to compress labels and recover the missing labels. At the same time, it utilizes sparsity regularization and local feature correlation induced manifold regularizations to select the discriminative features. After that, it unifies the label compression, missing labels recovery and feature selection into a joint objective, and develops an iterative algorithm with guaranteed convergence to optimize the objective. Experimental results show that FSLCLC can efficiently select features. The main contributions of our work are summarized as below.

(i) We utilize low-rank matrix factorization, global and local feature induced manifold regularizations to compress the original sparse label space into low-dimension compact labels and to recover missing labels.
(ii) We adopt an $\ell_{2,1}$-norm based spare regularization and a group of local feature induced manifold regularizations to explore the potential relation between the features and labels, and to achieve selecting the most relevant and discriminant features.
(iii) We introduce an iterative optimization procedure with proved convergence to optimize the non-smooth unified objective function with $\ell_{2,1}$-norm regularization. Experimental results on benchmark datasets show that FSLCLC outperforms state-of-the-art related algorithms [22,25–27] and converges within limited iterations.

The rest of this paper is organized as follows. Section 2 reviews multi-label feature selection methods, Section 3 elaborates on the proposed method. The optimization procedure for the unified objective is provided in Section 4. Section 5 gives the experimental results and analysis, and Section 6 concludes the paper.

2. Related works

Feature selection is a classical machine learning task, which has been studied for more than three decades. A comprehensive review of them can be found in [28] and [29]. In this paper, we mainly review feature selection methods target for multi-label data.

Reyes et al. [30] adopted the Pruned Problem Transformation method [31] to create new multi-class dataset from the original multi-label dataset and then used popular ReliefF [32] for feature selection. Lin et al. [20] proposed a feature selection for multi-label classification based on max-dependency and min-redundancy [33], which simultaneously enhances the dependence between the candidate features and labels while reduces the redundancy between the candidate features and the selected features. These filter-based feature selection methods isolate the process of feature selection and classifier training. As a result, the selected features may be not optimal for the follow-up classifier.

Gharroudi et al. [34] proposed three wrapper-based multi-label feature selection methods based on Random Forest (RF), which measures the importance of features based on the predictive performance of RF. Two of the three wrapper methods utilize the Bianary Relevace (BR) or the Label PowerSet (LP) approach to transform the multi-label learning problem into the binary classification or multi-class classification problems. The other method treats each RFPCT (Random Forest Predictive Clustering Tree), a decision tree that predicts multiple target attributes at once, as a base classifier to directly handle the multi-label data. The features selected by wrapper methods extremely interplay with the adopted classifiers and may not generalize well with another classifier. To address this problem, Zhang et al. [35] introduced a feature selection framework that combines the filter and wrapper techniques. In this framework, principal component analysis (PCA) [36] is firstly used to reduce the dimensionality of original data, and then a genetic algorithm [37] is used to select the most appropriate features. However, this framework is still difficult to obtain a better solution due to its slow convergence speed. In addition, the new features projected by PCA (or other feature extractors [38,39]) are less interpretable and straightforward than those directly selected from the original feature space.

Compared with filter and wrapper-based feature selection methods, embedded methods integrate feature selection and model-learning into a joint framework, such that the computational cost can be reduced while a better performance can be pursued. Nie et al. [40] applied joint $\ell_{2,1}$-norm minimization on both the loss function and regularization to select features. Cai et al. [26] considered the geometric structure of feature manifold in the progress of multi-label feature selection. Gu et al. [41] introduced a matrix-variate Normal prior distribution based model to capture the label correlation and to select features by minimizing the label correlation regularized loss of label ranking. Zhang and Wu [42] introduced an approach called multi-label learning with label specific features, which firstly constructs features specific to each label by conducting clustering analysis on its positive and negative instances, and then performs training and testing by querying the clustering results. Liu et al. [43] studied online multi-label group feature selection, which firstly designs a criterion to select feature groups that are important to label set and then considers the feature interaction and feature redundancy to select an optimal feature subset. Melo and Paulheim [44] recently compared the effect of local and global feature selection with binary relevance, they showed that local outperforms global feature selection in terms of classification accuracy, without drawbacks in runtime performance. However, the term of 'local' and 'global' is defined with respect to binary labels and all labels. These multi-label feature selection algorithms assume all the labels of training data are completely available, they usually need abundant labeled data to guarantee the performance. But we can only obtain a large number of unlabeled data and a few labeled data in many tasks. To utilize unlabeled data, some researchers borrow semi-supervised learning techniques to select features. Chang et al. [22] proposed an efficient convex semi-supervised embedded feature selection algorithm without eigen-decomposition and graph construction, which can apply on large-scale datasets. Chang et al. [45] introduced another semi-supervised feature selection framework, which assumes the label correlation contributing to select features. However, these algorithms assume that each labeled instances have complete labels. The full label matrix assumption is impractical and often violated for most scenarios. It is difficult to reliably capture the correlations between features and labels, given the incomplete labels. Given that, MLMLFS [25] recovers the missing labels of training data to select features.

However, it is still a hard challenge to perform feature selection on datasets with a large and sparse label space. Label space dimensionality reduction (LSDR), alike the canonical feature space dimensionality reduction, can be adopted to handle massive labels [46,47]. Inspired by the merit of LSDR, Jian et al. [48] decomposed the label matrix into a low-dimensional space to reduce the negative effects of imperfect label information for multi-label informed feature selection (MIFS). Braytee et al. [27] decomposed the feature and label space into low-dimensional spaces for correlated multi-label feature selection (CMFS). However, MIFS and CMFS may be misled by missing labels of training data, which consequently compromise the performance.

The key to achieve effective embedded feature selection is to capture the relations between features and labels during model learning. We consider that the correlation between features contributes to capture the relations between labels and features. Specially, if two features are similar, they tend to be related with similar labels; and if a group of samples have the same label, some distinct features maybe shared by these samples. In addition, different labels have their own distinct features. As such, the correlation between features should be locally modelled. Given that, our FSLCLC utilizes the local feature correlation to learn the potential relation between the features and labels, and to coordinate the label compression and missing labels recovery, and finally achieves feature selection in a unified model. Ren et al. [49] also employed the local and global structure of data to select features. The term ‘local’ is derived from the neighborhood relationships between samples and the ‘global’ is determined by the farthest distance between samples. In contrast, FSLCLC uses different subets of samples to explore local feature correlations and all samples to capture the global feature correlation.

3. The proposed method

Suppose $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ is the training data matrix, where n is the number of instances and d is the number of features. $\mathbf{Y} \in \{0, 1\}^{n \times c}$ is the observed label matrix with missing labels, where c is the number of distinct labels. For each sample $\mathbf{x}_i$, $\mathbf{y}_i$ is the label vector with respect to c classes. If $\mathbf{x}_i$ belongs to the jth class, $\mathbf{y}_{ij} = 1$, while $\mathbf{y}_{ij} = 0$ indicates it is unknown whether $\mathbf{x}_i$ belongs to the jth class or not.

Embedded feature selection techniques generally try to train a model to capture the relation between the features and labels, and simultaneously achieve the feature selection in the process of model training. The key issue is to design a model to extract the relations between the label space and feature space based on the data matrix $\mathbf{X}$ and the observed label matrix $\mathbf{Y}$. This traditional model is often formulated as:

$$\arg\min_{\mathbf{W},\mathbf{Y}} \left\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\right\|_2^2 + \alpha\Omega(\mathbf{W}) \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{d\times c}$ is the coefficient matrix, it reflects the relation between the feature space and label space. We define $\mathbf{W} = [\mathbf{w}_1; \mathbf{w}_2; \cdots; \mathbf{w}_d]$, where $\mathbf{w}_i$ indicates the i-th row of $\mathbf{W}$. $\Omega(\mathbf{W})$ is a regularization term with respect to $\mathbf{W}$ and α is a trade-off parameter.

To select features, it is popular to choose the $\ell_{2,1}$-norm as the sparse regularization [22,23,40]. The $\ell_{2,1}$-norm with respect to $\mathbf{W}$ is:

$$\Omega(\mathbf{W}) = \|\mathbf{W}\|_{2,1} = \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m} w_{ij}^2} = \sum_{i=1}^{n}\|\mathbf{w}_i\|_2 \tag{2}$$

Compared with the classical LASSO [50], which can obtain $\mathbf{W}$ with sparse entries, the $\ell_{2,1}$-norm regularization can additionally enforce the sparisty of $\mathbf{W}$ in the row-wise, i.e., $\mathbf{w}_i$ is a row vector which quantifies the contribution of the ith feature for each label. In this way, we can select feature subset based on $\mathbf{W}$. Eq. (1) can be formulated as follows:

$$\arg\min_{\mathbf{W},\mathbf{Y}} \left\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\right\|_2^2 + \alpha\|\mathbf{W}\|_{2,1} \tag{3}$$

However, the above model faces a challenge. When the label space is large and some labels are missing, the label matrix becomes more sparse. We observe that the missing labels may terribly distort the original label structure and mislead the correlation between labels. As a result, it is difficult for Eq. (3) to extract the relations between the labels and features based on the observed labels in the original high-dimensional label space. In other words, the huge and sparse label space has an adverse effect on recovering the missing labels and feature selection.

To reduce this effect, we try to compress the observed label matrix and recover the missing labels. Inspired by the matrix factorization in exploring the low-rank label correlation [51] and in weak-label learning [52], we utilize the low-rank matrix factorization technique to compress sparse labels into a low-dimensional space. Particularly, we adopt the nonnegative matrix factorization [53] for its wide usage, easy interpretation and the nonnegative characteristics of the observed label matrix. NMF decomposes the label matrix into two low-rank non-negative matrices $\mathbf{Y} = \mathbf{GV}$, where $\mathbf{G} \in \mathbb{R}^{n\times r}$ is the compressed label matrix of $\mathbf{Y}$ and $r < c$, $\mathbf{V} \in \mathbb{R}^{r\times c}$ aims at encoding label correlations between c labels via the low-dimensional r new semantic labels. In this way, we can reformulate Eq. (3) as follows:

$$\arg\min_{\mathbf{W},\mathbf{G},\mathbf{V},\mathbf{Y}} \left\{ \begin{array}{c} \left\|\mathbf{X}^T\mathbf{W} - \mathbf{G}\right\|_2^2 + \|\mathbf{Y} - \mathbf{GV}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1} \\ s.t.\ \mathbf{G} \ge 0, \mathbf{V} \ge 0 \end{array} \right\} \tag{4}$$

The first term in Eq. (4) is the loss function to capture the loss between the predicted labels and compressed labels. Unlike Eq. (3), $\mathbf{W}$ is reduced to $d \times r$ with much reduced coefficients, which is achieved by the compressed label matrix $\mathbf{G}$. The second term wants to minimize the reconstruction error between the label matrix $\mathbf{Y}$ and the product of $\mathbf{G}$ and $\mathbf{V}$, and to coordinate the feature selection.

To ensure the similar instances have similar compressed labels, we added a manifold regularization term to guide the decomposition of $\mathbf{G}$ as follows:

$$\arg\min_{\mathbf{G}} tr(\mathbf{G}^T\mathbf{LG}) \tag{5}$$

where $tr(\cdot)$ is the matrix trace operator, $\mathbf{L} = \mathbf{D}_s - \mathbf{W}_s$ is the graph Laplacian matrix. $\mathbf{D}_s$ is a diagonal matrix and its diagonal element is the row sum of $\mathbf{W}_s$, which encodes the similarity between pairwise samples and can be quantified in various ways.

The similarity measure is crucial to define a faithful graph to capture the structural relationship between samples, and consequent for graph-based classification [54–56], clustering [57–59] and many other tasks [60–63]. We want to remark that how to optimally quantify the similarity between samples is an open problem [57] and is out of the scope of this paper. In this work, we simply use cosine similarity to measure the similarity between two samples. Eq. (4) is then extended as:

$$\arg\min_{\mathbf{W},\mathbf{Y},\mathbf{G},\mathbf{V}} \left\{ \begin{array}{c} \left\|\mathbf{X}^T\mathbf{W} - \mathbf{G}\right\|_2^2 + \|\mathbf{Y} - \mathbf{GV}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1} \\ +\beta tr(\mathbf{G}^T\mathbf{LG}) \\ s.t.\mathbf{G} \ge 0, \mathbf{V} \ge 0 \end{array} \right\} \tag{6}$$

By minimizing $tr(\mathbf{G}^T\mathbf{L}\mathbf{G})$ and $\|\mathbf{Y}-\mathbf{G}\mathbf{V}\|_2^2$, we can force two samples with high feature similarity having similar compressed labels. In this way, Eq. (6) can recover missing labels of $\mathbf{Y}$ by jointly exploring the latent relationships between observed labels from the compressed label space, and from the feature space.

Local feature correlations may also provide guidance to explore the underlying true labels and select distinct features for individual labels. For example, if some samples have a same label, they all may embody a shared feature subset closely related to the label. In other words, the relevant feature sets are different for different labels [42]. However, Eq. (6) only considers the global feature correlations. Given that, we introduce the local feature correlation based manifold regularization to learn the compressed labels and thus to more credibly recover the missing labels, and consequently to achieve an improved coefficient matrix $\mathbf{W}$. For this purpose, we divide original dataset $\mathbf{X}$ into r groups $\{\mathbf{X}_1, \cdots, \mathbf{X}_g, \cdots, \mathbf{X}_r\}$ by k-means clustering, where $\mathbf{X}_g \in \mathbb{R}^{d\times n_g}$ has n_g samples. We model a subset of discriminative features with respect to a label for samples in $\mathbf{X}_g$. Let $\mathbf{W}_g \in \mathbb{R}^{d\times d}$ be the local feature correlation matrix of group g, we can compute $\mathbf{W}_g$ as follows:

$$[\mathbf{W}_g]_{i,j} = \frac{|\mathbf{x}_g(i,:)\mathbf{x}_g(j,:)^T|}{||\mathbf{x}_g(i,:)||_2^2||\mathbf{x}_g(j,:)||_2^2} \tag{7}$$

where $\mathbf{x}_g(i,:)$ is the i-th row of $\mathbf{X}_g$. In this way, we can obtain r different local feature correlation matrices. These matrices will be utilized to construct r local feature correlation based manifold regularization terms with respect to $\mathbf{W}_g$, which is formulated as follows:

$$\sum_{g=1}^{r}\frac{n_g}{n}\left(\sum_{i,j}||\mathbf{w}_i-\mathbf{w}_j||^2[\mathbf{W}_g]_{ij}\right) \tag{8}$$

Eq. (8) can be further rewritten as:

$$\begin{aligned}\sum_{g=1}^{r}\frac{n_g}{n}\left(\sum_{i,j}||\mathbf{w}_i-\mathbf{w}_j||^2[\mathbf{W}_g]_{ij}\right) &= \sum_{g=1}^{r}\frac{n_g}{n}tr(\mathbf{W}^T(\mathbf{D}_g-\mathbf{W}_g)\mathbf{W})\\ &= \sum_{g=1}^{r}\frac{n_g}{n}tr(\mathbf{W}^T\mathbf{L}_g\mathbf{W})\end{aligned} \tag{9}$$

Based on the above analysis, we formulate the final objective function of FSLCLC as follows:

$$\arg\min_{\mathbf{W},\mathbf{Y},\mathbf{G},\mathbf{V}}\left\{\begin{array}{c}\left\|\mathbf{X}^T\mathbf{W}-\mathbf{G}\right\|_2^2+\|\mathbf{Y}-\mathbf{G}\mathbf{V}\|_2^2+\gamma\sum_{g=1}^{r}\frac{n_g}{n}tr(\mathbf{W}^T\mathbf{L}_g\mathbf{W})\\ +\beta tr(\mathbf{G}^T\mathbf{L}\mathbf{G})+\alpha\|\mathbf{W}\|_{2,1}\\ s.t.\ \mathbf{G}\geqslant 0, \mathbf{V}\geqslant 0\end{array}\right\} \tag{10}$$

It can be observed from Eq. (10) that the compressed label matrix $\mathbf{G}$ is involved with three terms. The first term is used to capture the relation between compressed label matrix $\mathbf{G}$ and $\mathbf{X}$, the second term captures the relation between the compressed label space and original sparse label space, and the third term enforces similar instances having similar outputs in the compressed label space. The coefficient matrix $\mathbf{W}$ also is related to three terms. The first term makes $\mathbf{X}^T\mathbf{W}$ approximate to $\mathbf{G}$ via a least square loss function, the second term forces $\mathbf{W}$ to respect the local feature correlation manifold, and the third term is to enforce a sparse feature selection.

4. Optimization algorithm

There are four variables ($\mathbf{W}$, $\mathbf{G}$, $\mathbf{V}$ and $\mathbf{Y}$) in our unified objective function Eq. (10). The optimization problem is convex but non-smooth because of the $\ell_{2,1}$-norm. Hence, we adopt an iterative algorithm to optimize these variables based on the idea of ADMM [64].

Firstly, we transform our objective function with constraint terms into the Lagrangian function as:

$$\Omega = \left\{\begin{array}{c}tr\left(\mathbf{X}^T\mathbf{W}-\mathbf{G}\right)^T\left(\mathbf{X}^T\mathbf{W}-\mathbf{G}\right)+\gamma\sum_{g=1}^{r}\frac{n_g}{n}tr\mathbf{W}^T\mathbf{S}_g\mathbf{W}\\ +tr(\mathbf{Y}-\mathbf{G}\mathbf{V})^T(\mathbf{Y}-\mathbf{G}\mathbf{V})+\alpha\|\mathbf{W}\|_{2,1}\\ +\beta tr(\mathbf{G}^T\mathbf{L}_s\mathbf{G})+tr(\Phi\mathbf{G}^T)+tr(\Psi\mathbf{V}^T)\end{array}\right\} \tag{11}$$

where $\Phi \geq 0$ and $\Psi \geq 0$ are Lagrangian multiplier matrices for constraints $\mathbf{G} \geqslant 0$ and $\mathbf{V} \geqslant 0$. Then we alternatively fix three variables and optimize the other variable until the convergence.

4.1. Update *W* with *G, V* and *Y* fixed

Due to the non-smoothness of the $\ell_{2,1}$ -norm of $\mathbf{W}$, it is difficult to directly get the solution of $\mathbf{W}$. Fortunately, Nie et al. [40] provided an alternative technique to solve this hard problem. Following this technique, we can compute the partial derivative of $\|\mathbf{W}\|_{2,1}$ with respect to $\mathbf{W}$ as follows:

$$\frac{\partial\|\mathbf{W}\|_{2,1}}{\partial\mathbf{W}} = 2\mathbf{D}\mathbf{W} \tag{12}$$

where $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix and its i-th diagonal element is

$$\mathbf{D}_{ii} = \frac{1}{2\|\mathbf{w}_i\|_2} \tag{13}$$

We obtain the partial derivative of Lagrangian function with respect to $\mathbf{W}$

$$\frac{\partial(\Omega(\mathbf{W}))}{\partial\mathbf{W}} = \mathbf{X}\mathbf{X}^T\mathbf{W}-\mathbf{X}\mathbf{G}+\alpha\mathbf{D}\mathbf{W}+\gamma\sum_{g=1}^{r}\frac{n_g}{n}\mathbf{S}_g\mathbf{W} \tag{14}$$

Letting the partial derivative as zero, we arrive at the solution of $\mathbf{W}$ as:

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^T+\alpha\mathbf{D}+\sum_{g=1}^{r}\frac{\gamma n_g}{n}\mathbf{S}_g)^{-1}\mathbf{X}\mathbf{G} \tag{15}$$

4.2. Update *G* and *V* with *W* and *Y* fixed

Following the techniques of standard NMF [53], we can update $\mathbf{G}$ and $\mathbf{V}$ alternatively with other variables fixed. Particularly, the partial derivative of Lagrangian function with respect to $\mathbf{G}$ becomes

$$\left\{\begin{array}{c}\frac{\partial(\Omega(\mathbf{G}))}{\partial\mathbf{G}} = -2\mathbf{X}^T\mathbf{W}+2\mathbf{G}-2\mathbf{Y}\mathbf{V}^T+2\mathbf{G}\mathbf{V}\mathbf{V}^T\\ +2\beta(\mathbf{D}_s-\mathbf{W}_s)\mathbf{G}+\Phi\end{array}\right\} \tag{16}$$

Then we can utilize the Karush-Kuhn-Tucker (KKT) conditions [64] for the nonnegativity of $\mathbf{G}$ as:

$$(-\mathbf{A}+\mathbf{G}-\mathbf{Y}\mathbf{V}^T+\mathbf{G}\mathbf{V}\mathbf{V}^T+\beta\mathbf{B}-\beta\mathbf{C})_{ip}\mathbf{G}_{ip} = 0 \tag{17}$$

where $\mathbf{A} = \mathbf{X}^T\mathbf{W}$, $\mathbf{B} = \mathbf{D}_s\mathbf{G}$, $\mathbf{C} = \mathbf{W}_s\mathbf{G}$. Let $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$, $\mathbf{B} = \mathbf{B}^+ - \mathbf{B}^-$, $\mathbf{C} = \mathbf{C}^+ - \mathbf{C}^-$, For any matrix $\mathbf{M}$, we define $\mathbf{M}^+ = \frac{|\mathbf{M}_{ij}|+\mathbf{M}_{ij}}{2}$ and $\mathbf{M}^- = \frac{|\mathbf{M}_{ij}|-\mathbf{M}_{ij}}{2}$. Then Eq. (17) can be rewritten as:

$$\begin{aligned}&(-\mathbf{A}^+ + \mathbf{A}^- + \mathbf{G}-\mathbf{Y}\mathbf{V}^T+\mathbf{G}\mathbf{V}\mathbf{V}^T+\beta(\mathbf{B}^+ - \mathbf{B}^-)-\beta(\mathbf{C}^+ + \mathbf{C}^-))_{ip}\\ &\quad\times\mathbf{G}_{ip} = 0\end{aligned} \tag{18}$$

Eq. (18) leads to the following update formula:

$$\mathbf{G}_{ip} = \frac{\mathbf{G}_{ip}(\mathbf{A}^+ + \mathbf{Y}\mathbf{V}^T+\beta(\mathbf{C}^+ + \mathbf{B}^-))_{ip}}{(\mathbf{A}^- + \mathbf{G}+\beta(\mathbf{B}^+ + \mathbf{C}^-)+\mathbf{G}\mathbf{V}\mathbf{V}^T)_{ip}} \tag{19}$$

Similarly, we can obtain the partial derivative of Lagrangian function with respect to $\mathbf{V}$ as follows:

$$\frac{\partial(\Omega(\mathbf{V}))}{\partial\mathbf{V}} = -2\mathbf{G}^T\mathbf{Y}+2\mathbf{G}^T\mathbf{G}\mathbf{V}+\Psi \tag{20}$$

Then we utilize the KKT conditions for the nonnegative of $\mathbf{V}$ as:

$$(-\mathbf{G}^T\mathbf{Y} + \mathbf{G}^T\mathbf{G}\mathbf{V})_{pj}\mathbf{V}_{pj} = 0 \tag{21}$$

We can update the $\mathbf{V}$ by the following formula:

$$\mathbf{V}_{pj} = \frac{\mathbf{V}_{pj}(\mathbf{G}^T\mathbf{Y})}{\mathbf{G}^T\mathbf{G}\mathbf{V}} \tag{22}$$

To make the solution of $\mathbf{G}$ and $\mathbf{V}$ unique [65], the normalized $\mathbf{G}$ and $\mathbf{V}$ can be achieved by:

$$\mathbf{V}_{pj} = \frac{\mathbf{V}_{pj}}{\sqrt{\sum \mathbf{V}_{pj}^2}}, \quad \mathbf{G}_{ip} = \frac{\mathbf{G}_{ip}}{\sqrt{\sum \mathbf{G}_{ip}^2}} \tag{23}$$

4.3. Update Y with the other variables fixed

Finally, we derive the partial derivative of $\mathbf{Y}$ with respect to $\mathbf{Y}$

$$\frac{\partial\Omega(\mathbf{Y})}{\partial\mathbf{Y}} = 2\mathbf{Y} - 2\mathbf{G}\mathbf{V} \tag{24}$$

Then we can update the predicted matrix $\mathbf{Y}$:

$$\mathbf{Y} = \mathbf{G}\mathbf{V} \tag{25}$$

From the above analysis, we observe that the solution of $\mathbf{W}$ depends on $\mathbf{D}$ and $\mathbf{G}$, while $\mathbf{D}$ is related to $\mathbf{W}$. To address this problem, we propose an iterative approach. At first, we initialize $\mathbf{D}$ as an identity matrix, and randomly set $\mathbf{G}$ and $\mathbf{V}$ with elements within [0,1]. Then we can update $\mathbf{W}$ according to initialized $\mathbf{G}$ and $\mathbf{D}$. Next, **D, G** and $\mathbf{V}$ are updated based on the updated $\mathbf{W}$. Finally, we update the label predicted matrix $\mathbf{Y}$. Based on the above analysis, we summarize the procedure of FSLCLC in Algorithm 1.

Algorithm 1 Feature Selection with missing labels based on Label Compression and Local feature Correlation (FSLCLC).

```
Input:
    The training data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$
    The training label matrix with missing labels $\mathbf{Y} \in \mathbb{R}^{n\times c}$
    Parameters $\alpha$, $\beta$, $\gamma$ and the target dimensionality of label space $r$
    $maxIter$, Maximum number of iterations
    $\epsilon$, the threshold for convergence.
Output:
    $\mathbf{W} \in \mathbb{R}^{d\times r}$: feature selection matrix.
Let $t = 0$, $\mathbf{D} \in \mathbb{R}^{d\times d}$ be an identity matrix.
Randomly initialize $\mathbf{G}, \mathbf{V} \in [0, 1]$
$\Omega_t = 0$
Repeat
    Calculate $\mathbf{W}_{t+1} = (\mathbf{XX}^T + \alpha\mathbf{D_t} + \sum_{g=1}^{r} \frac{\gamma n_g}{n}\mathbf{S}_g)^{-1}\mathbf{XG}$.
    Update $\mathbf{G}$ according to Eq. (19)
    Update $\mathbf{V}$ according to Eq. (22)
    Update $\mathbf{Y} = \mathbf{GV}$
    Compute the diagonal matrix $\mathbf{D}_{t+1}$ according to Eq. (13).
    Normalize $\mathbf{G}_{t+1}, \mathbf{V}_{t+1}$ according to Eq. (23).
    $t = t + 1$.
    update $\Omega_t$ using Eq. (11).
    $\theta = \|\Omega_t - \Omega_{t-1}\|$.
Until $t > maxIter$ or $\theta < \epsilon$
Return W
```

Finally, we briefly discuss the computational complexity of FSLCLC. We observe that learning $\mathbf{W}$ needs the calculation of matrix inverse and multiplication. Thus, the computational complexity with respect to $\mathbf{W}$ is $\mathcal{O}(d^3 + d^2n)$. Similarly, the computational complexity of updating $\mathbf{G}$ and $\mathbf{V}$ are $\mathcal{O}(r^2n)$ and $\mathcal{O}(nrc)$, respectively. So the overall complexity of FSLCLC is $\mathcal{O}(d^3 + d^2n + r^2n + nrc)$.

Table 1
Statistics of used datasets for experiments. *Avg* is the average number of labels per sample and *Spr* is the average label density per sample.

Datasets	samples	features	labels	Avg	Spr
Corel5k	4396	1000	206	3.53	0.017
Enron	1702	1001	53	3.38	0.063
Bibtex	7395	1836	159	2.40	0.015
EUR-Lex	19338	5000	201	2.21	0.011
Delicious	14975	500	983	19.97	0.020

5. Experiments

In this section, we conduct experiments on five benchmark multi-lable datasets to validate the performance of FSLCLC and compare its performance with that of other related state-of-the-art multi-label feature selection methods.

5.1. Experimental setup

Five publicly available real-world datasets are used to evaluate the performance of FSLCLC. All these datasets could be obtained from Mulan (http://mulan.sourceforge.net/datasets-mlc.html) and the statistics of them are listed in Table 1. We can observe that the original feature dimension of datasets varies from 500 to 5000, and the number of labels on these dataset varies from 53 to 983. In addition, *Avg* indicates the average number of labels for each sample and *Spr* (*Avg/labels*) indicates the sparse degree of each datset. We can conclude that the label space is sparse for all the datasets, given the pretty small *Spr*.

To comparatively evaluate the performance of our proposed FSLCLC, we consider the our algorithm should be compared with MLMLFS [25], which is the only available method for feature selection with missing labels. Besides, we compare FSLCLC against three supervised and semi-supervised feature selection methods. CMFS [27] is a robust feature selection algorithm which compresses the original label space and feature space at the same time. MSSL is a supervised feature selection method based on feature manifold regularization, it considers the global feature correlation to guide the feature selection [26]. CSFS is a state-of-art method for semi-supervised multi-label feature selection [22]. In addition, we set a Baseline which utilizes original features without feature selection to classify the data.

All these methods can obtain the coefficient matrix $\mathbf{W}$ and we can quantify the importance of each feature by this matrix. After ranking features via the importance of features, we gradually select features as the selected feature subset from the top one to the last one. We set the ratio of selected features from 0.1 to 1 with step size as 0.1. When the ratio is 1, all features are selected, the rate 0.1 indicates only the top 10% ranked features are selected. On the other hand, if the performance of FSLCLC arrives at the peak when we select the top 10%, we further try to select fewer features. For instance, the ratio of selected feature varies from the 0.01 to 0.1 stepped by 0.01. Similarly, we take this strategy to search the suitable dimensionality for label space compression. For all the comparing methods, we select the same ratio of features for comparison. We choose ML-*k*NN [66] as the Baseline to evaluate the performance of these feature selection algorithms. The number of nearest neighbors *k* is set to the default 10, and the input value of smooth *s* is set to the default 1. In the experiments, we fix the input parameters of these comparing algorithms as the authors suggested in the original papers. For our method, we use grid search strategy to study the effect of parameters on the model. The parameters α and β are tuned from 10^{-4} to 10^6 with a fixed value of r. We fixed the suitable parameters to study the effect of r.

To quantify the performance of these multi-label feature selection methods, we adopt five popular multi-label learning metrics [1]: *Average precision, Macro Average F1, Coverage, One-error*, and *Ranking loss*. *Average precision* evaluates the average fraction of labels ranked ahead of a particular relevant label of sample. *Coverage* evaluates how many steps we need, on the average, to go down the list of labels in order to find all the correct labels of the sample. *One-error* evaluates the frequency that top ranked labels do not belong to the sets of correct labels of the sample. *Ranking loss* evaluates the average numbers of irrelevant labels ranked ahead of relevant labels of the sample. *Macro Average F1* firstly calculates the *F1* measure (harmonic mean of precision and recall) of each label and then takes the average *F1* measure of all labels. The formal definitions of these metrics are given in [1,2]. We want to remark that the larger the value of *Average precision* and *Macro Average F1*, the better performance is; while the smaller the value of other metrics, the better the performance is.

5.2. Experimental results analysis

For each of the five datasets, we assume the original label space is complete. Firstly, we randomly divided the data into two parts, 70% of them is the training data, the rest is the testing data. To simulate the training data with missing labels, we mask labels in their label spaces. We set a variable m to show the number of masked labels for each sample. For instance, $m = 3$ means that we randomly mask three labels of all training instances. Specially, if a sample has two labels, we only mask one label to ensure the sample having at least one label. Then we independently and randomly split datasets into the training and testing sets 10 rounds, with randomly masked labels in each round. We report the average results of these methods under each fixed setting in Table 2. In the table, •/○ indicates whether FSLCLC is statistically superior/inferior to the other comparing method. Particularly, we use the widely-used paired samples t-test (at 95% significance level) [67], which assumes that the differences between paired variable are normally distributed and then adopts the average and standard deviation of paired variables (results of multiple independent runs of comparing methods) to study the statistical difference between them.

We can observe that the proposed FSLCLC outperforms other feature selection methods for most of evaluation metrics. Taking the results of these algorithms w.r.t. *Average precision* for example, we have the following observations. When $m = 1$, FSLCLC has 2% average improvement compared to the second best algorithm CMFS on the *Enron* dataset. Compared with the second best approach MSSL, 3% average improvement is obtained by FSLCLC on the *Bibtex*. For dataset *Corel5k*, the gap of performance between FSLCLC and other comparing methods is smaller. FSLCLC only has average 1% improvement than other methods with respect to *Average precision*. These comparisons indicate that FSLCLC can select the most relevant features and improve the performance.

We also observe that our method outperforms Baseline on the *EUR-Lex* and *Delicious*, while other comparing methods fail to do so. The explanation is that the label space of these two datasets is large and more sparse due to the missing labels. It difficult for supervised methods such as MSSL and CMFS, which only utilize limited labels, to select relevant features. Similarly, it is hard to directly recover the missing labels for CSFS and MLMLFS in the sparse label space. Different from these comparing algorithms, FSLCLC tries to recover the missing labels in the compressed label space, and it additionally uses the global and local feature manifold regularizations to coordinate the recovery of missing labels and label compression.

The value of m has a negative effect on the performance of feature selection methods. We observe that when m becomes larger, all feature selection methods perform worse. That is because the number of missing labels increases as the value of m increases. It has adverse influence on evacuating the relation between features and labels in the progress of feature selection. Specially, performance gap becomes smaller when m increases. For instance, when m varies from 1 to 2, the *Average precision* drops by about 3% on *EUR-Lex*. When m varies from 2 to 3, the value drops by less than 1%. That is because when m increases from 2 to 3, fewer labels are missing compared to that from 1 to 2, since most instance mainly have two or three labels, even with a large label space.

Since the number of labels for different samples are different. To comprehensively evaluate the performance, we change the way of masking labels for each sample. Particularly, we mask a fixed ratio of labels for each instance from 0% to 80% stepped by 20%. If a sample has five labels, 20% means one label is masked, and 30% also means one label, but 40% indicates 2 labels randomly masked. The *Average precision* variations under different ratios on three datasets are shown in Fig. 1. Generally, the more labels we mask, the worse performance we have. The missing labels have different effects on feature selection methods on different datasets. For instance, *Average precision* drops quickly as the ratio of masked labels increases on *Corel5k* dataset. However, it drops smoothly for *Bibtex* and *Enron* under the same ratio. That is because *Corel5k* has fewer labels per instance compared to other datasets. When the ratio of masked labels is 60%, the label information becomes less and more sparse. The sparse label information affects the performance of feature selection. FSLCLC still performs better than other comparing methods, but it also manifest a reduced performance.

Overall, FSLCLC performs better than other feature selection methods, no matter how the ratio (number) of masked labels is, since it effectively reduces the impact of missing labels and accounts for sparse label space.

To verify above the explanation, we add a group of experiments under four types of scenarios. In the first scenario, we do not mask any label of training data and utilize the complete labels of them to select features. In the second scenario, we randomly mask one label per instance of the training data to study the impact of incomplete labels. In the third scenario, we randomly and completely mask all the labels of 95% training samples to study the impact of insufficient labels. In the forth scenario, to study the joint impact of insufficient labels and incomplete labels, we firstly randomly mask one label for each of training sample then randomly mask all the labels of 20% training samples. In other words, these 20% samples can be regarded as unlabeled data. The experimental results under these scenarios are shown in Table 3.

From Table 3, we can summarize the following observations. (i) Even the labels of training data are complete, FSLCLC still obtains much higher value of *Average precision* than other comparing methods in the first scenario. That is because FSLCLC additionally considers the local feature correlations, which contribute to select informative features. (ii) By comparing the results of the first scenario with those of other scenarios, all the methods have reduced *Average precision* with incomplete labels or insufficient labels, which corroborate that the label information has a significant impact on effective feature selection. The forth scenario has comparable results with the second and better than the third one, since it uses more labeled data than the third one. (iii) The performance gap between the first scenario and other scenarios for FSLCLC is the smallest, compared with those gaps for other comparing methods. That is because FSLCLC can leverage both labeled and unlabeled data, and account for the missing labels, while MSSL and CMFS can only utilize the labeled training data and assume all the labels are complete. Although MLMLFS can also recover the missing labels, it recovers these labels in the original sparse label space without making concrete use of the global and local feature manifolds. In contrast, FSLCLC makes use both manifolds to seek a

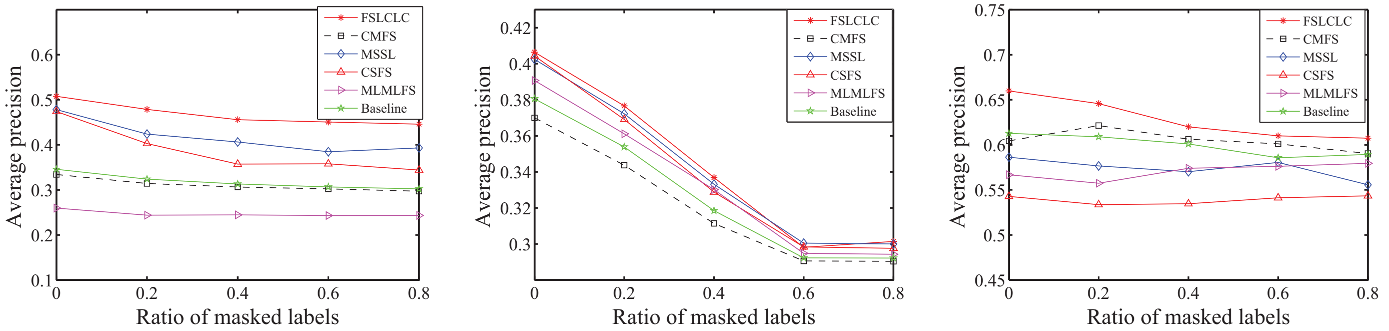
Table 2
Results (mean ± std) of comparing methods on five datasets. For each metric,'↑' means "the larger, the better".'↓' means "the smaller, the better", ●/○ indicates whether FSLCLC is statistically (pairwise *t*-test at 95% significance level) superior/inferior to the comparing method under a particular evaluation metric.

Enron							
Evaluation criterion	*m*	Baseline	CMFS	MSSL	CSFS	MLMLFS	FSLCLC
Average Precision↑	1	0.5577 ± 0.0104●	0.5634 ± 0.0102●	0.5327 ± 0.0155●	0.5343 ± 0.0099●	0.5368 ± 0.0097●	0.5891 ± 0.0110
	2	0.5446 ± 0.0122●	0.5466 ± 0.0126●	0.5202 ± 0.0165●	0.5237 ± 0.0089●	0.5250 ± 0.0209●	0.5668 ± 0.0099
	3	0.5386 ± 0.0120●	0.5372 ± 0.0115●	0.5207 ± 0.0105●	0.5239 ± 0.0104●	0.5206 ± 0.0124●	0.5592 ± 0.0124
Macro Average F1↑	1	0.1095 ± 0.0080●	0.1076 ± 0.0091●	0.0832 ± 0.0044●	0.0753 ± 0.0034●	0.0905 ± 0.0084●	0.1243 ± 0.0085
	2	0.1097 ± 0.0115	0.0987 ± 0.0060●	0.0818 ± 0.0078●	0.0709 ± 0.0023●	0.0884 ± 0.0092●	0.1158 ± 0.0073
	3	0.1030 ± 0.0130	0.0962 ± 0.0110●	0.0835 ± 0.0099●	0.0717 ± 0.0056●	0.0849 ± 0.0051●	0.1135 ± 0.0056
Coverage↓	1	14.0389 ± 0.3801	14.2205 ± 0.3766	14.8751 ± 0.3517●	14.9264 ± 0.3486●	14.7513 ± 0.2796●	13.7810 ± 0.4453
	2	14.3871 ± 0.3980	14.6295 ± 0.3821	15.1131 ± 0.5018●	15.3168 ± 0.3789●	15.0344 ± 0.3942●	14.2611 ± 0.4565
	3	14.6777 ± 0.4034	14.9299 ± 0.3940	15.2147 ± 0.3982●	15.3501 ± 0.4335●	15.2274 ± 0.2661●	14.6836 ± 0.4261
One-error ↓	1	0.3826 ± 0.0202●	0.3761 ± 0.0190●	0.4168 ± 0.0215●	0.4074 ± 0.0192●	0.4168 ± 0.0245●	0.3585 ± 0.0146
	2	0.4070 ± 0.0305●	0.3978 ± 0.0221●	0.4419 ± 0.0323●	0.4192 ± 0.0233●	0.4444 ± 0.0452●	0.3810 ± 0.0234
	3	0.4211 ± 0.0301●	0.4082 ± 0.0308●	0.4554 ± 0.0256●	0.4323 ± 0.0425●	0.4595 ± 0.0442●	0.3781 ± 0.0180
Ranking loss ↓	1	0.1036 ± 0.0046●	0.1047 ± 0.0048●	0.1122 ± 0.0048●	0.1125 ± 0.0050●	0.1106 ± 0.0038●	0.0985 ± 0.0065
	2	0.1071 ± 0.0050●	0.1090 ± 0.0050●	0.1148 ± 0.0066●	0.1165 ± 0.0042●	0.1140 ± 0.0060●	0.1041 ± 0.0054
	3	0.1089 ± 0.0053	0.1118 ± 0.0057●	0.1152 ± 0.0052●	0.1166 ± 0.0054●	0.1152 ± 0.0043●	0.1076 ± 0.0059
Bibtex							
Average precision↑	1	0.2321 ± 0.0051●	0.2235 ± 0.0063●	0.3215 ± 0.0216●	0.2716 ± 0.0125●	0.2196 ± 0.0135●	0.3593 ± 0.0095
	2	0.2151 ± 0.0064●	0.2095 ± 0.0052●	0.2814 ± 0.0149●	0.2354 ± 0.0104●	0.2120 ± 0.0132●	0.3275 ± 0.0082
	3	0.2095 ± 0.0059●	0.2039 ± 0.0051●	0.2689 ± 0.0166●	0.2111 ± 0.0208●	0.2079 ± 0.0134●	0.3180 ± 0.0072
Macro Average F1↑	1	0.1221 ± 0.0044●	0.0851 ± 0.0046●	0.1836 ± 0.0103●	0.1437 ± 0.0049●	0.0949 ± 0.0097●	0.2158 ± 0.0073
	2	0.1094 ± 0.0044●	0.0809 ± 0.0038●	0.1566 ± 0.0096●	0.1189 ± 0.0071●	0.0926 ± 0.0100●	0.1977 ± 0.0059
	3	0.1047 ± 0.0050●	0.0799 ± 0.0031●	0.1458 ± 0.0099●	0.1058 ± 0.0088●	0.0897 ± 0.0083●	0.1930 ± 0.0066
Coverage↓	1	80.3856 ± 1.2707●	77.9751 ± 1.5388●	63.3001 ± 1.0405	68.6504 ± 1.2134●	75.8475 ± 1.6193●	61.3653 ± 1.2954
	2	77.2217 ± 1.4155●	81.7913 ± 1.5353●	71.0854 ± 1.5648●	75.1917 ± 1.1917●	79.3384 ± 1.0064●	67.8842 ± 1.3866
	3	80.7407 ± 1.7322●	82.3652 ± 1.7928●	73.2131 ± 2.0608●	77.6958 ± 2.8423●	79.7064 ± 1.4055●	70.6273 ± 2.0682
One-error↓	1	0.6977 ± 0.0092●	0.7169 ± 0.0101●	0.6093 ± 0.0171●	0.6657 ± 0.0210●	0.7259 ± 0.0257●	0.5736 ± 0.0161
	2	0.7129 ± 0.0155●	0.7242 ± 0.0121●	0.6409 ± 0.0201●	0.7025 ± 0.0181●	0.7286 ± 0.0282●	0.5973 ± 0.0122
	3	0.7140 ± 0.0167●	0.7278 ± 0.0110●	0.6476 ± 0.0202●	0.7399 ± 0.0161●	0.7322 ± 0.0289●	0.5990 ± 0.0144
Ranking loss↓	1	0.3420 ± 0.0084●	0.3453 ± 0.0086●	0.2734 ± 0.0087●	0.3044 ± 0.0078●	0.3366 ± 0.0111●	0.2638 ± 0.0100
	2	0.3545 ± 0.0079●	0.3593 ± 0.0090●	0.3055 ± 0.0097●	0.3279 ± 0.0087●	0.3495 ± 0.0066●	0.2891 ± 0.0068
	3	0.3523 ± 0.0096●	0.3584 ± 0.0109●	0.3120 ± 0.0135●	0.3377 ± 0.0161●	0.3475 ± 0.0078●	0.2969 ± 0.0118
Corel5k							
Average precision↑	1	0.3536 ± 0.0066●	0.3463 ± 0.0069●	0.3703 ± 0.0015●	0.3696 ± 0.0041●	0.3680 ± 0.0047●	0.3771 ± 0.006 3
	2	0.3212 ± 0.0062●	0.3157 ± 0.0075●	0.3330 ± 0.0032●	0.3289 ± 0.0043●	0.3296 ± 0.0029●	0.3361 ± 0.0031
	3	0.2927 ± 0.0054●	0.2893 ± 0.0047●	0.3000 ± 0.0036●	0.3010 ± 0.0049●	0.3000 ± 0.0050●	0.3059 ± 0.0042
Macro Average F1↑	1	0.1187 ± 0.0062●	0.1122 ± 0.0065●	0.1306 ± 0.0065●	0.1311 ± 0.0079●	0.1325 ± 0.0061●	0.1380 ± 0.0057
	2	0.1009 ± 0.0064●	0.0950 ± 0.0080●	0.1099 ± 0.0055	0.1067 ± 0.0072●	0.1068 ± 0.0072●	0.1134 ± 0.0036
	3	0.0744 ± 0.0067●	0.0741 ± 0.0061●	0.0820 ± 0.0038●	0.0833 ± 0.0057●	0.0825 ± 0.0049●	0.0884 ± 0.0050
Coverage↓	1	71.5015 ± 1.3846●	72.0407 ± 1.3523●	70.6733 ± 1.4609	70.6744 ± 1.3782●	70.9207 ± 1.1386●	70.0406 ± 1.1948
	2	79.1269 ± 1.2547●	79.3802 ± 1.2170●	78.4574 ± 1.1582●	78.6568 ± 1.2312●	78.6344 ± 1.1286●	77.9138 ± 1.2839
	3	85.1610 ± 1.0413●	85.3105 ± 1.0639●	85.1376 ± 0.9200●	84.8848 ± 1.0812	85.1373 ± 1.1048	84.8614 ± 1.0026
One-error ↓	1	0.5930 ± 0.0116●	0.5981 ± 0.0142●	0.5662 ± 0.0121	0.5679 ± 0.0081●	0.5713 ± 0.0093●	0.5507 ± 0.0122
	2	0.6243 ± 0.0145●	0.6268 ± 0.0162●	0.6073 ± 0.0111	0.6111 ± 0.0106	0.6083 ± 0.0145	0.6064 ± 0.0064
	3	0.6559 ± 0.0113●	0.6531 ± 0.0122●	0.6369 ± 0.0135●	0.6362 ± 0.0148	0.6382 ± 0.0142	0.6271 ± 0.0092
Ranking loss ↓	1	0.1476 ± 0.0033●	0.1492 ± 0.0031●	0.1443 ± 0.0028●	0.1442 ± 0.0027●	0.1450 ± 0.0021●	0.1422 ± 0.0025
	2	0.1647 ± 0.0026●	0.1660 ± 0.0026●	0.1617 ± 0.0019●	0.1626 ± 0.0020●	0.1625 ± 0.0017●	0.1600 ± 0.0026
	3	0.1775 ± 0.0023●	0.1785 ± 0.0021●	0.1761 ± 0.0014●	0.1757 ± 0.0018●	0.1763 ± 0.0020●	0.1747 ± 0.0021
EUR-Lex							
Average precision↑	1	0.6130 ± 0.0044●	0.5716 ± 0.0087●	0.4605 ± 0.0077●	0.4548 ± 0.0055●	0.4432 ± 0.0058●	0.6218 ± 0.0055
	2	0.5828 ± 0.0066●	0.5450 ± 0.0084●	0.4338 ± 0.0087●	0.4248 ± 0.0074●	0.4221 ± 0.0058●	0.5906 ± 0.0064
	3	0.5753 ± 0.0068●	0.5396 ± 0.0084●	0.4286 ± 0.0075●	0.4198 ± 0.0067●	0.4178 ± 0.0053●	0.5833 ± 0.0079
Macro Average F1↑	1	0.3118 ± 0.0089	0.2701 ± 0.0098●	0.1659 ± 0.0053●	0.1597 ± 0.0056●	0.1551 ± 0.0076●	0.3101 ± 0.0090
	2	0.2841 ± 0.0063●	0.2440 ± 0.0068●	0.1476 ± 0.0063●	0.1388 ± 0.0055●	0.1395 ± 0.0065●	0.2819 ± 0.0084
	3	0.2801 ± 0.0050●	0.2419 ± 0.0065●	0.1441 ± 0.0051●	0.1348 ± 0.0042●	0.1363 ± 0.0067●	0.2789 ± 0.0066
Coverage↓	1	21.0558 ± 0.5532	23.7141 ± 0.5664●	30.8826 ± 0.5935●	30.8026 ± 0.5800●	31.8818 ± 0.6422●	20.8687 ± 0.6482
	2	24.2117 ± 0.6538	26.5605 ± 0.4548●	33.1415 ± 0.6247●	33.5789 ± 0.5595●	33.7956 ± 0.5605●	24.0469 ± 0.6957
	3	24.9635 ± 0.7312	27.1791 ± 0.3933●	33.6195 ± 0.6587●	33.9575 ± 0.5818●	34.1213 ± 0.5636●	24.7803 ± 0.7418
One-error ↓	1	0.3891 ± 0.0050●	0.4287 ± 0.0118●	0.5372 ± 0.0104●	0.5403 ± 0.0064●	0.5501 ± 0.0076●	0.3765 ± 0.0063
	2	0.4103 ± 0.0113●	0.4466 ± 0.0105●	0.5530 ± 0.0098●	0.5603 ± 0.0101●	0.5642 ± 0.0071●	0.4010 ± 0.0084
	3	0.4170 ± 0.0097●	0.4488 ± 0.0116●	0.5562 ± 0.0084●	0.5628 ± 0.0079●	0.5680 ± 0.0063●	0.4055 ± 0.0104
Ranking loss ↓	1	0.0665 ± 0.0025●	0.0760 ± 0.0021●	0.1018 ± 0.0025●	0.1033 ± 0.0022●	0.1050 ± 0.0027●	0.0655 ± 0.0028
	2	0.0747 ± 0.0027●	0.0832 ± 0.0021●	0.1075 ± 0.0028●	0.1082 ± 0.0022●	0.1093 ± 0.0025●	0.0738 ± 0.0025
	3	0.0762 ± 0.0030●	0.0842 ± 0.0019●	0.1083 ± 0.0028●	0.1093 ± 0.0023●	0.1097 ± 0.0024●	0.0753 ± 0.0028
Delicious							
Average precision↑	1	0.3230 ± 0.0027●	0.2746 ± 0.0135●	0.3089 ± 0.0043●	0.3027 ± 0.0035●	0.2547 ± 0.0017●	0.3288 ± 0.0026
	2	0.3200 ± 0.0025●	0.2724 ± 0.0139●	0.3035 ± 0.0033●	0.2991 ± 0.0029●	0.2590 ± 0.0024●	0.3256 ± 0.0025
	3	0.3169 ± 0.0025●	0.2704 ± 0.0136●	0.3003 ± 0.0031●	0.2952 ± 0.0034●	0.2525 ± 0.0016●	0.3226 ± 0.0029
Macro Average F1↑	1	0.1034 ± 0.0020	0.0585 ± 0.0079●	0.0878 ± 0.0022●	0.0841 ± 0.0027●	0.0511 ± 0.0019●	0.1040 ± 0.0015
	2	0.1017 ± 0.0016○	0.0570 ± 0.0077●	0.0851 ± 0.0025●	0.0825 ± 0.0026●	0.0504 ± 0.0019●	0.1017 ± 0.0012
	3	0.1001 ± 0.0014○	0.0561 ± 0.0073●	0.0835 ± 0.0018●	0.0802 ± 0.0023●	0.0502 ± 0.0016●	0.1004 ± 0.0024

(*continued on next page*)

Table 2 (*continued*)

Enron							
Evaluation criterion	*m*	Baseline	CMFS	MSSL	CSFS	MLMLFS	FSLCLC
Coverage↓	1	606.0065 ± 3.7518○	638.7227 ± 6.0479•	625.2808 ± 3.6584•	629.7692 ± 3.7979•	654.6617 ± 2.9751•	608.7609 ± 3.1202
	2	610.8653 ± 3.5807○	642.0510 ± 5.4990•	630.0105 ± 3.5864•	633.4823 ± 3.3993•	656.1009 ± 2.4705•	613.1472 ± 3.2041
	3	615.9203 ± 3.7199○	645.0936 ± 2.4013•	634.1437 ± 3.4401•	637.2374 ± 2.63425•	657.9852 ± 2.9015•	618.1149 ± 3.1157
One-error ↓	1	0.3980 ± 0.0050•	0.4629 ± 0.0226•	0.4242 ± 0.0110•	0.4359 ± 0.0048•	0.5135 ± 0.0055•	0.3868 ± 0.0092
	2	0.4039 ± 0.0077	0.4697 ± 0.0200•	0.4349 ± 0.0084•	0.4415 ± 0.0097•	0.5165 ± 0.0077•	0.3963 ± 0.0067
	3	0.4109 ± 0.0061	0.4750 ± 0.0205•	0.4408 ± 0.0100•	0.4480 ± 0.0086•	0.5202 ± 0.0072•	0.4028 ± 0.0080
Ranking loss ↓	1	0.1277 ± 0.0012	0.1424 ± 0.0035•	0.1343 ± 0.0013•	0.1365 ± 0.0015•	0.1497 ± 0.0010•	0.1270 ± 0.0009
	2	0.1287 ± 0.0011	0.1437 ± 0.0037•	0.1361 ± 0.0012•	0.1377 ± 0.0014•	0.1501 ± 0.0009•	0.1280 ± 0.0008
	3	0.1298 ± 0.0011	0.1441 ± 0.0036•	0.1372 ± 0.0012•	0.1390 ± 0.0014•	0.1506 ± 0.0009•	0.1292 ± 0.0009



(a) Bibtex (b) Corel5k (c) Enron

Fig. 1. Results under different ratios of missing labels on three datasets.

Table 3
Average precision (mean ± std) of comparing algorithms with missing labels and unlabeled data.

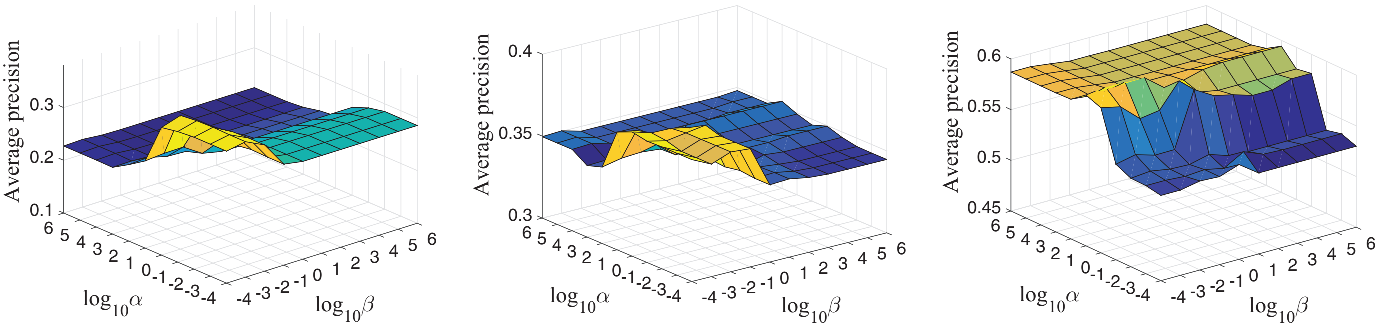
Bibtex						
	Baseline	CMFS	MSSL	CSFS	MLMLFS	FSLCLC
Complete label data	0.3412 ± 0.0079•	0.3270 ± 0.0071•	0.3645 ± 0.0102•	0.4726 ± 0.0079•	0.3124 ± 0.0131•	0.5065 ± 0.0085
Incomplete label data	0.2321 ± 0.0051•	0.2235 ± 0.0063•	0.3215 ± 0.0216•	0.2716 ± 0.0125•	0.2196 ± 0.0135•	0.3593 ± 0.0095
Insufficient label data	0.1853 ± 0.0153•	0.1988 ± 0.0103•	0.2592 ± 0.0070•	0.1784 ± 0.0142•	0.2123 ± 0.0091•	0.2735 ± 0.0106
Incomplete and Insufficient label data	0.2204 ± 0.0061•	0.2219 ± 0.0050•	0.2955 ± 0.0077•	0.2283 ± 0.0151•	0.2849 ± 0.0126•	0.3145 ± 0.0068
Corel5k						
Complete label data	0.3822 ± 0.0027•	0.3761 ± 0.0049•	0.4032 ± 0.0045	0.4032 ± 0.0016	0.3969 ± 0.0035•	0.4065 ± 0.0066
Incomplete label data	0.3536 ± 0.0066•	0.3463 ± 0.0069•	0.3703 ± 0.0015•	0.3696 ± 0.0041•	0.3680 ± 0.0047•	0.3771 ± 0.0063
Insufficient label data	0.2486 ± 0.0036•	0.2448 ± 0.0027•	0.2455 ± 0.0044•	0.2580 ± 0.0075	0.2550 ± 0.0061	0.2503 ± 0.0033
Incomplete and Insufficient label data	0.3415 ± 0.0069•	0.3347 ± 0.0057•	0.3593 ± 0.0054	0.3623 ± 0.0070	0.3605 ± 0.0028	0.3612 ± 0.0057
Enron						
Complete label data	0.6256 ± 0.0154•	0.6187 ± 0.0177•	0.6005 ± 0.0226•	0.5382 ± 0.0103•	0.5726 ± 0.0135•	0.6498 ± 0.0134
Incomplete label data	0.5577 ± 0.0104•	0.5634 ± 0.0102•	0.5327 ± 0.0155•	0.5343 ± 0.0099•	0.5368 ± 0.0097•	0.5891 ± 0.0110
Insufficient label data	0.5036 ± 0.0104•	0.5228 ± 0.0129•	0.5322 ± 0.0066	0.5165 ± 0.0146•	0.5110 ± 0.0040•	0.5232 ± 0.0039
Incomplete and Insufficient label data	0.5510 ± 0.0061•	0.5637 ± 0.0054•	0.5367 ± 0.0019•	0.5295 ± 0.0069•	0.5293 ± 0.0022•	0.5811 ± 0.0081

Table 4
Prediction results (mean ± std) of comparing algorithms on three datasets.

Average precision						
	Baseline	CMFS	MSSL	CSFS	MLMLFS	FSLCLC
Bibtex	0.2321 ± 0.0051•	0.1067 ± 0.0049•	0.3995 ± 0.0085•	0.3525 ± 0.0071•	0.0847 ± 0.0035•	0.4326 ± 0.0095
Corel5k	0.3536 ± 0.0066•	0.1956 ± 0.0138•	0.3026 ± 0.0075•	0.2987 ± 0.0062•	0.1312 ± 0.0047•	0.3939 ± 0.0046
Enron	0.5577 ± 0.0104•	0.4363 ± 0.0320•	0.4703 ± 0.0110•	0.3480 ± 0.0131•	0.3580 ± 0.0149•	0.5693 ± 0.0051

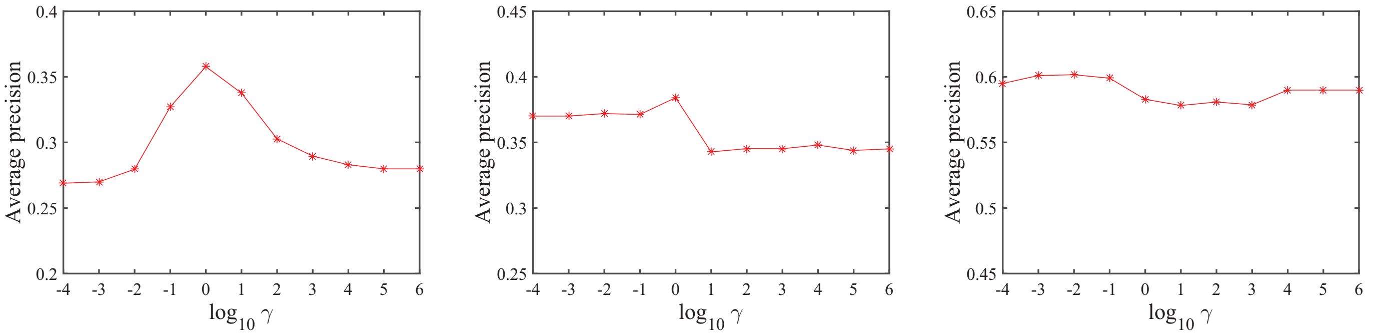
compressed label space and recover missing labels therein. For this reason, FSLCLC obtains better results than MLMLFS.

We want to remark that FSLCLC and other feature selection methods are embedded feature selection methods, which achieve the model learning and feature selection simultaneously. In other words, if an embedded method can select a better subset of features, its prediction performance should be better too. Thus, we utilize the weight coefficient matrix **W** to directly predict the labels of testing data and reveal the experimental results in Table 4. FSLCLC again shows the best performance among these comparing methods, FSLCLC performs well both in the model learning and feature selection. That is because FSLCLC uses both the global and local feature manifolds to compress the label space, which consequently helps to recover the missing labels. For this advantage, it more faithfully captures the relation between features and labels, and thus achieves more prominent performance in feature selection under different scenarios. We note that MLMLFS loses to other comparing methods, although it also replenishes the missing label of training data. The possible cause is that MLMLFS does not well handle a large label space, it recovers missing labels in the original space and mainly focuses on the global feature correlation.



(a) Bibtex (b) Corel5k (c) Enron

Fig. 2. Results under different combinations of α and β on three datasets.



(a) Bibtex (b) Corel5k (c) Enron

Fig. 3. Results under different input values of γ on three datasets.

5.3. Parameter sensitivity analysis

We conduct experiments to study the sensitivity of three parameters α, β and γ on three datasets (*Bibtex, Corel5k* and *Enron*). We fix γ to study the effect of other parameters based on the grid search strategy. When we find an optimal combination of α and β, then we search γ. All the parameters vary from 10^{-4} to 10^{6} with 70% data used for training. For the training data, we randomly mask one label for each instance. Fig. 2 reports the *Average precision* variations under different combinations of α and β.

We can observe that FSLCLC performs worse on the *Bibtex* and *Enron* when α is too large or too small. If the value of α is small, the $\ell_{2,1}$-norm regularization has little effect on the coefficient matrix $\mathbf{W}$, and thus it is difficult to select the more important features. On the other hand, if the value of α is too large, $\mathbf{W}$ is excessively sparse. It is no sense to rank the features according to the too sparse $\mathbf{W}$, whose elements are mostly close to zero. The suitable value of α are 1 or 10 by analyzing these results. β controls the contribution of global feature manifold regularization. We can see that β should not be very large for these datasets. That is because the missing label information is excessively recovered if β is large. On the other hand, if β is too small, it has little effect for recovering missing labels. Overall, FSLCLC obtain a relatively better performance when α within 1 and 10, β within 10^{-2} and 10. An interesting observation is that both a small value of α and β seems better for Bibtex and Corel5k, but not so for Enron. That is because Bibtex and Corel5k have a larger number of samples, labels and features than those of Enron, which needs moderate input values of α and β to sufficiently regularize the compressed labels and coefficient matrix $\mathbf{W}$.

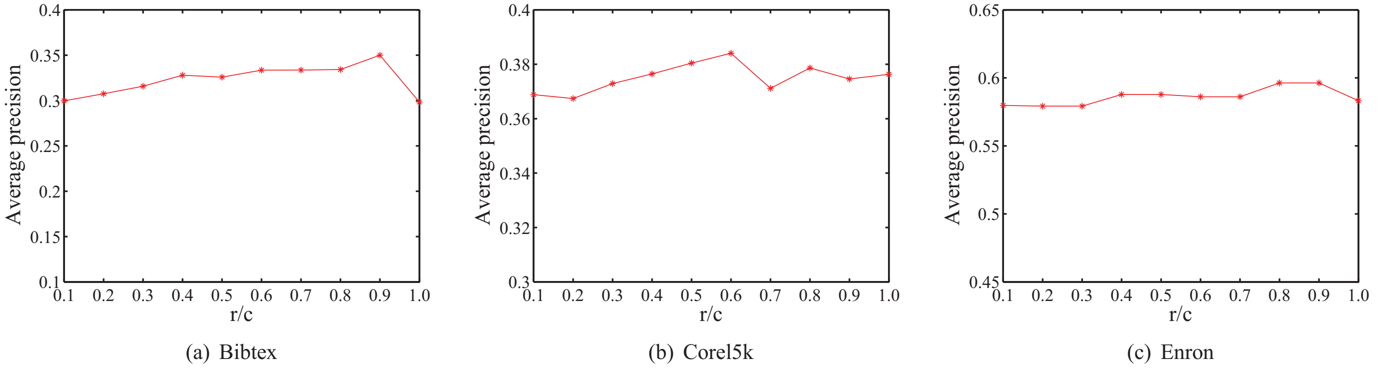
When α and β are fixed to 1 and 10, we show the performance variety under different values of γ in Fig. 3. For *Bibtex* and *Corel5k*, we observe that when the value of γ is smaller than 1, the performance is worse. When γ is 1, FSLCLC arrives at the peak performance. The performance begins to decrease as γ further increase. For *Enron*, a small γ seems better. Since γ adjusts the local feature manifold regularization to capture the relation between the features and compressed label matrix, we can say that local feature manifolds contribute to select relevant features for labels.

5.4. Analysis of label space dimension reduction

In this subsection, we conduct a set of experiments to further investigate the contribution of compressing labels. We observe the performance variation by varying the target dimensionality r (0.1 to 1.0) of compressed labels on these datasets. The average precision variations w.r.t. r are shown in Fig. 4. We observe that FSLCLC achieves better performance in the compressed label space than in the original space, and different datasets have their own optimal target dimensionalities. For example, $r = 0.8c$ (c is the size of orignal label space) is optimal for *Enron*. If r is too small, the compressed labels can not encode original labels and recover the missing labels. For *Bibtex*, FSLCLC obtains the best performance when $r = 0.9c$. In practice, its performance is better than the original labels when $r = 0.1c$. For *Corel5k*, the suitable size of compressed label space is $0.6c$. These results confirm the effectiveness of compressing labels for feature selection.

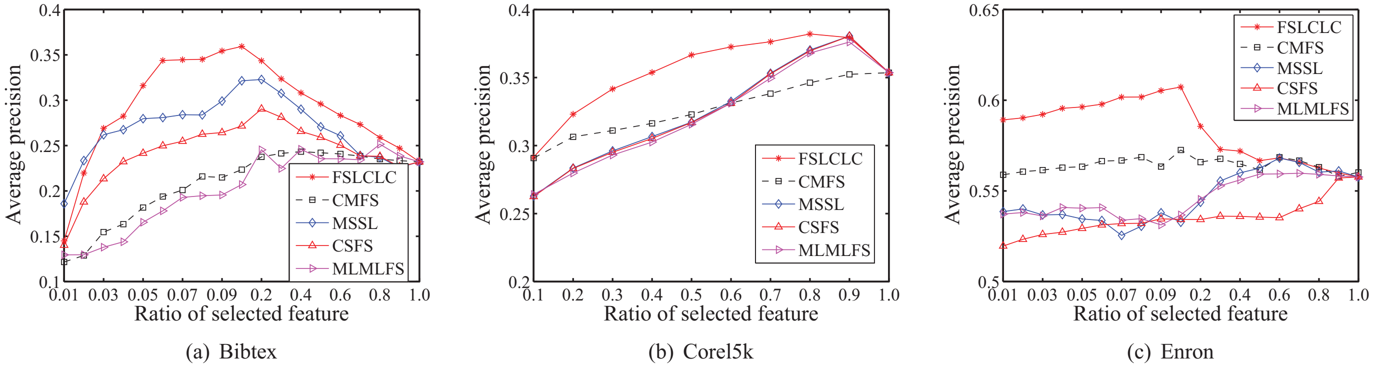
5.5. Impact of selected features

In the previous experiments, all the comparing methods select a fixed number of features. In this subsection, we conduct additional experiments to investigate how the ratio of selected features affect the performance of these algorithms. As shown in Fig. 5, FSLCLC gives the highest *Average precision* values, although it only selects the top 10% features on *Bibtex*. On the other hand, these comparing methods obtain the best performance when the ratio is 20%. As



(a) Bibtex (b) Corel5k (c) Enron

Fig. 4. Results under different ratios (r/c) of compressed labels.



(a) Bibtex (b) Corel5k (c) Enron

Fig. 5. Results under different numbers of selected features on three datasets.

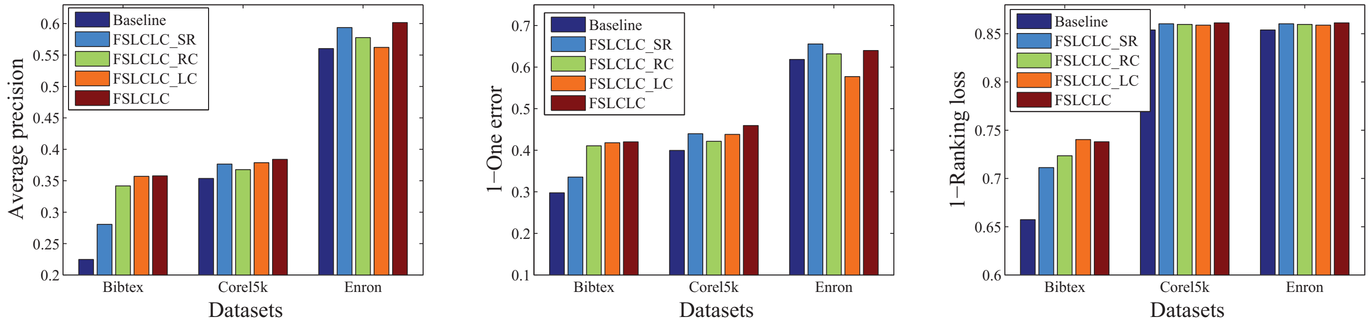


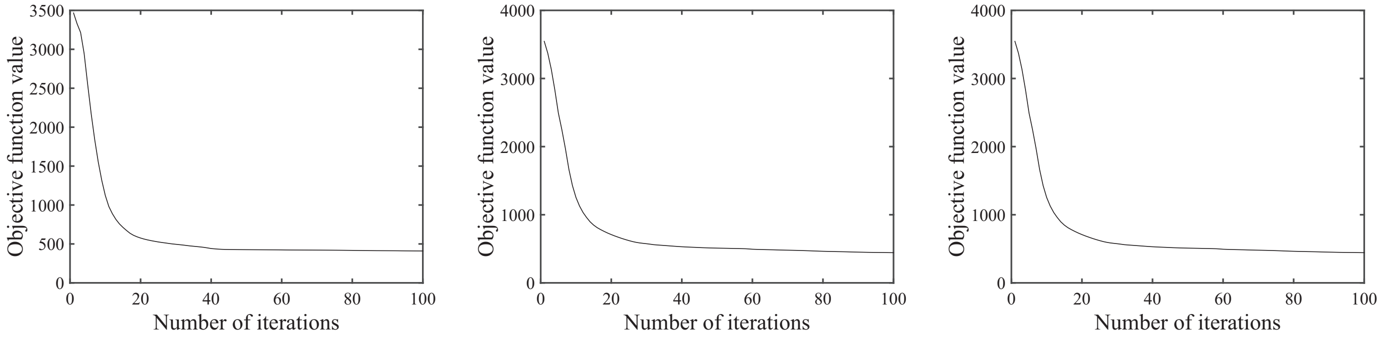
Fig. 6. Results of FSLCLC and its variants on three datasets.

to the *Enron* and *Bibtex* datasets, the performance of FSLCLC generally is superior to other comparing algorithms, no matter what the ratio is. When the ratio is smaller than 10%, *Average Precision* of FSLCLC increases as the ratio rises, and arrives at the peak when the ratio is 10%. *Average precision* drops as the ratio further increases. The reason is that when the ratio is fixed to 10%, most relevant features are selected, and more selected features bring in irrelevant or redundant features. We can observe that *Average precision* is the highest when we select 80% of original features on *Corel5k*. It means that the irrelevant features are fewer for *Corel5k* compared with other datasets. Overall, FSLCLC makes a better performance than these comparing methods even with fewer selected features. This observation suggests that FSLCLC can more credibly identify relevant features than these competitive methods.

5.6. Component analysis

There are three regularization terms in FSLCLC. To investigate whether these regularization terms contribute to the feature selection or not, we introduce three variants of FSLCLC. The first variant (FSLCLC_SR) disregards the local feature manifold regularization (namely $\gamma = 0$). The second variant (FSLCLC_LC) excludes the $\ell_{2,1}$-norm regularization. As for the third variant FSLCLC_RC, it removes the second regularization term of Eq. (10), namely compressed label manifold term is not accounted. Fig. 6 show the results of FSLCLC and its variants on three datasets.

The performance of four methods generally are superior to the Baseline, which indicates that these regularization terms contribute to select features. In addition, we can observe that the $\ell_{2,1}$-norm regularization is more important on the *Enron* dataset, while the local feature manifold regularization is more important on the *Bibtex* dataset. That is because *Enron* has simpler relation between features and labels compared with *Bibtex*. The $\ell_{2,1}$-norm regularization is easier to select informative features on *Enron*. For *Bibtex*, given the more complex relations between the features and labels, it is better to consider the local feature manifold regularizations to extract the relation and select features. Overall, we can observe that FSLCLC generally outperforms other three variants. Although each dataset has different preferences to the components



(a) Bibtex (b) Corel5k (c) Enron

Fig. 7. Convergence curves of the objective function value on three datasets.

of FSLCLC, the integration of these components contributes to a more prominent performance.

5.7. Analysis of convergence

In this section, a group of experiments are conducted to demonstrate that our proposed optimization procedure monotonically decreases the objective function value until convergence. The parameter α, β and γ are all fixed to 1. In addition, we set the maximum number of iterations as 100 for these datasets. The convergence curves on three datasets (*Bibtex,Corel5k* and *Enron*) are plotted in Fig. 7. We observe that all the objective function values on these datasets converge within 60 iterations, which proves that our optimization procedure can converge and obtain the solution within limited iterations.

6. Conclusion and future work

In this paper, we introduce a feature selection method (FSLCLC) with missing labels based on label compressing and local feature correlation. Our method takes advantage of low-rank matrix factorization technique to obtain a compressed label matrix of original sparse labels. In addition, we add a global feature manifold regularization term and a group of local feature manifold regularizations to guide the sought of compressed label matrix and to recover missing labels. At the same time, the least square regression, $\ell_{2,1}$ sparsity term, and these regularization terms are integrated to select informative features. We present an alternative optimization procedure to solve the non-smooth objective function involved with the $\ell_{2,1}$-norm. Extensive experimental results show that FSLCLC significantly outperforms the state-of-art feature selection methods on multi-label data. In our future work, we will study multi-label feature selection on multi-view data.

Declaration of Interest Statement

None Declared.

Acknowledgement

This work is supported by Natural Science Foundation of China (61873214, 61872300 and 61871020), Fundamental Research Funds for the Central Universities (XDJK2020B028 and XDJK2019B024), the National Key Research and Development Plan Task of China (Grant No. 2016YFC0901902), Natural Science Foundation of CQ CSTC (cstc2018jcyjAX0228).

References

[1] M.L. Zhang, Z.H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1819–1837.
[2] E. Gibaja, S. Ventura, A tutorial on multilabel learning, ACM Comput. Surv. 47 (3) (2015) 52.
[3] J. Nam, J. Kim, E.L. Mencía, I. Gurevych, J. Fürnkranz, Large-scale multi-label text classification revisiting neural networks, in: Proceedings of the European Conference on Machine Learning, 2014, pp. 437–452.
[4] J. Weston, S. Bengio, N. Usunier, Wsabie: Scaling up to Large Vocabulary Image Annotation, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 2764–2770.
[5] G. Yu, C. Domeniconi, H. Rangwala, G. Zhang, Z. Yu, Transductive Multi-label Ensemble Classification for Protein Function Prediction, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1077–1085.
[6] G. Yu, H. Zhu, C. Domeniconi, Predicting protein functions using incomplete hierarchical labels, BMC Bioinformat. 16 (1) (2015) 1.
[7] G. Yu, H. Rangwala, C. Domeniconi, G. Zhang, Z. Yu, Protein function prediction with incomplete annotations, IEEE/ACM Trans. Comput. Biol. Bioinf. 11 (3) (2014) 579–591.
[8] Z.H. Zhou, A brief introduction to weakly supervised learning, Natl. Sci. Rev. 5 (1) (2017) 44–53.
[9] Y.Y. Sun, Y. Zhang, Z.H. Zhou, Multi-label learning with weak label, in: Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010, pp. 593–598.
[10] Z. Ma, F. Nie, Y. Yang, J.R. Uijlings, N. Sebe, Web image annotation via subspace-sparsity collaborated feature selection, IEEE Trans. Multimed. 14 (4) (2012) 1021–1030.
[11] S.S. Bucak, R. Jin, A.K. Jain, Multi-label learning with incomplete class assignments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 2801–2808.
[12] B. Wu, Z. Liu, S. Wang, B.G. Hu, Q. Ji, Multi-label learning with missing labels, in: Proceedings of the 22nd International Conference on Pattern Recognition, 2014, pp. 1964–1968.
[13] Q. Tan, Y. Yu, G. Yu, J. Wang, Semi-supervised multi-label classification using incomplete label information, Neurocomputing 260 (2017) 192–202.
[14] Q. Tan, G. Yu, C. Domeniconi, J. Wang, Z. Zhang, Incomplete Multi-view Weak-label Learning, in: Proceedings of International Joint Conference on Artificial Intelligence, 2018, pp. 2703–2709.
[15] R. Bellman, Dynamic programming and lagrange multipliers, Proc. Natl. Acad. Sci. 42 (10) (1956) 767–769.
[16] L. Zhang, L. Jiang, C. Li, G. Kong, Two feature weighting approaches for naive bayes text classifiers, Knowl. Based Syst. 100 (2016) 137–144.
[17] L. Zhang, L. Jiang, C. Li, A new feature selection approach to naive bayes text classifiers, Int. J. Pattern Recognit. Artif. Intell. 30 (02) (2016) 1650003.
[18] L. Jiang, L. Zhang, C. Li, J. Wu, A correlation-based feature weighting filter for naive bayes, IEEE Trans. Knowl. Data Eng. 31 (2) (2019) 201–213.
[19] J. Lee, D.W. Kim, Feature selection for multi-label classification using multivariate mutual information, Pattern Recognit. Lett. 34 (3) (2013) 349–357.
[20] Y. Lin, Q. Hu, J. Liu, J. Duan, Multi-label feature selection based on max-dependency and min-redundancy, Neurocomputing 168 (2015) 92–103.
[21] Y. Liu, F. Nie, J. Wu, L. Chen, Efficient semi-supervised feature selection with noise insensitive trace ratio criterion, Neurocomputing 105 (2013) 12–18.
[22] X. Chang, F. Nie, Y. Yang, H. Huang, A convex formulation for semi-supervised multi-label feature selection, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2014, pp. 1171–1177.
[23] X. Chen, G. Yuan, F. Nie, J.Z. Huang, Semi-supervised Feature Selection via Rescaled Linear Regression, in: Proceedings fo the International Joint Conference on Artificial Intelligence, 2017, pp. 1525–1531.
[24] Z. Ma, F. Nie, Y. Yang, J.R. Uijlings, N. Sebe, A.G. Hauptmann, Discriminating joint feature analysis for multimedia data understanding, IEEE Trans. Multimed. 14 (6) (2012) 1662–1672.
[25] P. Zhu, Q. Xu, Q. Hu, C. Zhang, H. Zhao, Multi-label feature selection with missing labels, Pattern Recognit. 74 (2018) 488–502.
[26] Z. Cai, W. Zhu, Multi-label feature selection via feature manifold learning and sparsity regularization, Int. J. Mach. Learn. Cybern. 9 (8) (2018) 1321–1334.

[27] A. Braytee, W. Liu, D.R. Catchpoole, P.J. Kennedy, Multi-label Feature Selection Using Correlation Information, in: Proceedings of the ACM Conference on Information and Knowledge Management, 2017, pp. 1649–1656.

[28] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowl. Inf. Syst. 34 (3) (2013) 483–519.

[29] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: a review, in: Data Classification: Algorithms and Applications, 2014, p. 37.

[30] O. Reyes, C. Morell, S. Ventura, Scalable extensions of the Relieff algorithm for weighting and selecting features on the multi-label learning context, Neurocomputing 161 (2015) 168–182.

[31] J. Read, A Pruned Problem Transformation Method for Multi-label Classification, in: Proceedings of the New Zealand Computer Science Research Student Conference, 2008, pp. 143–150.

[32] I. Kononenko, Estimating Attributes: Analysis and Extensions of RELIEF, in: Proceedings of the European Conference on Machine Learning, 1994, pp. 171–182.

[33] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 8 (2005) 1226–1238.

[34] O. Gharroudi, H. Elghazel, A. Aussem, A comparison of multi-label feature selection methods using the random forest paradigm, in: Proceedings of the Canadian Conference on Artificial Intelligence, 2014, pp. 95–106.

[35] M.L. Zhang, J.M. Pea, V. Robles, Feature selection for multi-label naive bayes classification, Inf. Sci. (Ny) 179 (19) (2009) 3218–3229.

[36] I. T. Jolliffe, Principal component analysis and factor analysis, Princ. Comp. Anal. (1986) 115–128.

[37] J.H. Holland, et al., Adaptation in natural and artificial systems: an introductory analysis with applications to biology, in: Control, and Artificial Intelligence, MIT Press, 1992.

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

[39] X. He, P. Niyogi, Locality Preserving Projections, in: Advances in Neural Information Processing Systems, 2004, pp. 153–160.

[40] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint 2, 1-norms minimization, Adv. Neural Inf. Process. Syst. (2010) 1813–1821.

[41] Q. Gu, Z. Li, J. Han, Correlated Multi-label Feature Selection, in: Proceedings of the ACM International Conference on Information and Knowledge Msanagement, 2011, pp. 1087–1096.

[42] M.-L. Zhang, L. Wu, Lift: multi-label learning with label-specific features, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2014) 107–120.

[43] J. Liu, Y. Lin, S. Wu, C. Wang, Online multi-label group feature selection, Knowl. Based Syst. 143 (2018) 42–57.

[44] A. Melo, H. Paulheim, Local and global feature selection for multilabel classification with binary relevance, Artif. Intell. Rev. 51 (1) (2019) 33–60.

[45] X. Chang, H. Shen, S. Wang, J. Liu, X. Li, Semi-supervised feature analysis for multimedia annotation by mining label correlation, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2014, pp. 74–85.

[46] D.J. Hsu, S. Kakade, J. Langford, T. Zhang, Multi-label prediction via compressed sensing, in: Advances in Neural Information Processing Systems, 2009, pp. 772–780.

[47] F. Tai, H. Lin, Multilabel classification with principal label space transformation, Neural Comput. 24 (9) (2012) 2508–2542.

[48] L. Jian, J. Li, K. Shu, H. Liu, Multi-label informed feature selection, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 1627–1633.

[49] Y. Ren, G. Zhang, G. Yu, X. Li, Local and global structure preserving based feature selection, Neurocomputing 89 (2012) 147–157.

[50] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodol.) (1996) 267–288.

[51] L. Xu, Z. Wang, Z. Shen, Y. Wang, E. Chen, Learning low-rank label correlations for multi-label classification with missing labels, in: Proceedings of the International Conference on Data Mining, 2014, pp. 1067–1072.

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

[53] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788.

[54] L. Berton, A. De Andra, Graph Construction for Semi-supervised Learning, in: Proceedings of the International Conference on Artificial Intelligence, 2015, pp. 4343–4344.

[55] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, J. You, Semi-supervised classification based on random subspace dimensionality reduction, Pattern Recognit. 45 (3) (2012) 1119–1135.

[56] G. Yu, G. Zhang, Z. Zhang, Z. Yu, L. Deng, Semi-supervised classification based on subspace sparse representation, Knowl. Inf. Syst. 43 (1) (2015) 81–101.

[57] M. Maier, U.V. Luxburg, M. Hein, Influence of graph construction on graph-based clustering measures, in: Advances in Neural Information Processing Systems, 2009, pp. 1025–1032.

[58] Z. Kang, H. Xu, B. Wang, H. Zhu, Z. Xu, Clustering with similarity preserving, Neurocomputing 365 (2019a) 211–218.

[59] Z. Kang, L. Wen, W. Chen, Z. Xu, Low-rank kernel learning for graph-based clustering, Knowl. Based Syst. 163 (2019b) 510–517.

[60] L. Zhang, S. Chen, L. Qiao, Graph optimization for dimensionality reduction with sparsity constraints, Pattern Recognit. 45 (3) (2012) 1205–1210.

[61] G. Yu, H. Peng, J. Wei, Q. Ma, Enhanced locality preserving projections using robust path based similarity, Neurocomputing 74 (4) (2011) 598–605.

[62] Z. Kang, C. Peng, Q. Cheng, Kernel-driven similarity learning, Neurocomputing 267 (1) (2017) 210–219.

[63] Z. Kang, H. Pan, S.C. Hoi, Z. Xu, Robust graph learning from noisy data, IEEE Trans. Cybern. 99 (1) (2019) 1–11.

[64] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[65] W. Xu, X. Liu, Y. Gong, Document Clustering Based on Non-negative Matrix Factorization, in: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 267–273.

[66] M. Zhang, Z. Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

[67] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.

Lin Jiang is an M.Phil. student in the College of Computer and Information Science, Southwest University, Chongqing, China. He received B.Sc. degree in Computer Science from Shandong Normal University, Shandong, China. His current research interests include machine learning and data-mining, especially semi-supervised learning, multi-label learning and feature selection.

Guoxian Yu is a Professor in the College of Computer and Information Science, Southwest University, Chongqing, China. His current research interests include data mining and bioinformatics. He has served as reviewers for AAAI, IJCAI KDD and other prestigious conferences and journals. He is a recipient of Best Poster Award of SDM2012 Doctral Forum and Best Paper Award of 10th IEEE International Conference on Machine Learning and Cybernetics (ICMLC).

Maozu Guo is a professor at the College of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China. He received the Ph.D. degree in Computer Science and Technology from Harbin Institute of Technology. His research interests include bioinformatics, machine learning, and data mining.

Jun Wang is an Associate Professor in the College of Computer and Information Science, Southwest University, Chongqing, China. She received B.Sc. degree in Computer Science, M.Eng. degree in Computer Science and Ph.D. in Artificial Intelligence from Harbin Institute of Technology, Harbin, China in 2004, 2006 and 2010, respectively. Her current research interests include machine learning, data mining and their applications in bioinformatics