Knowledge-Based Systems 187 (2020) 104830

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection☆

Ronghua Shang*,[1], Kaiming Xu, Fanhua Shang[1], Licheng Jiao[2]

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, Shaanxi Province 710071, China

ARTICLE INFO

Article history:
Received 10 January 2019
Received in revised form 28 June 2019
Accepted 2 July 2019
Available online 4 July 2019

Keywords:
Subspace learning
Data manifold
Feature manifold
Inner product regularization term
Feature selection

ABSTRACT

Feature selection can reduce the dimension of data and select the representative features. The available researches have shown that the underlying geometric structures of both the data and the feature manifolds are important for feature selection. However, few feature selection methods utilize the two geometric structures simultaneously in subspace learning. To solve this issue, this paper proposes a novel algorithm, called sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection (SLSDR). Based on the framework of subspace learning-based graph regularized feature selection, SLSDR extends it by introducing the data graph. Specifically, both data graph and feature graph are introduced into subspace learning, so SLSDR preserves the geometric structures of the data and feature manifolds, simultaneously. Consequently, the features which best preserve the manifold structures are selected. Additionally, the inner product regularization term, which guarantees the sparsity of rows and considers the correlations between features, is imposed on the feature selection matrix to select the representative and low-redundant features. Meanwhile, the $l_{2,1}$-norm is imposed on the residual matrix of subspace learning to ensure the robustness to outlier samples. Experimental results on twelve benchmark datasets show that the proposed SLSDR is superior to the six state-of-the-art algorithms from the literature.

1. Introduction

With the rapid development of information technology, high-dimensional data have emerged in the fields of computer vision, pattern recognition and machine learning [1,2]. How to deal with high-dimensional data has become a challenging problem [3]. High-dimensional data often contain noise and redundant features [4,5], and only a small number of features are informative and representative [6]. Noise and redundant features decrease the effectiveness of learning algorithms and increase the time complexity of data processing, so it is necessary to use dimensionality reduction techniques to remove noise and redundant features. Feature extraction and feature selection are two commonly used dimensionality reduction techniques [7,8]. Feature extraction needs to find a projection to map the original high-dimensional data to a low-dimensional subspace [9,10], and feature selection is to select an optimal feature subset to obtain a compact data representation [11,12]. In feature selection, the representative features are selected, so the dimension of the features is significantly reduced, making data processing more efficient [13]. Additionally, since noise and redundant features are removed, the accuracy of clustering and classification tasks is improved. Compared with feature extraction, feature selection keeps the original representation of the features, so the semantic information of the original features can be preserved [14,15]. Benefiting from the advantages of feature selection, many feature selection algorithms have been proposed recently.

Based on whether the supervised information is used or not, feature selection methods can be divided into three categories: supervised, semi-supervised and unsupervised [16,17]. Supervised feature selection methods select features based on the correlations between training samples and class labels [18,19]. In semi-supervised feature selection methods, a small number of labeled training samples are required, and the unlabeled and labeled training samples are combined together to improve the performance of feature selection algorithms [20–22]. However, the large-scale data obtained in the practical applications is unlabeled, and marking the unlabeled data can bring high time cost. Unsupervised feature selection methods require no label information of samples and only make use of the intrinsic structure of data to select features [23,24]. Therefore, it is particularly important to develop some efficient unsupervised feature selection

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.07.001.

* Corresponding author.
E-mail address: rhshang@mail.xidian.edu.cn (R. Shang).
1 Member, IEEE.
2 Fellow, IEEE.

https://doi.org/10.1016/j.knosys.2019.07.001
0950-7051/

algorithms. Unsupervised feature selection algorithms include filter, wrapper and embedded methods [25–27]. The filter methods do not depend on any learning algorithm and only use the intrinsic properties of data to select the important features [25]. The wrapped methods use a specific learning algorithm to select the feature subset that makes the learning algorithm perform best [28]. For the embedded methods, feature selection is completed in the learning process of the corresponding learning algorithm [29]. Among the three kinds of methods, the filter methods have the advantages of high calculation speed and great scalability. The wrapped methods can make the learning algorithm achieve the best performance, but these methods have high computational cost. The embedded methods make full use of the advantages of the filter and wrapped methods, trying to select a better feature subset while maintaining low computational cost. The embedded methods have many advantages and have attracted more and more attention of researchers [30].

Subspace learning can reduce the dimension of data effectively and obtain the low-dimensional representation of high-dimensional space. Benefiting from the application of matrix decomposition strategy, subspace learning has been applied from feature extraction to feature selection. In [31], Wang et al. proposed subspace learning for unsupervised feature selection via matrix factorization (MFFS). This algorithm applies the matrix decomposition strategy to subspace learning, thus the unsupervised feature selection problem is treated as a matrix decomposition problem. Then, Wang et al. [32] proposed unsupervised feature selection via maximum projection and minimum redundancy (MPMR). MPMR presents the estimation degree of the selected feature subset by all the features, and the minimized redundancy regularization term is used to ensure the low redundancy of the selected features. In [33], subspace learning-based graph regularized feature selection (SGFS) was proposed. SGFS constructs the feature graph to preserve the geometric structure information of the feature manifold. The above three algorithms achieve good feature selection results, but these algorithms still have some shortcomings. For MFFS and MPMR, they ignore the local geometric information of the data manifold and the feature manifold. Although SGFS considers the local geometric information of the feature manifold, it neglects the structure information of the data manifold. In contrast, the proposed SLSDR takes the local geometric information of both the data manifold and the feature manifold into account, which significantly improves the performance of feature selection.

Researches show that the manifold structure of data can improve the learning efficiency of algorithms [34,35]. And many manifold learning algorithms have been proposed to preserve the intrinsic structure of data, such as Local Linear Embedding (LLE) [36], Laplacian Eigenmap (LE) [37] and Locality Preserving Projections (LPP) [38]. All these learning algorithms preserve the manifold information of the high-dimensional space into the low-dimensional embedding. In order to make use of manifold information, many feature selection methods preserving the intrinsic geometric structure of data have been proposed. For Laplacian Score (LapScor) [28], the local geometric information contained in data is used to calculate the score for each feature separately. For spectral feature selection (SPEC) [39], it is based on spectral graph theory and establishes a unified framework for feature selection. Yang et al. [13] proposed $l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning (UDFS). For this algorithm, the local discriminant model is used to guide the feature selection, and the local geometric structure of the data manifold is also considered. For multi-clustering feature selection (MCFS) [18], it uses spectral analysis and the l_1-norm to select features, and the features that retain the multi-clustering structure are selected. For joint embedding learning and sparse regression (JELSR) [34], it adopts a one-step strategy, which combines embedding learning and sparse regression to perform feature selection. Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection is proposed in [4]. This algorithm is based on the framework of JELSR and constructs graphs on the data manifold and the feature manifold simultaneously for feature selection, so a better feature selection effect is achieved.

The manifold structure information has also been widely used in classification problems. Ye et al. [40] proposed weighted twin support vector machines with local information (WLTSVM) and its application. WLTSVM makes full use of the potential similarity information between samples to achieve higher classification accuracy. Xu et al. [41] proposed a KNN-based weighted rough v-twin support vector machine (Weighted rough v-TSVM). The Weighted rough v-TSVM not only considers different penalties for negative samples, but also takes into account the local structure information of positive samples, which enhances the effectiveness of the proposed algorithm. The k-nearest neighbor based structural twin support vector machine algorithm (KNN-STSVM) was proposed in [42]. KNN-STSVM makes full use of the KNN method and imposes different weights on different samples within the class to make full use of the structural information contained in data, thus improving the classification accuracy of the algorithm.

Many evolutionary-based feature selection methods have been proposed recently. Das et al. [43] proposed an ensemble feature selection method using a bi-objective genetic algorithm. In this algorithm, several data subsets are generated by stratified sampling and the genetic algorithm based feature selectors are used to produce non-dominated feature subsets. Then the dominance based method yields the final feature subset. Zhang et al. [44] proposed a multi-objective particle swarm optimization (MOPSO) approach for cost-based feature selection. And the feature subsets to be used are considered as the generated Pareto front. Then the probability-based encoding technology, the crowding distance, and the external archive are used to improve the searching ability of the algorithm. The method of feature selection of unreliable data using an improved multi-objective PSO algorithm was proposed in [45]. And the reinforced memory and hybrid mutation strategies are used to improve the performance of the algorithm.

Recently, several safe feature screening rules were proposed to remove the inactive features before learning tasks. In [46], Xu et al. proposed E-ENDPP: a safe feature selection rule for speeding up Elastic Net. By using this rule, the inactive features can be identified and deleted before they are trained, so the problem scale can be reduced to speed up the training process. Meanwhile, E-ENDPP can achieve the same solution as the original model because this rule is safe. Pan et al. [47] proposed a safe reinforced feature screening strategy for lasso. Based on enhanced screening rule via Dual Polytope Projection (EDPP) and feasible solutions, the proposed algorithm can improve training efficiency of lasso for large-scale datasets. The safe screening rules are also applied to classification models. Pan et al. [48] proposed safe screening rules for accelerating twin support vector machine classification. In this model, the safe screening rule (SSR) and modified SSR (MSSR) for twin support vector machine (TSVM) are proposed. As a result, a large number of training samples are deleted and the scale of the TSVM problem is reduced. For SLSDR, it makes full use of the manifold information in dual graph and selects the representative and low-redundant features by using the inner product regularization term. Therefore, SLSDR can achieve a good feature selection effect.

In this paper, a novel algorithm is proposed, called sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection (SLSDR). SLSDR is based on the

framework of subspace learning-based graph regularized feature selection (SGFS). For SGFS, the nearest neighbor graph of the feature manifold is constructed, and the $l_{2,1}$-norm is imposed on the feature selection matrix $\boldsymbol{S}$ to guide feature selection. On the basis of SGFS, SLSDR introduces the data graph for the data manifold and uses the local geometric information of the data manifold and the feature manifold simultaneously to guide the subspace learning. The linear transformation matrix $\boldsymbol{Z}$ and the low-dimensional embedding matrix $\boldsymbol{Y}$ directly guide the learning of the feature selection matrix $\boldsymbol{S}$ and the coefficient matrix $\boldsymbol{V}$, respectively. So the features which best preserve the manifold structures can be selected. Moreover, the dual graph used in SLSDR can be easily transplanted to other feature selection algorithms, which can effectively improve the performance of the algorithms. Additionally, for the proposed SLSDR, the inner product regularization term is used to replace the $l_{2,1}$-norm that imposed on the feature selection matrix $\boldsymbol{S}$. The $l_{2,1}$-norm guarantees the sparsity of rows of $\boldsymbol{S}$ to select the representative features, but ignores the correlations between features. So the high-redundant features are selected, which decreases the performance of the feature selection algorithms. The inner product regularization term consists of l_1-norm and l_2-norm, ensures the sparsity of the rows of the feature selection matrix $\boldsymbol{S}$ and considers the correlations between features, so the representative and low-redundant features are selected to obtain a better feature selection result. Meanwhile, the outlier samples are often included in the real world datasets, which seriously affect the effectiveness of the algorithms. So the $l_{2,1}$-norm is imposed on the residual matrix of subspace learning to ensure the robustness to outlier samples. Then, an alternating iterative optimization mechanism is used to optimize the objective function. The evaluation values of different features are calculated according to the feature selection matrix $\boldsymbol{S}$, then the most representative features are selected. Finally, the proposed SLSDR is compared with six other algorithms on twelve benchmark datasets, and the experimental results show that the proposed SLSDR has better performance.

The novelties and contributions are highlighted as follows:

(1) Based on the existing feature graph, the data graph is introduced into the framework of subspace learning. As a result, the local geometric information of both the data manifold and the feature manifold can be preserved. So the features which best preserve the manifold structure can be selected.

(2) The inner product regularization term is used to constrain the feature selection matrix $\boldsymbol{S}$. Since the inner product term ensures the sparsity of rows of $\boldsymbol{S}$ and considers the correlations between features, the proposed SLSDR can select the representative and low-redundant features to get a compact and clear representation of the original data.

(3) The $l_{2,1}$-norm is imposed on the residual matrix of subspace learning. As the Frobenius norm is sensitive to outlier samples, $l_{2,1}$-norm is used to guarantee the robustness of SLSDR to outlier samples.

The remainder of this paper is organized as follows. In Section 2, the proposed SLSDR, the iterative update rules, computational complexity analysis and convergence analysis are presented. In Section 3, the experimental results of SLSDR and other algorithms are provided. Then Section 4 gives the summary of the whole paper.

2. The proposed method

This section presents the framework of the proposed SLSDR algorithm. SLSDR can be divided into three main parts: sparse and low-redundant subspace learning, manifold structure preservation and feature evaluation. Then the iterative update formulas, computational complexity analysis and convergence analysis of SLSDR are provided.

2.1. Related notations

Some related notations to be used in this paper are first introduced. Here, scalars, vectors and matrices are denoted as lowercase letters, bold lowercase letters and bold uppercase letters, respectively. $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \Re^{m\times n}$ denotes a data matrix, where m is the number of features of each sample and n is the total number of samples of the data matrix. $\boldsymbol{x}_i \in \Re^m$ is the ith sample of $\boldsymbol{X}$ and is located in the ith column. For a square matrix $\boldsymbol{A}$, $Tr(\boldsymbol{A})$ represents the trace of $\boldsymbol{A}$. The f_p-norm of vector $\boldsymbol{x} \in \Re^m$ is defined as:

$$\|\boldsymbol{x}\|_p = \left(\sum_{j=1}^{m} |x_j|^p\right)^{\frac{1}{p}} \tag{1}$$

where x_j is the jth element of the vector $\boldsymbol{x}$. For a matrix $\boldsymbol{X} \in \Re^{m\times n}$, its l_r-norm and $l_{r,t}$-norm are defined as follows:

$$\|\boldsymbol{X}\|_r = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |\boldsymbol{X}_{ij}|^r\right)^{\frac{1}{r}} \tag{2}$$

$$\|\boldsymbol{X}\|_{r,t} = \left(\sum_{i=1}^{m}\left(\sum_{j=1}^{n} |\boldsymbol{X}_{ij}|^r\right)^{\frac{t}{r}}\right)^{\frac{1}{t}} \tag{3}$$

When $p = 2$, the f_2-norm of a vector can be obtained. In this paper, the f_2-norm of row vectors of the feature selection matrix $\boldsymbol{S}$ is used to evaluate the importance of different features. When $r = 1$ or $r = 2$, $t = 1$, the l_1-norm, l_2-norm and $l_{2,1}$-norm of matrices, which are commonly used in this paper, can be obtained. And the combination of l_1-norm and l_2-norm imposed on the feature selection matrix $\boldsymbol{S}$ composes the inner product regularization term. The $l_{2,1}$-norm used to constrain the residual matrix of subspace learning guarantees the robustness of SLSDR to outlier samples [49–51]. The l_2-norm of a matrix is also called the Frobenius norm.

2.2. Sparse and low-redundant subspace learning

2.2.1. Subspace learning

In [31], Wang et al. proposed MFFS from the viewpoint of subspace distance. MFFS uses the subspace spanned by the selected feature subset to characterize the space spanned by all the features, and the feature selection is completed in this process. The feature selection problem can be expressed as follows:

$$\begin{aligned} &\arg\ \min_{\boldsymbol{S},\boldsymbol{V}} \left\|\boldsymbol{X}^T - \boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_2^2 \\ &s.t.\, \boldsymbol{S} \geq 0, \boldsymbol{V} \geq 0, \boldsymbol{S}^T\boldsymbol{S} = \boldsymbol{I}_l \end{aligned} \tag{4}$$

where $\boldsymbol{X} \in \Re^{m\times n}$ is the original data matrix, and $\boldsymbol{V} \in \Re^{l\times m}$ is the coefficient matrix for reconstruction. $\boldsymbol{I}_l \in \Re^{l\times l}$ is an identity matrix, and l represents the number of selected features. $\boldsymbol{S} \in \Re^{m\times l}$ is the feature selection matrix and only contains 0–1 elements. If I represents the index set of the selected features, the definition of $\boldsymbol{S}$ is given as follows:

$$\boldsymbol{S}_{i,j} = \begin{cases} 1, & \text{the } j\text{th element of } I \text{ is } i, \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

There are l elements in I, indicating that the total number of the selected features is l.

2.2.2. Sparse subspace learning

However, the definition of $\boldsymbol{S}$ is too strict to meet the requirements. It is difficult to select the most representative features by using only the non-negative constraints and the orthogonal constraint, so the performance of feature selection is reduced. To solve this problem, a sparse subspace learning framework is proposed in [33], as follows:

$$\begin{aligned} &\arg \min_{\boldsymbol{S},\boldsymbol{V}} \left\|\boldsymbol{X}^T - \boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_2^2 + \beta \left\|\boldsymbol{S}\right\|_{2,1} \\ &s.t. \boldsymbol{S} \geq 0, \boldsymbol{V} \geq 0, \boldsymbol{S}^T\boldsymbol{S}=\boldsymbol{I}_l \end{aligned} \tag{6}$$

where β is a balance parameter, and $\beta > 0$. Compared with MFFS, the $l_{2,1}$-norm is additionally imposed on $\boldsymbol{S}$ in this framework, so $\boldsymbol{S}$ is much closer to the matrix defined in Eq. (5) in the learning process. Meanwhile, the $l_{2,1}$-norm guarantees the sparsity of the rows of the matrix $\boldsymbol{S}$, so the representative features can be selected.

2.2.3. Sparse and low-redundant subspace learning

Since the $l_{2,1}$-norm ignores the correlations between features, there are some problems when using this norm to deal with the informative but high-redundant features. When the $l_{2,1}$-norm is used, high-redundant features are selected, reducing the effectiveness of feature selection. In [6], Han et al. proposed a new regularization term, which consists of the sum of the absolute values of the inner product of different feature weight vectors. And in this paper, it is called the inner product regularization term. This regularization term has the form of a combination of l_1-norm and l_2-norm of the feature selection matrix $\boldsymbol{S}$. By using this novel regularization term, the sparsity of rows of the matrix $\boldsymbol{S}$ can be guaranteed, so the representative features are selected. Additionally, the correlations between features are considered, so the low-redundant features can be selected, which improves the effectiveness of feature selection [6]. Therefore, the inner product regularization term is used to replace the $l_{2,1}$-norm imposed on the matrix $\boldsymbol{S}$. The inner product regularization term can be written as follows:

$$\begin{aligned} \Omega(\boldsymbol{S}) &= \sum_{i=1}^{m}\sum_{j=1,j\neq i}^{m} \left|\langle \boldsymbol{s}_i, \boldsymbol{s}_j\rangle\right| \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} \left|\langle \boldsymbol{s}_i, \boldsymbol{s}_j\rangle\right| - \sum_{i=1}^{m} \left|\langle \boldsymbol{s}_i, \boldsymbol{s}_i\rangle\right| \\ &= \left(\left\|\boldsymbol{S}\boldsymbol{S}^T\right\|_1 - trace\left(\boldsymbol{S}\boldsymbol{S}^T\right)\right) \\ &= \left(\left\|\boldsymbol{S}\boldsymbol{S}^T\right\|_1 - \left\|\boldsymbol{S}\right\|_2^2\right) \end{aligned} \tag{7}$$

where $\boldsymbol{s}_i$ represents the ith row of the matrix $\boldsymbol{S}$. If $\Omega(\boldsymbol{S})$ is kept small enough during the optimization process of the proposed SLSDR, the invalid and redundant features can be removed. $\Omega(\boldsymbol{S})$ is then applied to the framework of subspace learning, and the obtained sparse and low-redundant subspace learning framework is as follows:

$$\begin{aligned} &\arg \min_{\boldsymbol{S},\boldsymbol{V}} \left\|\boldsymbol{X}^T - \boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_2^2 + \beta\Omega\left(\boldsymbol{S}\right) \\ &s.t. \boldsymbol{S} \geq 0, \boldsymbol{V} \geq 0, \boldsymbol{S}^T\boldsymbol{S}=\boldsymbol{I}_l \end{aligned} \tag{8}$$

By substituting Eq. (7) into Eq. (8), Eq. (8) can be rewritten as follows:

$$\begin{aligned} &\arg \min_{\boldsymbol{S},\boldsymbol{V}} \left\|\boldsymbol{X}^T - \boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_2^2 + \beta\left(\left\|\boldsymbol{S}\boldsymbol{S}^T\right\|_1 - \left\|\boldsymbol{S}\right\|_2^2\right) \\ &s.t. \boldsymbol{S} \geq 0, \boldsymbol{V} \geq 0, \boldsymbol{S}^T\boldsymbol{S}=\boldsymbol{I}_l \end{aligned} \tag{9}$$

2.3. Manifold structure preserving

The data distributed in high-dimensional space often contains important local information, and making full use of the potential local information can improve the learning efficiency of the algorithms [33]. By using spectral graph theory, the high-dimensional data can be embedded into the low-dimensional subspace, during which the local structure information can be preserved [4]. Based on the advantages of spectral graph theory, it can be applied to feature selection. Recent studies have shown that local structure information is included in both the feature manifold and the data manifold [52,53], thus the potential local information of the feature and data manifolds can be preserved to improve the performance of feature selection.

First, a k-nearest neighbor graph $\boldsymbol{G_0}$=(V_0, E_0) is constructed to model the geometric structure of the feature manifold efficiently, where V_0 is the vertex set $\{\boldsymbol{X}_{1,:}, \boldsymbol{X}_{2,:}, \ldots, \boldsymbol{X}_{m,:}\}$, and E_0 denotes the weights of the edges connecting different vertices. For each vertex, it denotes a feature of the data matrix. In this paper, the Gaussian function is adopted as the measurement of the weights and its expression is as follows:

$$\left[\boldsymbol{W}^{\boldsymbol{V}}\right]_{ij} = \begin{cases} \exp\left(-\left\|\boldsymbol{X}_{i,:} - \boldsymbol{X}_{j,:}\right\|_2^2/\sigma^2\right), & \text{if } \boldsymbol{X}_{i,:} \in N\left(\boldsymbol{X}_{j,:}\right) \\ & \text{or } \boldsymbol{X}_{j,:} \in N\left(\boldsymbol{X}_{i,:}\right) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $i, j = 1,2, \ldots, m$, and $\boldsymbol{X}_{i,:}$ is located in the ith row of the data matrix and represents the ith feature. $N(\boldsymbol{X}_{i,:})$ denotes the k-nearest neighbor set of the feature $\boldsymbol{X}_{i,:}$, and σ denotes the Gaussian scale parameter. $[\boldsymbol{W}^{\boldsymbol{V}}]_{ij}$ measures the similarity of the features $\boldsymbol{X}_{i,:}$ and $\boldsymbol{X}_{j,:}$, and the larger the value of $[\boldsymbol{W}^{\boldsymbol{V}}]_{ij}$, the higher the similarity between the ith and jth features. Thus, $\boldsymbol{W}^{\boldsymbol{V}}$ is regarded as the similarity matrix. The graph Laplacian matrix of the feature manifold is $\boldsymbol{L}^{\boldsymbol{V}} = \boldsymbol{D}^{\boldsymbol{V}} - \boldsymbol{W}^{\boldsymbol{V}}$, where $\boldsymbol{D}^{\boldsymbol{V}}$ is a diagonal matrix and $[\boldsymbol{D}^{\boldsymbol{V}}]_{ii} = \Sigma_j[\boldsymbol{W}^{\boldsymbol{V}}]_{ij}$.

After that, a k-nearest neighbor graph $\boldsymbol{G_1}$=(V_1, E_1) is constructed for the data manifold, where V_1 denotes the vertex set $\{\boldsymbol{X}_{:,1}, \boldsymbol{X}_{:,2}, \ldots, \boldsymbol{X}_{:,n}\}$, and each vertex represents a sample of the data matrix. The Gaussian function of the data manifold is defined as follows:

$$\left[\boldsymbol{W}^{\boldsymbol{S}}\right]_{ij} = \begin{cases} \exp\left(-\left\|\boldsymbol{X}_{:,i} - \boldsymbol{X}_{:,j}\right\|_2^2/\sigma^2\right), & \text{if } \boldsymbol{X}_{:,i} \in N\left(\boldsymbol{X}_{:,j}\right) \\ & \text{or } \boldsymbol{X}_{:,j} \in N\left(\boldsymbol{X}_{:,i}\right) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $i, j = 1, 2, \ldots, n$, and $\boldsymbol{X}_{:,i}$ is located in the ith column of the data matrix and represents the ith sample. $N(\boldsymbol{X}_{:,i})$ denotes the k-nearest neighbor set of the sample $\boldsymbol{X}_{:,i}$. $\boldsymbol{W}^{\boldsymbol{S}}$ is the similarity matrix of the data manifold. The graph Laplacian matrix of the data manifold is defined as $\boldsymbol{L}^{\boldsymbol{S}} = \boldsymbol{D}^{\boldsymbol{S}} - \boldsymbol{W}^{\boldsymbol{S}}$, where $\boldsymbol{D}^{\boldsymbol{S}}$ is a diagonal matrix and $[\boldsymbol{D}^{\boldsymbol{S}}]_{ii} = \Sigma_j[\boldsymbol{W}^{\boldsymbol{S}}]_{ij}$.

Using Eqs. (10) and (11), the similarity matrix as well as the Laplacian matrix of the feature manifold and the data manifold can be obtained, respectively. Denote $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m] \in \Re^{l\times m}$ as the low-dimensional embedding matrix of the feature manifold. To preserve the local structure information of high-dimensional features, the objective function to be solved is as follows:

$$\arg \min_{\boldsymbol{Y}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 \left[\boldsymbol{W}^{\boldsymbol{V}}\right]_{ij} = Tr\left(\boldsymbol{Y}\boldsymbol{L}^{\boldsymbol{V}}\boldsymbol{Y}^T\right) \tag{12}$$

It can be seen from Eq. (10) that if the similarity between the features $\boldsymbol{X}_{i,:}$ and $\boldsymbol{X}_{j,:}$ is high, $\boldsymbol{W}_{ij}^{\boldsymbol{V}}$ can take a large value. In order to minimize the Eq. (12), the vectors $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ should also have high similarity. So the local geometric structure information of the high-dimensional features can be preserved into the low-dimensional embedding matrix $\boldsymbol{Y}$.

In the data manifold, the locality preserving projections (LPP) method is used [38]. Specifically, a linear transformation matrix

$\boldsymbol{Z} \in \Re^{m\times l}$ is defined to map the high-dimensional data to its low-dimensional representation. To preserve the local structure information of the data manifold, the objective function is as follows:

$$\begin{aligned}&\arg\min_{\boldsymbol{Z}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|\boldsymbol{Z}^T\boldsymbol{X}_{:,i}-\boldsymbol{Z}^T\boldsymbol{X}_{:,j}\right\|_2^2\left[\boldsymbol{W}^{\boldsymbol{S}}\right]_{ij}=\\&Tr\left(\boldsymbol{Z}^T\boldsymbol{X}\boldsymbol{L}^{\boldsymbol{S}}\boldsymbol{X}^T\boldsymbol{Z}\right)\end{aligned} \tag{13}$$

Similar to the analysis of local structure preservation in feature manifold, the local structure information of the high-dimensional data can be preserved via the matrix $\boldsymbol{Z}$.

2.4. The framework of SLSDR

In order to make full use of the local geometric information of both the data manifold and the feature manifold to guide feature selection, the matrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are unified with the matrices $\boldsymbol{V}$ and $\boldsymbol{S}$, respectively. Then Eqs. (9), (12) and (13) are combined together and the obtained expression is as follows:

$$\begin{aligned}&\arg\min_{\boldsymbol{S},\boldsymbol{V}}\left\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_2^2+\alpha_1 Tr\left(\boldsymbol{V}\boldsymbol{L}^{\boldsymbol{V}}\boldsymbol{V}^T\right)+\alpha_2 Tr\left(\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{L}^{\boldsymbol{S}}\boldsymbol{X}^T\boldsymbol{S}\right)\\&\quad+\beta\Omega\left(\boldsymbol{S}\right)\\&s.t.\,\boldsymbol{S}\geq 0,\boldsymbol{V}\geq 0,\boldsymbol{S}^T\boldsymbol{S}=\boldsymbol{I}_l\end{aligned} \tag{14}$$

where $\Omega\left(\boldsymbol{S}\right)$ is the inner product regularization term. To ensure the robustness to outlier samples, the Frobenius norm imposed on the residual matrix of subspace learning is replaced by the $l_{2,1}$-norm. For ease to adjust parameters, we set $\alpha_1=\alpha_2=\alpha$, and the objective function of the proposed SLSDR is as follows:

$$\begin{aligned}&\arg\min_{\boldsymbol{S},\boldsymbol{V}}\left\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_{2,1}+\alpha\left(Tr\left(\boldsymbol{V}\boldsymbol{L}^{\boldsymbol{V}}\boldsymbol{V}^T\right)+Tr\left(\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{L}^{\boldsymbol{S}}\boldsymbol{X}^T\boldsymbol{S}\right)\right)\\&\quad+\beta\left(\left\|\boldsymbol{S}\boldsymbol{S}^T\right\|_1-\left\|\boldsymbol{S}\right\|_2^2\right)+\tfrac{\lambda}{2}\left\|\boldsymbol{S}^T\boldsymbol{S}-\boldsymbol{I}_l\right\|_2^2\\&s.t.\,\boldsymbol{S}\geq 0,\boldsymbol{V}\geq 0\end{aligned} \tag{15}$$

where $\alpha>0$, $\beta>0$, and $\lambda>0$ are the balance parameters.

2.5. Feature selection

After optimizing the objective function of the proposed SLSDR, the matrix $\boldsymbol{S}$ can be obtained, where $\boldsymbol{S}=[\boldsymbol{s}_1;\boldsymbol{s}_2;\ldots;\boldsymbol{s}_m]$, and $\boldsymbol{s}_i$ is the ith row of $\boldsymbol{S}$. Then $\|\boldsymbol{s}_i\|_2$ can be used as the evaluation value of the ith feature, and the larger the value of $\|\boldsymbol{s}_i\|_2$, the more important the ith feature. The evaluation values of all the features are sorted in the descending order, and the features corresponding to the first l evaluation values are selected. Finally, a new data matrix $\boldsymbol{X}_{new}\in\Re^{l\times n}$ is obtained and the feature selection is completed.

2.6. Connection with SGFS

It can be seen from Eq. (15) that when removing the graph regularization term of the data manifold, using the $l_{2,1}$-norm constrains the matrix $\boldsymbol{S}$ instead of the inner product regularization term, and replacing the $l_{2,1}$-norm imposed on the residual matrix of subspace learning by the Frobenius norm, SLSDR degenerates into SGFS. The objective function of SGFS is as follows:

$$\begin{aligned}&\arg\min_{\boldsymbol{S},\boldsymbol{V}}\left\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_2^2+\alpha Tr\left(\boldsymbol{V}\boldsymbol{L}\boldsymbol{V}^T\right)+\beta\left\|\boldsymbol{S}\right\|_{2,1}\\&\quad+\tfrac{\lambda}{2}\left\|\boldsymbol{S}^T\boldsymbol{S}-\boldsymbol{I}_l\right\|_2^2\\&s.t.\,\boldsymbol{S}\geq 0,\boldsymbol{V}\geq 0\end{aligned} \tag{16}$$

2.7. Update rules for SLSDR

Now the update rules are provided to optimize the objective function in Eq. (15). Since the function is non-convex for the matrices $\boldsymbol{S}$ and $\boldsymbol{V}$, it is difficult to find a globally optimal solution. To improve computational efficiency, an alternating iterative update method [54,55] is used to optimize this problem. Two Lagrange multipliers ψ_{ij} and ϕ_{ij} are introduced to constrain $\boldsymbol{S}_{ij}\geq 0$ and $\boldsymbol{V}_{ij}\geq 0$, respectively. The form of Lagrange function of the formula (15) is as follows:

$$\begin{aligned}\mathrm{L}\left(\boldsymbol{S},\boldsymbol{V}\right)=&\left\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_{2,1}+\alpha\left(Tr\left(\boldsymbol{V}\boldsymbol{L}^{\boldsymbol{V}}\boldsymbol{V}^T\right)\right.\\&\left.+\;Tr\left(\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{L}^{\boldsymbol{S}}\boldsymbol{X}^T\boldsymbol{S}\right)\right)\\&+\beta\left(\left\|\boldsymbol{S}\boldsymbol{S}^T\right\|_1-\left\|\boldsymbol{S}\right\|_2^2\right)+\tfrac{\lambda}{2}\left\|\boldsymbol{S}^T\boldsymbol{S}-\boldsymbol{I}_l\right\|_2^2\\&+Tr\left(\psi\boldsymbol{S}^T\right)+Tr\left(\phi\boldsymbol{V}^T\right)\end{aligned} \tag{17}$$

A diagonal matrix $\boldsymbol{U}\in\Re^{n\times n}$ is first introduced, and its ith element is defined as follows:

$$\boldsymbol{U}_{ii}=\frac{1}{\|\boldsymbol{e}_i\|_2} \tag{18}$$

where $\boldsymbol{E}=\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}$, and $\boldsymbol{e}_i$ is the ith row of the matrix $\boldsymbol{E}$. To avoid overflow, a small constant ε is introduced into Eq. (18), and the obtained formula is as follows:

$$\boldsymbol{U}_{ii}=\frac{1}{\max\left(\|\boldsymbol{e}_i\|_2,\varepsilon\right)} \tag{19}$$

According to the definition of $\boldsymbol{U}$, the term $\left\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right\|_{2,1}$can be rewritten as $Tr\left(\left(\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right)^T\boldsymbol{U}\left(\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right)\right)$, so the formula (17) can obtain the following form:

$$\begin{aligned}\mathrm{L}\left(\boldsymbol{S},\boldsymbol{V}\right)=&Tr\left(\left(\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right)^T\boldsymbol{U}\left(\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\right)\right)\\&+\alpha\left(Tr\left(\boldsymbol{V}\boldsymbol{L}^{\boldsymbol{V}}\boldsymbol{V}^T\right)\right.\\&\left.+\;Tr\left(\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{L}^{\boldsymbol{S}}\boldsymbol{X}^T\boldsymbol{S}\right)\right)+\beta\left(\left\|\boldsymbol{S}\boldsymbol{S}^T\right\|_1-\left\|\boldsymbol{S}\right\|_2^2\right)\\&+\frac{\lambda}{2}\left\|\boldsymbol{S}^T\boldsymbol{S}-\boldsymbol{I}_l\right\|_2^2+Tr\left(\psi\boldsymbol{S}^T\right)+Tr\left(\phi\boldsymbol{V}^T\right)\end{aligned} \tag{20}$$

First, in order to update $\boldsymbol{S}$, $\boldsymbol{V}$ and $\boldsymbol{U}$ are fixed. By taking the partial derivative of formula (20) with respect to $\boldsymbol{S}$, the obtained formula is as follows:

$$\begin{aligned}\frac{\partial\mathrm{L}}{\partial\boldsymbol{S}}=&2\left(\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\boldsymbol{V}^T-\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{V}^T\right)+2\alpha\boldsymbol{X}\left(\boldsymbol{D}^{\boldsymbol{S}}-\boldsymbol{W}^{\boldsymbol{S}}\right)\boldsymbol{X}^T\boldsymbol{S}\\&+2\beta\left(\boldsymbol{1}_{m\times m}\boldsymbol{S}-\boldsymbol{S}\right)+2\lambda\left(\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{S}-\boldsymbol{S}\right)+\psi\end{aligned} \tag{21}$$

where $\boldsymbol{1}_{m\times m}$ is an $m\times m$ matrix with all the elements being 1. By using the Karush–Kuhn–Tucker (KKT) conditions [56,57]

$\psi_{ij}\boldsymbol{S}_{ij}=0$, the following formula can be obtained:

$$\begin{aligned}&\left[\left(\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\boldsymbol{V}^T-\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{V}^T\right)+\alpha\boldsymbol{X}\left(\boldsymbol{D}^{\boldsymbol{S}}-\boldsymbol{W}^{\boldsymbol{S}}\right)\boldsymbol{X}^T\boldsymbol{S}\right.\\&\left.+\beta\left(\boldsymbol{1}_{m\times m}\boldsymbol{S}-\boldsymbol{S}\right)+\lambda\left(\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{S}-\boldsymbol{S}\right)\right]_{ij}\boldsymbol{S}_{ij}=0\end{aligned} \tag{22}$$

Therefore, the iterative update rule for $\boldsymbol{S}$ is as follows:

$$\boldsymbol{S}_{ij}\leftarrow\boldsymbol{S}_{ij}\frac{\left[\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{V}^T+\left(\alpha\boldsymbol{X}\boldsymbol{W}^{\boldsymbol{S}}\boldsymbol{X}^T+(\beta+\lambda)\boldsymbol{I}_m\right)\boldsymbol{S}\right]_{ij}}{\left[\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}\boldsymbol{V}^T+\left(\alpha\boldsymbol{X}\boldsymbol{D}^{\boldsymbol{S}}\boldsymbol{X}^T+\beta\boldsymbol{1}_{m\times m}+\lambda\boldsymbol{S}\boldsymbol{S}^T\right)\boldsymbol{S}\right]_{ij}} \tag{23}$$

Then, to update $\boldsymbol{V}$, $\boldsymbol{S}$ and $\boldsymbol{U}$ are fixed. By taking the partial derivative of the formula (20) with respect to $\boldsymbol{V}$, the obtained formula is as follows:

$$\frac{\partial\mathrm{L}}{\partial\boldsymbol{V}}=2\left(\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}-\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\right)+2\alpha\boldsymbol{V}\left(\boldsymbol{D}^{\boldsymbol{V}}-\boldsymbol{W}^{\boldsymbol{V}}\right)+\phi \tag{24}$$

By using the Karush–Kuhn–Tucker (KKT) conditions $\phi_{ij}\boldsymbol{V}_{ij}=0$, the following formula can be obtained:

$$\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}-\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T+\alpha\boldsymbol{V}\left(\boldsymbol{D}^{\boldsymbol{V}}-\boldsymbol{W}^{\boldsymbol{V}}\right)\right]_{ij}\boldsymbol{V}_{ij}=0 \tag{25}$$

Table 1
The procedure of SLSDR.

Input: Data matrix $\boldsymbol{X} \in \Re^{m\times n}$; neighbor size k; balance parameters α, β, λ; maximum number of iterations *NIter*; Gaussian scale parameter σ; number of selected features l.
Output: Index set of the selected features *Index*; new data matrix $\boldsymbol{X}_{new} \in \Re^{l\times n}$.

1. Construct the k-nearest neighbor graphs $\boldsymbol{G_0}$=(V_0, E_0) and $\boldsymbol{G_1}$=(V_1, E_1) for the feature manifold and the data manifold, respectively.
2. Compute the similarity matrices $\boldsymbol{W^V}$ and $\boldsymbol{W^S}$, the graph Laplacian matrix $\boldsymbol{L^V}$ and $\boldsymbol{L^S}$.
3. Initialize $\boldsymbol{U}$, $\boldsymbol{S}$, $\boldsymbol{V}$.
4. Update $\boldsymbol{U}$, $\boldsymbol{S}$, $\boldsymbol{V}$ according to the iteration update rules in Eqs. (19), (23) and (26) until the maximum number of iterations *NIter* is reached.
5. Calculate the evaluation value of the ith feature based on $||\boldsymbol{s}_i||_2$, sort the evaluation values of all features in descending order and select the features corresponding to the first l evaluation values. Then obtain the index set of the selected features *Index* and a new data matrix $\boldsymbol{X}_{new} \in \Re^{l\times n}$.

Then, the obtained iterative update rule for $\boldsymbol{V}$ is as follows:

$$\boldsymbol{V}_{ij} \leftarrow \boldsymbol{V}_{ij}\frac{\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T + \alpha\boldsymbol{V}\boldsymbol{W}^{\boldsymbol{V}}\right]_{ij}}{\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V} + \alpha\boldsymbol{V}\boldsymbol{D}^{\boldsymbol{V}}\right]_{ij}} \tag{26}$$

Based on the above analyses, the optimization process of the proposed SLSDR algorithm is shown in Table 1.

2.8. Computational complexity analysis

The computational complexity of SLSDR in Table 1 is analyzed. Where n represents the total number of data samples, m represents the number of features included in each sample, l is the number of selected features, and t is the number of iterations. To construct the Laplacian matrices $\boldsymbol{L}^{\boldsymbol{V}}$ and $\boldsymbol{L}^{\boldsymbol{S}}$ in the data manifold and the feature manifold, the computational complexity is $O\left(mn^2 + nm^2\right)$. Additionally, in each iteration, to update the matrices $\boldsymbol{U}$, $\boldsymbol{S}$ and $\boldsymbol{V}$, the computational complexity is $O\left((m+l)\left(n^2+mn+ml\right)\right)$. So the total computational complexity is $O\left(t\,(m+l)\left(n^2+mn+ml\right)\right)$. In practical applications, since $l<<m$ and $l<<n$, and there is $m>n$ or $m<n$, thus the obtained overall complexity of SLSDR is $O\,(tnm\,(n+m))$.

2.9. Convergence analysis

Next, the convergence analysis of the proposed SLSDR is presented. Similar to the methods in [4,33], we will prove that the objective function in Eq. (15) is non-increasing under the update rules (23) and (26) of the variables $\boldsymbol{S}$ and $\boldsymbol{V}$.

Definition 1. If there is a function $J(h, h')$ that makes $C(h)$ satisfy the following conditions:

$$J(h,h') \geq C(h), J(h,h) = C(h) \tag{27}$$

Then C is non-increasing under the following update formula:

$$h^{(t+1)} = \arg\min_{h} J(h, h^{(t)}) \tag{28}$$

where $J(h, h')$ is an auxiliary function of $C(h)$.

Proof. $C(h^{(t+1)}) \leq J(h^{(t+1)}, h^{(t)}) \leq J(h^{(t)}, h^{(t)}) = C(h^{(t)})$.

Since the monotonicity of the objective function (15) under the update rule of the variable $\boldsymbol{V}$ needs to be proved, the terms relating to the variable $\boldsymbol{V}$ in Eq. (15) are retained, and the following function is obtained:

$$\begin{aligned} C\left(\boldsymbol{V}\right) = {} & Tr\left(\boldsymbol{V}^T\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V} - \boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V} - \boldsymbol{V}^T\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\right) \\ & + \alpha Tr\left(\boldsymbol{V}\boldsymbol{L}^{\boldsymbol{V}}\boldsymbol{V}^T\right) \end{aligned} \tag{29}$$

By taking the first-order and the second-order partial derivatives of $C(\boldsymbol{V}\ )$ with respect to $\boldsymbol{V}$, the following formulas can be obtained:

$$C'_{ij} = \left[\frac{\partial C}{\partial \boldsymbol{V}}\right]_{ij} = \left[2\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V} - 2\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T + 2\alpha\boldsymbol{V}\boldsymbol{L}^{\boldsymbol{V}}\right]_{ij} \tag{30}$$

$$C''_{ij} = 2\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\right]_{ii} + 2\alpha\left[\boldsymbol{L}^{\boldsymbol{V}}\right]_{jj} \tag{31}$$

Lemma 1. *Giving the auxiliary functions of C_{ij}, and the form is as follows:*

$$\begin{aligned} J\left(\boldsymbol{V}_{ij}, \boldsymbol{V}_{ij}^{(t)}\right) = {} & C_{ij}\left(\boldsymbol{V}_{ij}^{(t)}\right) + C'_{ij}\left(\boldsymbol{V}_{ij}^{(t)}\right)\left(\boldsymbol{V}_{ij} - \boldsymbol{V}_{ij}^{(t)}\right) \\ & + \frac{\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}^{(t)} + \alpha\boldsymbol{V}^{(t)}\boldsymbol{D}^{\boldsymbol{V}}\right]_{ij}}{\boldsymbol{V}_{ij}^{(t)}}\left(\boldsymbol{V}_{ij} - \boldsymbol{V}_{ij}^{(t)}\right)^2 \end{aligned} \tag{32}$$

Denoting the Taylor expansion of $C_{ij}(\boldsymbol{V}_{ij})$ as follows:

$$\begin{aligned} C_{ij}\left(\boldsymbol{V}_{ij}\right) = {} & C_{ij}\left(\boldsymbol{V}_{ij}^{(t)}\right) + C'_{ij}\left(\boldsymbol{V}_{ij}^{(t)}\right)\left(\boldsymbol{V}_{ij} - \boldsymbol{V}_{ij}^{(t)}\right) \\ & + \left\{\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\right]_{ii} + \alpha\left[\boldsymbol{L}^{\boldsymbol{V}}\right]_{jj}\right\}\left(\boldsymbol{V}_{ij} - \boldsymbol{V}_{ij}^{(t)}\right)^2 \end{aligned} \tag{33}$$

It can be seen from formulas (32) and (33) that $J\left(\boldsymbol{V}_{ij}, \boldsymbol{V}_{ij}^{(t)}\right) \geq C_{ij}\left(\boldsymbol{V}_{ij}\right)$ is equivalent to:

$$\frac{\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}^{(t)} + \alpha\boldsymbol{V}^{(t)}\boldsymbol{D}^{\boldsymbol{V}}\right]_{ij}}{\boldsymbol{V}_{ij}^{(t)}} \geq \left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\right]_{ii} + \alpha\left[\boldsymbol{L}^{\boldsymbol{V}}\right]_{jj} \tag{34}$$

It is obvious that the following formula holds:

$$\begin{aligned} \left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}^{(t)}\right]_{ij} = {} & \sum_{b=1}^{l}\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\right]_{ib}\boldsymbol{V}_{bj}^{(t)} \\ \geq {} & \left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\right]_{ii}\boldsymbol{V}_{ij}^{(t)} \end{aligned} \tag{35}$$

And there is:

$$\begin{aligned} \alpha\left[\boldsymbol{V}^{(t)}\boldsymbol{D}^{\boldsymbol{V}}\right]_{ij} = {} & \alpha\sum_{b=1}^{m}\boldsymbol{V}_{ib}^{(t)}\boldsymbol{D}_{bj}^{\boldsymbol{V}} \geq \alpha\boldsymbol{V}_{ij}^{(t)}\boldsymbol{D}_{jj}^{\boldsymbol{V}} \\ \geq {} & \alpha\boldsymbol{V}_{ij}^{(t)}\left[\boldsymbol{D}^{\boldsymbol{V}} - \boldsymbol{W}^{\boldsymbol{V}}\right]_{jj} = \alpha\boldsymbol{V}_{ij}^{(t)}\left[\boldsymbol{L}^{\boldsymbol{V}}\right]_{jj} \end{aligned} \tag{36}$$

So the inequality (34) holds, i.e. $J\left(\boldsymbol{V}_{ij}, \boldsymbol{V}_{ij}^{(t)}\right) \geq C_{ij}\left(\boldsymbol{V}_{ij}\right)$ holds. Obviously, the equation $J\left(\boldsymbol{V}_{ij}, \boldsymbol{V}_{ij}\right) = C_{ij}\left(\boldsymbol{V}_{ij}\right)$ also holds.

Then, we will prove that the update rule of variable $\boldsymbol{V}$ satisfies the update formula (28) that makes C_{ij} non-increasing.

By substituting$J\left(\boldsymbol{V}_{ij}, \boldsymbol{V}_{ij}^{(t)}\right)$ in Eq. (32) into Eq. (28), the following formula can be obtained:

$$\boldsymbol{V}_{ij}^{(t+1)} = \boldsymbol{V}_{ij}^{(t)} - \boldsymbol{V}_{ij}^{(t)}\frac{C'_{ij}\left(\boldsymbol{V}_{ij}^{(t)}\right)}{2\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}^{(t)} + \alpha\boldsymbol{V}^{(t)}\boldsymbol{D}^{\boldsymbol{V}}\right]_{ij}} \tag{37}$$

Substituting Eq. (30) into Eq. (37) gives the following expression:

$$\boldsymbol{V}_{ij}^{(t+1)} = \boldsymbol{V}_{ij}^{(t)}\frac{\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T + \alpha\boldsymbol{V}^{(t)}\boldsymbol{W}^{\boldsymbol{V}}\right]_{ij}}{\left[\boldsymbol{S}^T\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{V}^{(t)} + \alpha\boldsymbol{V}^{(t)}\boldsymbol{D}^{\boldsymbol{V}}\right]_{ij}} \tag{38}$$

It can be seen that Eq. (38) is the update rule of variable $\boldsymbol{V}$, so C_{ij} is non-increasing under the update rule (26). The convergence proof under the update rule of variable $\boldsymbol{S}$ is similar to that under the update rule of variable $\boldsymbol{V}$. And it is found that the objective function is also non-increasing under the update rule (23). Therefore, it can be concluded that the objective function in Eq. (15) is non-increasing under the update rules (23) and (26).

Table 2
The information of thirteen datasets.

Dataset	Size	Dim	Classes	Type
Ionosphere	351	34	2	Text image
JAFFE	213	676	10	Face image
YaleB	2414	1024	38	Face image
COIL20	1440	1024	20	Digital image
AR10P	130	2400	10	Face image
Umist	575	644	20	Face image
Isolet	1560	617	26	Letter image
Orl64	400	4096	40	Face image
ORL	400	1024	40	Face image
Lung_dis	73	325	7	Biological
PIE10P	210	2420	10	Face image
Yale64	165	4096	15	Face image
TOX_171	171	5748	4	Biological

3. Experiments

In this section, the experimental results of the proposed SLSDR and six other comparison algorithms on twelve benchmark datasets are presented. The *k-means* clustering algorithm [58,59] is used to evaluate the performance of different feature selection algorithms. The settings of the maximum number of iterations and the k value are provided. Meanwhile, the experimental results are analyzed and the robustness test of SLSDR to outlier samples is provided. Then, the effectiveness of the inner product regularization term is verified and the parameter sensitivity of SLSDR is analyzed.

3.1. Datasets

Thirteen datasets are used in the experiment, and they can be divided into text image, biological data, face image and digital image [33,60,61]. The detailed information of these datasets is described in Table 2.

3.2. Comparison algorithm

To verify the effectiveness of the proposed SLSDR, SLSDR is compared with the following six unsupervised feature selection algorithms:

(1) Baseline: clustering all features directly without feature selection.

(2) LapScor: Laplacian Score [28] uses local geometric information of data to select features and calculates the score of each feature separately.

(3) UDFS: unsupervised discriminant feature selection [13], which combines local discriminant analysis with $l_{2,1}$-norm regularization term to guide the process of feature selection.

(4) MFFS: matrix factorization feature selection [31], which introduces matrix decomposition strategy into the framework of subspace learning to complete feature selection.

(5) MCFS: multi-cluster data feature selection [18] uses l_1-norm and spectral analysis to select features.

(6) SGFS: subspace learning-based graph regularized feature selection [33] introduces the feature graph and $l_{2,1}$-norm into the framework of subspace learning of matrix factorization to select the representative feature subsets.

3.3. Evaluation metrics

Clustering Accuracy (ACC) [4,33] and Normalized Mutual Information (NMI) [4,6] are two widely used metrics for evaluating the performance of unsupervised feature selection algorithms. And the greater the values of the two metrics, the better the performance of the corresponding algorithm. Denoting e_j and d_j as the clustering label and the ground truth label of the sample $\boldsymbol{x}_i$, respectively. n is the total number of samples, then the formula of Clustering Accuracy (ACC) is as follows:

$$\mathrm{ACC} = \frac{1}{n}\sum_{j=1}^{n}\delta(d_j, \mathrm{map}(e_j)) \tag{39}$$

where map(.) is an optimal mapping function, in which Hungarian [62] is used to match the obtained cluster labels with the real labels of datasets. $\delta(a, b)$ is an indicator function that takes only 0–1 values, and $\delta(a, b) = 1$, if $a = b$, otherwise $\delta(a, b) = 0$.

Given two random variables P and R, the formula of Normalized Mutual Information (NMI) is as follows:

$$\mathrm{NMI(P, R)} = \frac{\mathrm{I(P, R)}}{\sqrt{\mathrm{H(P)H(R)}}} \tag{40}$$

where I(P, R) is the mutual information between P and R, and H(P) and H(R) are the entropy values of P and R, respectively. To apply this criterion to clustering tasks, P and R are considered as the clustering labels and the real labels of samples in this paper, respectively.

3.4. Experimental results and analysis

3.4.1. Experimental settings

First of all, the parameter settings of different algorithms are given in this paper. For LapScor, MCFS, SGFS and SLSDR, the parameter of nearest neighbors k is set to 5. The Gaussian scale parameter σ is searched in the range of $\{10^{+1}, 10^{+2}, 10^{+3}, 10^{+4}, 10^{+5}\}$. For SGFS and SLSDR, the balance parameters α and β are adjusted in the range of $\{10^{-8}, 10^{-7},\ldots, 10^{+7}, 10^{+8}\}$ and the range of the parameter λ is set to $\{10^{+0}, 10^{+1},\ldots,10^{+7}, 10^{+8}\}$. For MFFS, UDFS, MCFS, SGFS and SLSDR, the maximum number of iterations is set to 30. For all the datasets, the number of selected features l is set to {20, 30, 40, 50, 60, 70, 80, 90, 100}. In clustering tasks, *k-means* is sensitive to initial values, so this paper repeats *k-means* clustering 20 times to get the average values for all algorithms. Then, the balance parameters are adjusted so as to obtain the best average values of ACC and NMI.

Since the maximum number of iterations directly affects the convergence of SLSDR, and the k-nearest neighbor parameter has an important influence on the performance of SLSDR, the settings of the maximum number of iterations and the k-nearest neighbor parameter are analyzed in this paper. And the corresponding experiments are shown as follows.

(1) The setting of the maximum number of iterations

In this paper, the maximum number of iterations is set based on the results of the convergence test, and the convergence curves of SLSDR on twelve datasets are shown in Fig. 1. The maximum number of iterations is set to 30 and the reasons are analyzed as follows.

In Fig. 1, the horizontal axis and the vertical axis represent the number of iterations and the value of the objective function in each iteration, respectively. It can be seen from Fig. 1 that the value of the objective function decreases as the number of iterations increases. And on most datasets, the objective function becomes relatively stable within 20 iterations. Meanwhile, the objective function can converge within 30 iterations for all the datasets. So conclusions can be drawn that within 30 iterations, SLSDR can achieve good convergence performance. Therefore, the maximum number of iterations is set to 30 for the proposed SLSDR.

(2) The setting of *k* value

Since the k value has an important impact on experimental results, the setting of k is discussed here. In this experiment, six datasets of Lung_dis, Isolet, ORL, AR10P, ORL64 and COIL20 are

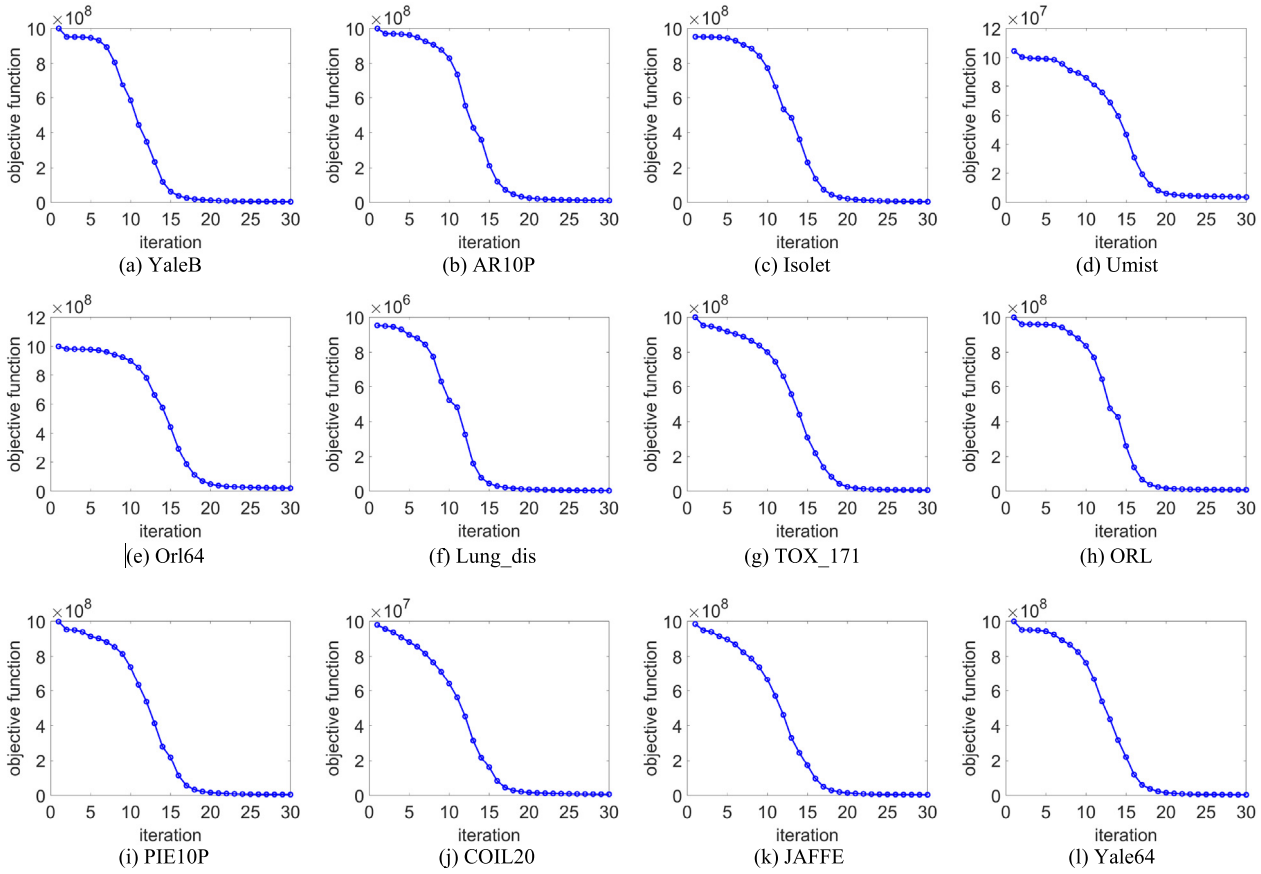


Fig. 1. The convergence curves of the objective function on twelve datasets.

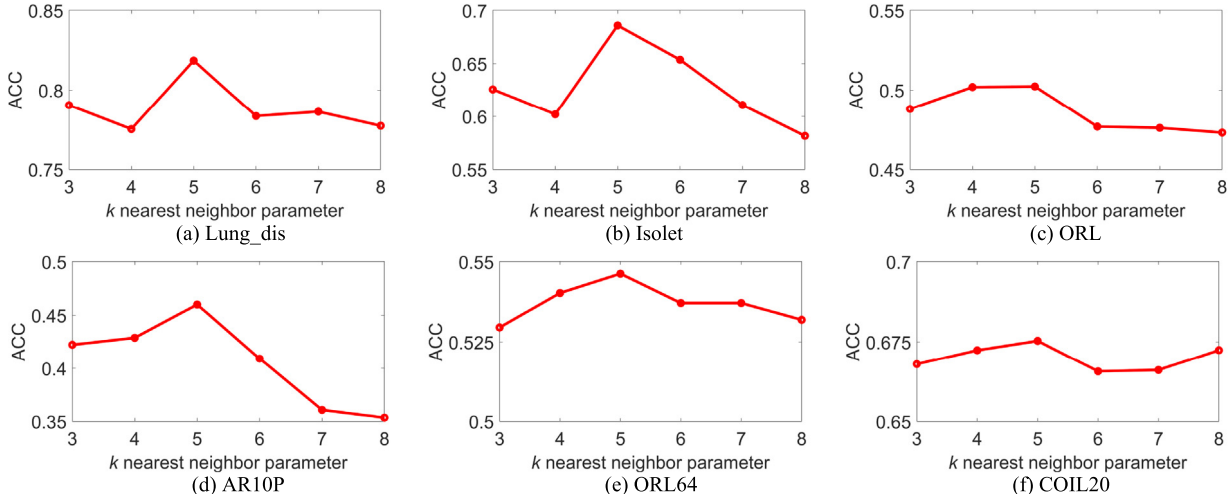


Fig. 2. Clustering accuracy of SLSDR on six datasets with different k values.

selected as the test datasets. Then other parameters are fixed and the value of k is adjusted in the range {3, 4, 5, 6, 7, 8}. Figs. 2 and 3 show the curves of the clustering results of SLSDR with different k values. The horizontal axis indicates the nearest neighbor parameter k, and the vertical axis indicates the clustering accuracy (ACC) in Fig. 2 and the normalized mutual information (NMI) in Fig. 3. The analyses are as follows.

As can be seen from the above experimental results, for the dataset ORL64, the best ACC can be obtained when k takes 5 and the best NMI can be obtained when k takes 5 and 6. For the datasets Lung_dis, Isolet, ORL, AR10P and COIL20, the best ACC and NMI values can be obtained when k takes 5, and the accuracy is reduced when k takes other values. Generally, in order to preserve the local structure information, the k value should be set relatively small. Meanwhile, it can be seen from the above figures that when k is set to 5, not only can the best ACC and NMI values be obtained for all the test datasets, but also the local structure information is maintained. Therefore, the value of k is set to 5 in this paper.

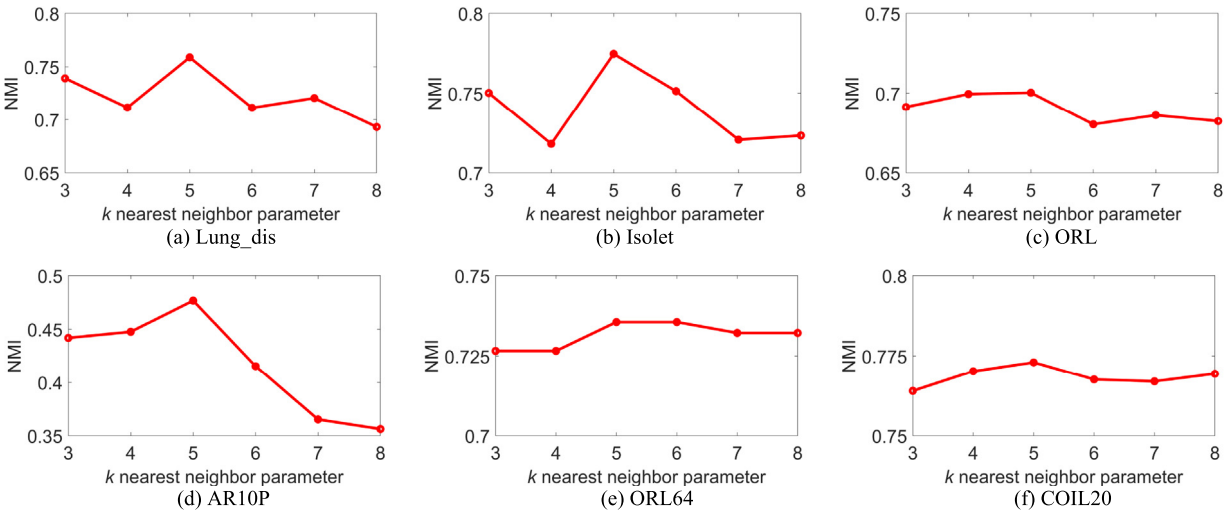


Fig. 3. Normalized Mutual Information of SLSDR on six datasets with different k values.

3.4.2. The effectiveness evaluation of SLSDR

To verify the effectiveness of the proposed SLSDR, the Ionosphere dataset is utilized to test whether SLSDR can select the most representative features or not. This dataset consists of 351 samples, and each sample contains 34 features. In this experiment, 66 new features need to be generated artificially, and each new feature is the linear combination of the 34 original features. In this process, the linear combination coefficients are randomly generated and they are normalized. All features are then put together and a total of 100 features are obtained. Among them, the first 34 features are the original features, and the rest are the generated features, so each sample possesses 100 features now. After solving the objective function in Eq. (15), the matrix $\boldsymbol{S}$ can be obtained, and then $||\boldsymbol{s}_i||_2$ is used as the evaluation value of the ith feature. Then the evaluation values of all features can be obtained. These evaluation values are used to generate a diagonal matrix shown in Fig. 4.

It can be seen from Fig. 4 that compared with the 66 synthetic features, the first 34 features can obtain larger evaluation values. So the proposed SLSDR has great effectiveness and the most representative features can be selected.

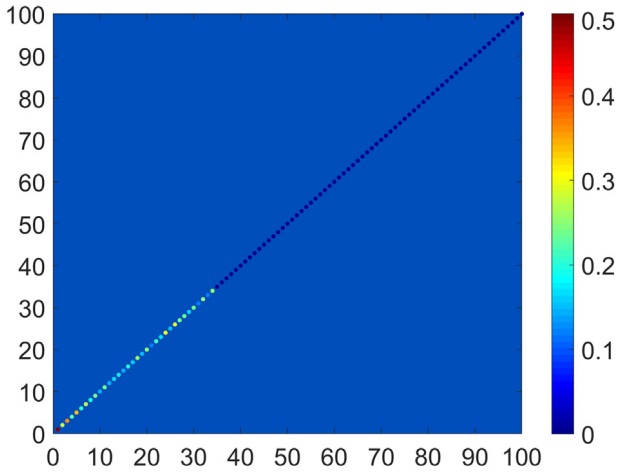


Fig. 4. The diagonal matrix of evaluation values of 100 features.

3.4.3. Experimental results and analysis

To compare the performance of seven different algorithms, the ACC and NMI values for these algorithms on twelve datasets are shown in Tables 3 and 4, respectively. The bold marked values are the best results, and the underline marked values are the second best results. And the analyses of these results are as follows.

It can be seen from Tables 3 and 4 that the proposed SLSDR can obtain the best ACC and NMI values in most cases compared with the other six comparison algorithms. Except the ORL dataset, the proposed SLSDR is superior to the Baseline method, which fully demonstrates that SLSDR has great advantages. On all twelve datasets, SLSDR has better results than SGFS algorithm. The reason is that SLSDR uses the local geometric structure information of both the data manifold and the feature manifold to guide the process of feature selection. Additionally, by using the inner product regularization term, SLSDR ensures the sparsity of rows of $\boldsymbol{S}$ and considers the correlations between features, so the representative and low-redundant features can be selected, which further improves the performance of feature selection.

Figs. 5 and 6 show the curves of the clustering results of seven different algorithms on twelve datasets. The horizontal axis indicates the number of selected features l, and the vertical axis indicates the clustering accuracy (ACC) and standard deviation (STD) in Fig. 5 and the normalized mutual information (NMI) and standard deviation (STD) in Fig. 6.

As can be seen from Fig. 5, the ACC values of SLSDR are higher than those of the other five comparison algorithms on all the datasets. And on the datasets YaleB, AR10P, Umist, Lung_dis, TOX_171 and PIE10P, SLSDR can obtain better performance than Baseline method, which fully demonstrates the advantages of SLSDR. It can be seen from Fig. 6 that the NMI values of SLSDR are higher than those of the other five comparison algorithms on all datasets. Meanwhile, on the datasets YaleB, AR10P, Umist, Lung_dis and PIE10P, SLSDR is superior to the Baseline method, which proves the effectiveness of the proposed SLSDR.

3.4.4. Robustness test to outlier samples

To test the robustness of SLSDR to outlier samples, three test datasets are utilized in this paper. Then, two kinds of outlier samples are added into these datasets, so the corrupted datasets that contain outlier samples are obtained. In this experiment, YaleB, COIL20 and Umist are used as the test datasets, and the "umbrella" and "watch" outliers which are selected from Caltech101 database [63] are used as two kinds of outlier samples. Each corrupted dataset consists of a test dataset and a kind of outlier samples, so a total of six corrupted datasets can be obtained. The number of outlier samples is set to 10, 20 and 30

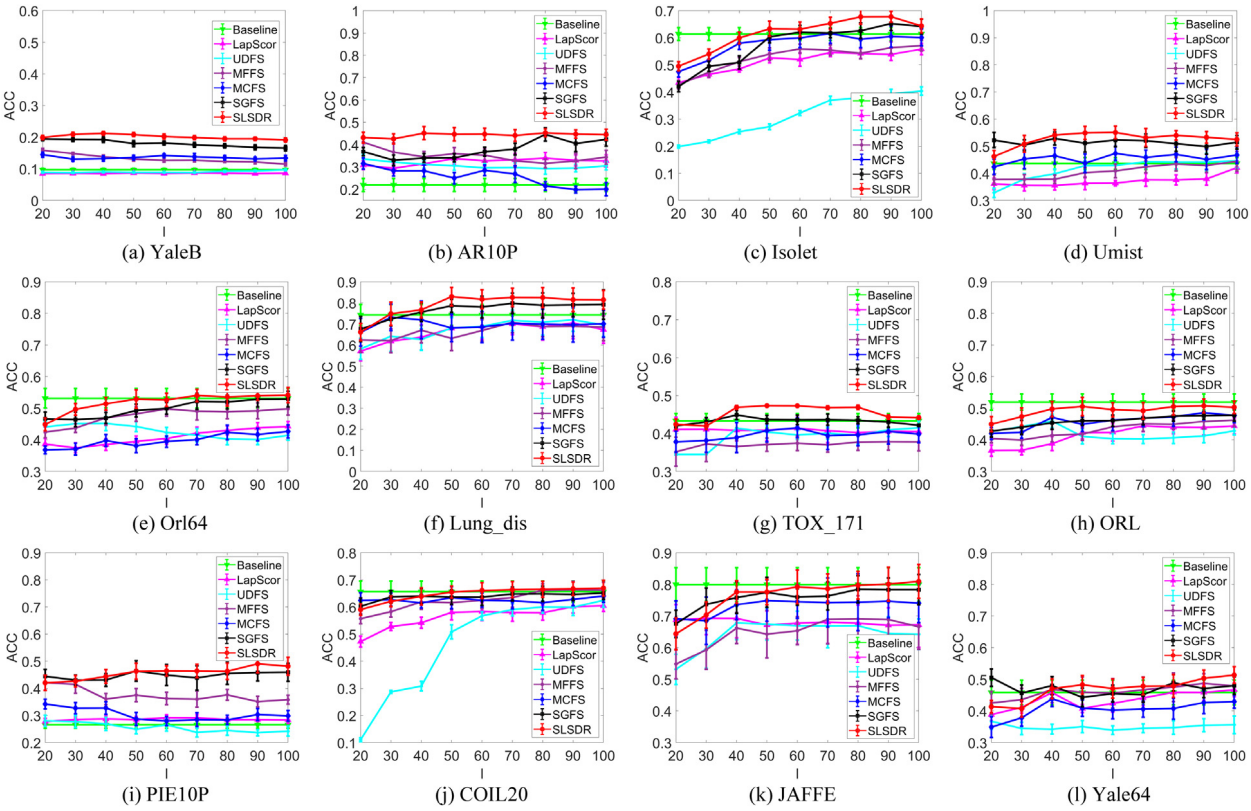


Fig. 5. Clustering accuracy of seven algorithms on twelve datasets with different number of selected features.

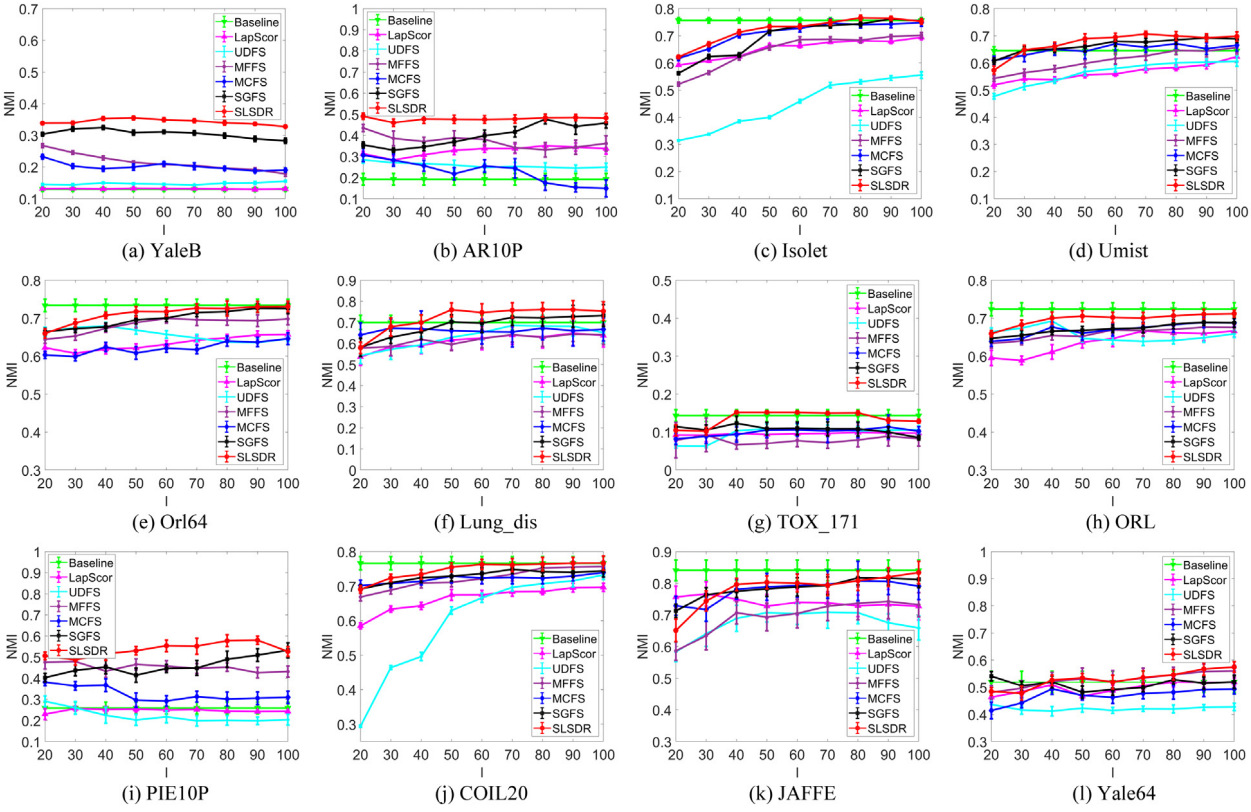


Fig. 6. Normalized Mutual Information of seven algorithms on twelve datasets with different number of selected features.

Table 3
Clustering accuracy of seven algorithms on twelve datasets (ACC±STD%).

Dataset	Baseline	LapScor	UDFS	MFFS	MCFS	SGFS	SLSDR
YaleB	9.69 ± 0.49	8.69 ± 0.25	9.66 ± 0.25	15.88 ± 0.65	14.47 ± 0.79	16.75 ± 0.88	**19.18 ± 0.75**
AR10P	21.96 ± 2.92	33.92 ± 2.57	33.54 ± 2.05	41.08 ± 2.41	31.54 ± 2.51	44.62 ± 2.95	**46.00 ± 4.29**
Isolet	61.41 ± 2.38	55.83 ± 2.14	40.36 ± 1.56	57.08 ± 2.00	61.60 ± 3.60	65.22 ± 2.53	**68.58 ± 2.39**
Umist	43.61 ± 2.16	42.09 ± 1.94	44.64 ± 2.67	44.41 ± 3.67	47.35 ± 2.62	52.28 ± 2.76	**54.07 ± 1.78**
Orl64	53.17 ± 3.07	44.21 ± 1.59	45.18 ± 2.24	49.84 ± 1.97	42.66 ± 2.06	52.94 ± 2.40	**54.62 ± 2.02**
Lung_dis	74.25 ± 4.94	70.41 ± 7.34	71.99 ± 5.51	70.75 ± 4.87	73.15 ± 6.26	79.79 ± 4.85	**81.85 ± 5.31**
TOX_171	43.36 ± 1.90	41.35 ± 2.96	41.52 ± 1.80	37.84 ± 2.20	41.52 ± 2.60	44.09 ± 2.32	**46.81 ± 0.55**
ORL	**51.96 ± 2.57**	44.44 ± 1.88	45.95 ± 2.17	46.24 ± 2.11	48.58 ± 2.34	47.51 ± 1.63	50.8 ± 1.88
PIE10P	26.52 ± 1.28	29.19 ± 1.03	28.00 ± 2.08	42.10 ± 2.74	34.19 ± 1.79	44.64 ± 3.91	**46.83 ± 2.84**
COIL20	65.64 ± 3.94	60.41 ± 2.11	62.44 ± 2.57	66.35 ± 3.13	64.06 ± 2.36	65.18 ± 2.63	**67.53 ± 3.89**
JAFFE	79.98 ± 5.32	69.27 ± 5.19	67.82 ± 5.29	69.06 ± 7.37	74.86 ± 6.56	79.44 ± 4.46	**81.85 ± 5.07**
Yale64	45.82 ± 3.90	46.67 ± 2.15	36.73 ± 2.44	48.73 ± 3.01	43.85 ± 2.48	50.39 ± 3.31	**52.33 ± 3.12**

Table 4
Normalized mutual information of seven algorithms on twelve datasets (NMI±STD%).

Dataset	Baseline	LapScor	UDFS	MFFS	MCFS	SGFS	SLSDR
YaleB	12.97 ± 0.58	13.24 ± 0.30	15.49 ± 0.29	26.77 ± 0.61	23.25 ± 0.68	28.57 ± 0.77	**32.83 ± 0.59**
AR10P	19.17 ± 2.77	35.08 ± 1.40	28.37 ± 1.57	43.46 ± 1.71	30.67 ± 3.07	47.75 ± 2.36	**48.48 ± 1.47**
Isolet	75.66 ± 1.00	69.45 ± 0.91	55.57 ± 1.18	70.19 ± 1.00	74.81 ± 1.28	76.18 ± 1.00	**77.47 ± 0.59**
Umist	64.47 ± 1.46	62.33 ± 1.91	60.63 ± 1.86	65.65 ± 2.04	67.07 ± 1.72	69.23 ± 1.29	**70.03 ± 1.65**
Orl64	73.36 ± 1.65	65.69 ± 1.00	67.93 ± 0.97	69.95 ± 1.27	64.58 ± 1.55	72.79 ± 1.38	**73.56 ± 1.41**
Lung_dis	69.97 ± 3.38	64.86 ± 5.71	68.52 ± 3.52	64.42 ± 3.42	67.22 ± 4.34	72.44 ± 3.94	**75.87 ± 4.42**
TOX_171	14.32 ± 1.55	9.90 ± 1.80	10.98 ± 1.03	9.26 ± 4.42	11.36 ± 3.13	12.47 ± 1.92	**16.21 ± 0.57**
ORL	**72.36 ± 1.71**	66.62 ± 1.33	69.25 ± 1.10	67.61 ± 1.63	68.86 ± 1.39	68.38 ± 1.22	71.08 ± 1.27
PIE10P	25.86 ± 2.80	25.58 ± 1.20	29.05 ± 2.47	47.87 ± 3.62	38.18 ± 1.78	52.45 ± 3.51	**57.06 ± 3.19**
COIL20	76.62 ± 1.92	69.67 ± 1.18	73.18 ± 1.27	75.64 ± 1.67	73.94 ± 1.29	74.79 ± 1.53	**77.30 ± 2.15**
JAFFE	84.07 ± 3.25	76.64 ± 3.83	70.82 ± 5.05	74.23 ± 5.44	80.76 ± 6.21	81.96 ± 2.37	**84.17 ± 3.17**
Yale64	52.01 ± 3.93	51.97 ± 1.90	43.61 ± 2.00	56.03 ± 2.12	49.38 ± 2.06	54.29 ± 2.11	**59.00 ± 2.09**

Table 5
Clustering accuracy of SLSDR and SGFS on six corrupted datasets with different numbers of outlier samples (ACC±STD%).

Number of outlier samples		10		20		30	
Dataset	Outlier samples	SGFS	SLSDR	SGFS	SLSDR	SGFS	SLSDR
YaleB	umbrella	16.23 ± 0.72	19.61 ± 0.65	16.83 ± 0.82	19.41 ± 0.85	16.74 ± 0.67	17.45 ± 0.61
	watch	17.73 ± 0.75	19.47 ± 0.53	16.68 ± 0.72	19.45 ± 0.79	16.02 ± 0.75	17.42 ± 0.80
COIL20	umbrella	63.88 ± 2.25	67.65 ± 3.34	64.68 ± 2.17	67.68 ± 3.04	66.37 ± 3.43	67.27 ± 2.57
	watch	65.85 ± 2.21	68.79 ± 2.69	63.33 ± 2.93	67.36 ± 2.82	62.19 ± 2.82	67.68 ± 2.10
Umist	umbrella	48.58 ± 1.89	53.05 ± 2.60	47.19 ± 2.08	51.84 ± 1.87	46.49 ± 2.82	52.87 ± 2.66
	watch	47.63 ± 2.28	51.84 ± 3.64	46.37 ± 2.62	49.83 ± 2.29	48.56 ± 4.15	49.31 ± 2.49

Table 6
Normalized mutual information of SLSDR and SGFS on six corrupted datasets with different numbers of outlier samples (NMI±STD%).

Number of outlier samples		10		20		30	
Dataset	Outlier samples	SGFS	SLSDR	SGFS	SLSDR	SGFS	SLSDR
YaleB	umbrella	28.06 ± 0.63	33.38 ± 0.47	28.15 ± 0.72	32.91 ± 0.66	27.64 ± 0.63	29.76 ± 0.73
	watch	29.87 ± 0.83	33.17 ± 0.51	27.57 ± 0.75	33.37 ± 0.73	26.39 ± 0.57	29.90 ± 0.69
COIL20	umbrella	74.11 ± 1.55	77.70 ± 1.94	74.49 ± 1.20	77.73 ± 1.71	75.91 ± 2.15	77.47 ± 2.04
	watch	75.58 ± 1.32	77.82 ± 1.47	73.45 ± 1.43	77.12 ± 1.55	72.28 ± 1.91	77.40 ± 1.29
Umist	umbrella	67.34 ± 1.17	68.99 ± 1.34	66.25 ± 1.68	68.87 ± 1.41	64.65 ± 1.29	70.06 ± 1.33
	watch	65.94 ± 1.41	68.23 ± 1.65	64.91 ± 1.53	67.99 ± 1.32	66.04 ± 2.47	67.53 ± 1.19

and the ACC and NMI values of SLSDR and SGFS on six corrupted datasets are shown in Tables 5 and 6, respectively.

It can be seen from Tables 5 and 6 that all the ACC and NMI values of SLSDR are higher than those of SGFS, showing the proposed SLSDR has great effectiveness. Additionally, the performance of SLSDR is hardly affected by outlier samples, which indicates that SLSDR is very robust to outlier samples.

3.4.5. Low redundancy test for the selected features

This experiment verifies that the inner product regularization term can select the representative and low-redundant features. Specifically, the page-blocks dataset from the UCI machine learning repository [64] is used as the test dataset. This dataset contains 5473 samples, each containing 10 features. Then, the proposed SLSDR is applied to feature selection to select important features. The correlations between these features can be obtained to evaluate the performance of inner product regularization term.

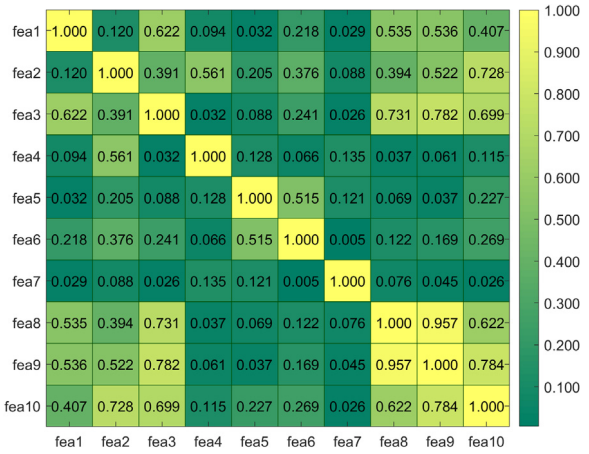
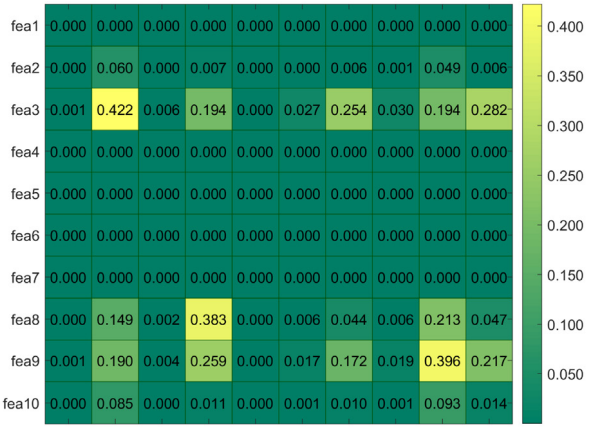
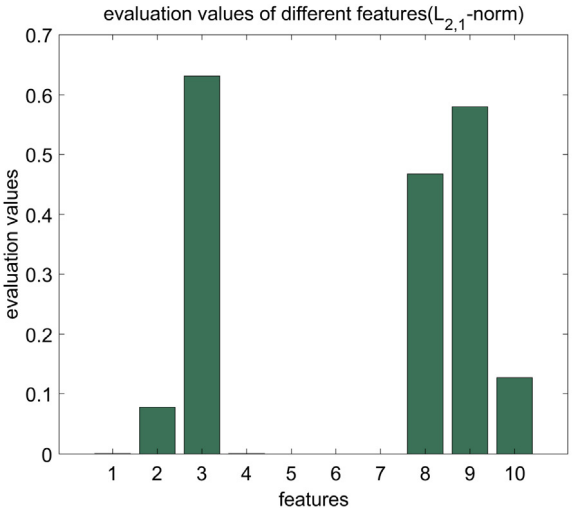


Fig. 7. The heat map of Pearson correlation coefficient matrix.



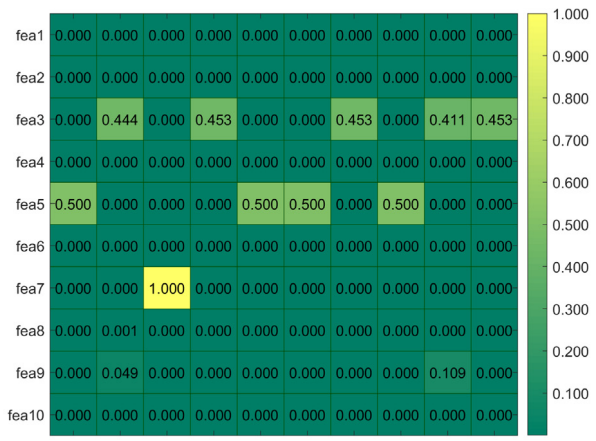
(a) The matrix $\boldsymbol{S}$ learned by the $l_{2,1}$-norm



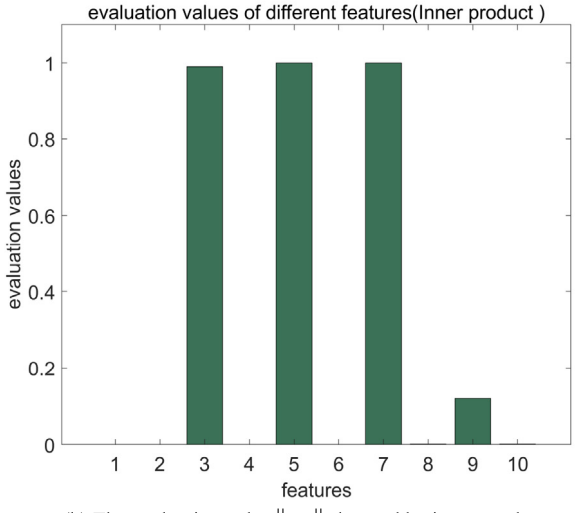
(b) The evaluation value $\| \boldsymbol{s}_i \|_2$ learned by the $l_{2,1}$-norm

Fig. 8. The matrix $\boldsymbol{S}$ and the evaluation value $\|\boldsymbol{s}_i\|_2$learned by the $l_{2,1}$-norm.

Meanwhile, the $l_{2,1}$-norm regularization term is used to replace the inner product regularization term in SLSDR, and feature selection is also performed to evaluate the performance of $l_{2,1}$-norm. The Pearson correlation coefficients are used to measure the



(a)The matrix $\boldsymbol{S}$ learned by inner product



(b) The evaluation value $\| \boldsymbol{s}_i \|_2$ learned by inner product

Fig. 9. The matrix $\boldsymbol{S}$ and the evaluation value $\|\boldsymbol{s}_i\|_2$learned by inner product.

correlations between different features in page-blocks dataset and the heat map of correlation coefficient matrix are shown in Fig. 7. The analyses and conclusions are given as follows.

The matrix $\boldsymbol{S}$ and the evaluation value $\|\boldsymbol{s}_i\|_2$learned by $l_{2,1}$-norm and inner product regularization term are shown in Figs. 8 and 9, respectively. It can be seen from Figs. 8 and 9(a) that the rows of matrix $\boldsymbol{S}$ are sparse, so the representative features can be selected. The evaluation values of all features are then calculated and arranged in descending order, and the features corresponding to the first three evaluation values are selected. As shown in Figs. 8 and 9(b), the features selected by $l_{2,1}$-norm are features 3, 8 and 9. It can be seen from Fig. 7 that the correlation coefficients between features 3 and 8, 3 and 9 and 8 and 9 are 0.731, 0.782 and 0.957, respectively. And the average value of the three correlation coefficients is 0.823. The features selected by the inner product regularization term are features 3, 5 and 7, and the obtained three correlation coefficients are 0.088, 0.026 and 0.121 with the average value 0.078. Therefore, the correlations between the features selected by inner product regularization term are significantly lower than those selected by the $l_{2,1}$-norm.

From the above analysis, it can be concluded that the $l_{2,1}$-norm can select the representative but high-redundant features, while the inner product regularization term can not only select

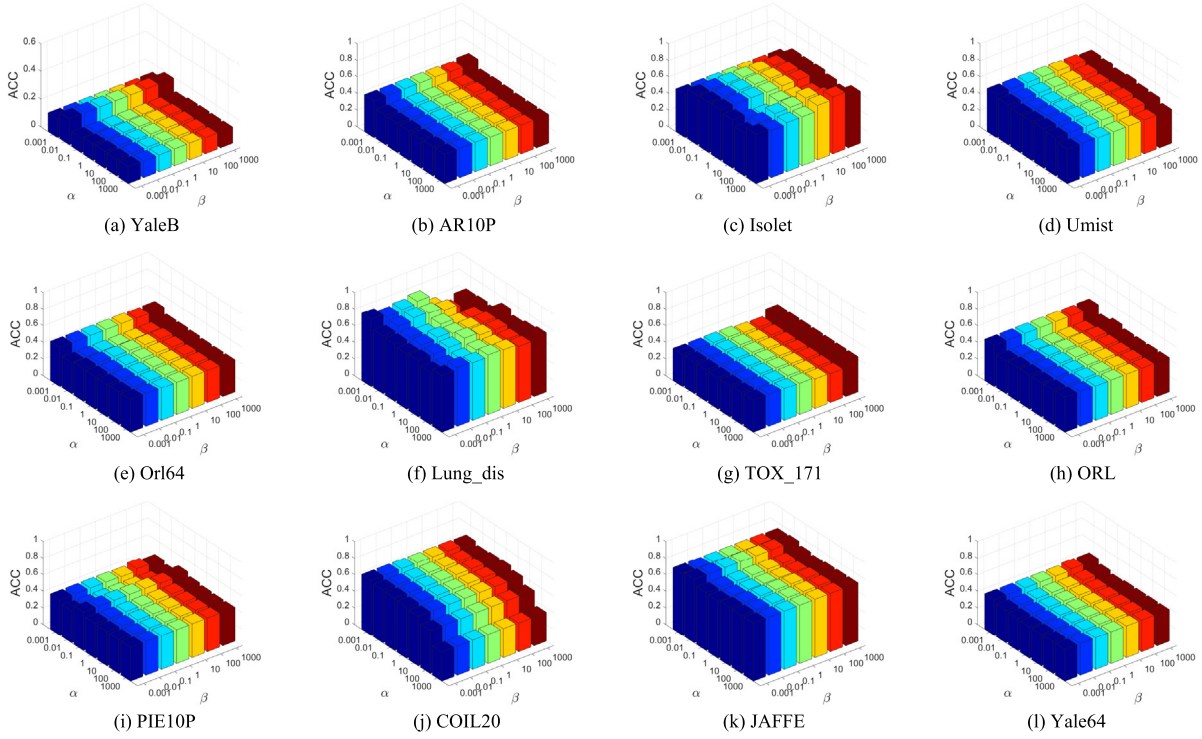


Fig. 10. Clustering accuracy of SLSDR on twelve datasets using different α and β.

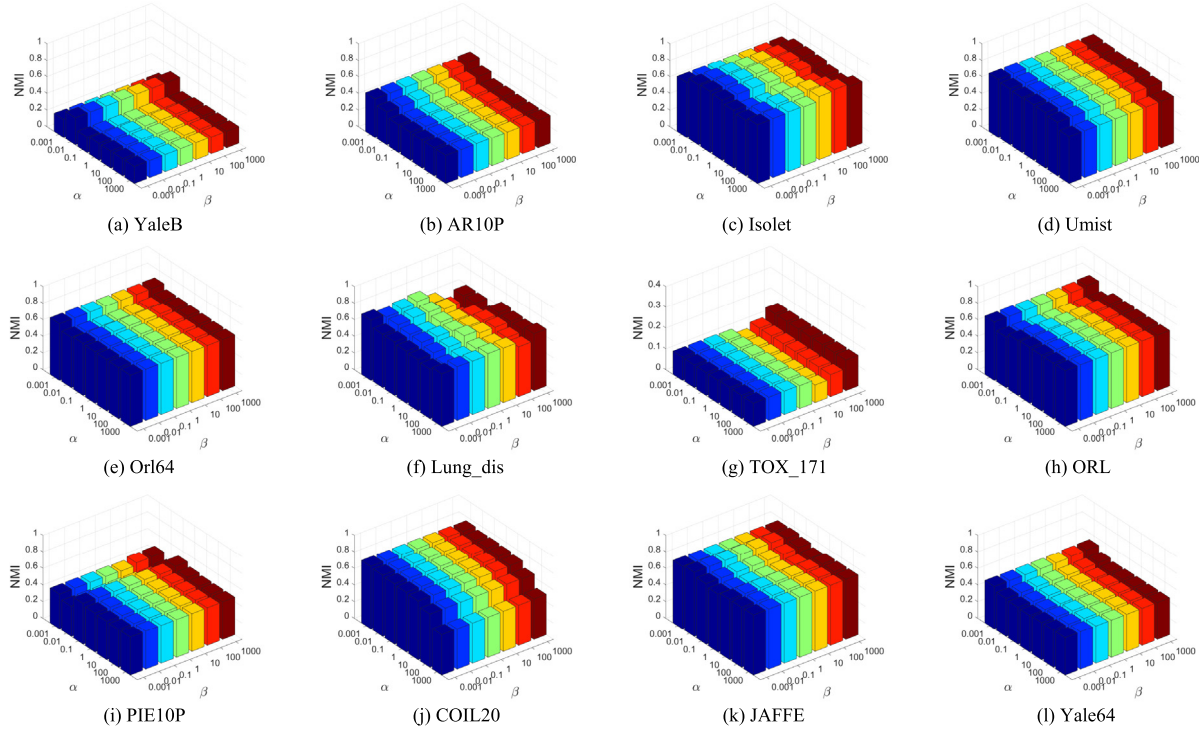


Fig. 11. Normalized Mutual Information of SLSDR on twelve datasets using different α and β.

the representative features, but also ensure the low redundancy of them.

3.4.6. Parameter sensitivity analysis

For the proposed SLSDR, the parameters that need to be adjusted include: the Gaussian scale parameter σ, the number of selected features l and the balance parameters α, β and λ. In this experiment, the sensitivity of parameters α and β is tested. The parameters α and β are searched in the range of $\{10^{-3},10^{-2},10^{-1},10^{+0},10^{+1},10^{+2},10^{+3}\}$ and the ACC and NMI values are obtained under the combination of each pair of parameters α and β. The three-dimensional histograms of ACC and NMI values on twelve datasets are shown in Figs. 10 and 11, respectively.

It can be seen from Figs. 10 and 11 that when the parameters α and β vary, the ACC and NMI values can keep relatively stable in most cases, especially for Umist, JAFFE and Yale64, so SLSDR is not sensitive to the parameters α and β.

The discussion of the experimental results can be summarized as follows. By analyzing the setting of the maximum number of iterations, it can be seen that SLSDR can achieve good convergence performance within 30 iterations. By discussing the setting of the k value, the experimental results show that when k is set to 5, not only can SLSDR obtain the best accuracy, but also the manifold structure information can be well preserved. It can be seen from the effectiveness evaluation experiment that the proposed SLSDR is effective and selects the most representative features. Additionally, compared with other six comparison algorithms, SLSDR obtains the best ACC and NMI values in most cases, indicating SLSDR is competitive. And the clustering result curves also show SLSDR has good performance. Then, it can be seen from the robustness test that SLSDR keeps effective and is very robust to outlier samples. From the low redundancy test, it can be found that the inner product regularization term can select the representative and low-redundant features. Meanwhile, SLSDR is not sensitive to the parameters α and β and keeps great performance. All the experimental results show SLSDR has great effectiveness.

4. Conclusions

In this paper, a novel algorithm has been proposed, called sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection (SLSDR). SLSDR is based on the framework of subspace learning-based graph regularized feature selection. For the proposed SLSDR, the inner product regularization term is first used to constrain the feature selection matrix $\boldsymbol{S}$. Since the sparsity of rows of $\boldsymbol{S}$ is guaranteed and the correlations between features are considered, the representative and low-redundant features are selected. It can be seen from the low redundancy experiment that the inner product regularization term is effective. Meanwhile, the data graph and feature graph are introduced into the framework of subspace learning simultaneously, so the local geometric structures are well preserved for both data and feature manifolds. It can be seen from the experiment results that SLSDR achieves the best accuracy in most cases, which fully demonstrates the effectiveness of SLSDR. Additionally, the $l_{2,1}$-norm is imposed on the residual matrix of subspace learning. And the experimental results show that SLSDR is very robust to outlier samples. A series of experimental results have shown that SLSDR has better performance than other compared algorithms.

In further studies, the following two aspects of tasks are hoped to be done. First, the adaptive graphs are hoped to be constructed to replace the fixed k-nearest neighbor graphs to better preserve the manifold structures information. Additionally, the alternating iterative update mechanism is prone to fall into local optimum, so the novel update mechanisms are hoped to be developed to achieve better optimization results.

Acknowledgments

We would like to express our sincere appreciation to the editors and the anonymous reviewers for their insightful comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Key R&D Program of China under Grants Nos. 2018YFC0825303 and 2018YFC0825305, the National Natural Science Foundation of China under Grants Nos. 61773304, 61836009, 61871306, 61772399 and U1701267, the Key Laboratory Fund, China 61421010402, and the Program for Cheung Kong Scholars and Innovative Research Team in University, China under Grant IRT1170.

References

[1] Y. Wan, X. Chen, J. Zhang, Global and intrinsic geometric structure embedding for unsupervised feature selection, Expert Syst. Appl. 93 (2018) 134–142.
[2] S. Feng, M.F. Duarte, Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation, Neurocomputing (2018).
[3] R. Shang, Z. Zhang, L. Jiao, et al., Self-representation based dual-graph regularized feature selection clustering, Neurocomputing 171 (2016) 1242–1253.
[4] R. Shang, W. Wang, R. Stolkin, et al., Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection, IEEE Trans. Cybern. 48 (2) (2018) 793–806.
[5] Y. Li, C. Lei, Y. Fang, et al., Unsupervised feature selection by combining subspace learning with feature self-representation, Pattern Recognit. Lett. 109 (2018) 35–43.
[6] J. Han, Z. Sun, H. Hao, Selecting feature subset with sparsity and low redundancy for unsupervised learning, Knowl.-Based Syst. 86 (2015) 210–223.
[7] S. Woo, C. Lee, Incremental feature extraction based on decision boundaries, Pattern Recognit. 77 (2018) 65–74.
[8] S. Du, Y. Ma, S. Li, et al., Robust unsupervised feature selection via matrix factorization, Neurocomputing 241 (2017) 115–127.
[9] Y. Zhang, Z. Zhang, J. Qin, et al., Semi-supervised local multi-manifold isomap by linear embedding for feature extraction, Pattern Recognit. 76 (2018) 662–678.
[10] M. Luo, F. Nie, X. Chang, et al., Adaptive unsupervised feature selection with structure regularization, IEEE Trans. Neural Netw. Learn. Syst. 29 (4) (2018) 944–956.
[11] K. Henni, N. Mezghani, C. Gouin-Vallerand, Unsupervised graph-based feature selection via subspace and pagerank centrality, Expert Syst. Appl. 114 (2018) 46–53.
[12] Q. Lu, X. Li, Y. Dong, Structure preserving unsupervised feature selection, Neurocomputing 301 (2018) 36–45.
[13] Y. Yang, H.T. Shen, Z. Ma, et al., 1-norm regularized discriminative feature selection for unsupervised learning, in: IJCAI proceedings-international joint conference on artificial intelligence, Vol. 22, (1), 2011, pp. 1589.
[14] M. Qi, T. Wang, F. Liu, et al., Unsupervised feature selection by regularized matrix factorization, Neurocomputing 273 (2018) 593–610.
[15] J. Wang, L. Wu, J. Kong, et al., Maximum weight and minimum redundancy: a novel framework for feature subset selection, Pattern Recognit. 46 (6) (2013) 1616–1627.
[16] T. Wu, Y. Zhou, R. Zhang, et al., Self-weighted discriminative feature selection via adaptive redundancy minimization, Neurocomputing 275 (2018) 2824–2830.
[17] W. He, X. Cheng, R. Hu, et al., Feature self-representation based hypergraph unsupervised feature selection via low-rank representation, Neurocomputing 253 (2017) 127–134.
[18] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 333–342.
[19] K. Zheng, X. Wang, Feature selection method with joint maximal information entropy between features and class, Pattern Recognit. 77 (2018) 20–29.
[20] J. Xu, B. Tang, H. He, et al., Semisupervised feature selection based on relevance and redundancy criteria, IEEE Trans. Neural Netw. Learn. Syst. 28 (9) (2017) 1974–1984.
[21] Z. Zeng, X. Wang, J. Zhang, et al., Semi-supervised feature selection based on local discriminative information, Neurocomputing 173 (2016) 102–109.
[22] Y. Wang, J. Wang, H. Liao, et al., An efficient semi-supervised representatives feature selection algorithm based on information theory, Pattern Recognit. 61 (2017) 511–523.
[23] X. Wang, X. Zhang, Z. Zeng, et al., Unsupervised spectral feature selection with l1-norm graph, Neurocomputing 200 (2016) 47–54.
[24] J.G. Dy, C.E. Brodley, Feature selection for unsupervised learning, J. Mach. Learn. Res. (2004) 845–889.
[25] S. Solorio-Fernández, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, A new unsupervised spectral feature selection method for mixed data: a filter approach, Pattern Recognit. 72 (2017) 314–326.
[26] J. Guo, W.W. Zhu, Dependence Guided Unsupervised Feature Selection, AAAI, 2018, pp. 2232–2239.
[27] P. Zhu, Q. Xu, Q. Hu, et al., Co-regularized unsupervised feature selection, Neurocomputing 275 (2018) 2855–2863.
[28] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, 2006, pp. 507–514.
[29] T. Hamed, R. Dara, S.C. Kremer, An accurate, fast embedded feature selection for SVMs, in: Machine Learning and Applications (ICMLA), in: 2014 13th International Conference on, IEEE, 2014, pp. 135–140.
[30] C. Tang, X. Zhu, J. Chen, et al., Robust graph regularized unsupervised feature selection, Expert Syst. Appl. 96 (2018) 64–76.

[31] S. Wang, W. Pedrycz, Q. Zhu, et al., Subspace learning for unsupervised feature selection via matrix factorization, Pattern Recognit. 48 (1) (2015) 10–19.
[32] S. Wang, W. Pedrycz, Q. Zhu, et al., Unsupervised feature selection via maximum projection and minimum redundancy, Knowl.-Based Syst. 75 (2015) 19–29.
[33] R. Shang, W. Wang, R. Stolkin, et al., Subspace learning-based graph regularized feature selection, Knowl.-Based Syst. 112 (2016) 152–165.
[34] C. Hou, F. Nie, D. Yi, et al., Feature selection via joint embedding learning and sparse regression, in: IJCAI Proceedings-International Joint Conference on Artificial Intelligence, Vol. 22, (1), 2011, pp. 1324.
[35] X. Fang, Y. Xu, X. Li, et al., Locality and similarity preserving embedding for feature selection, Neurocomputing 128 (2014) 304–315.
[36] S.T. Roweis, Saul L. K., Nonlinear dimensionality reduction by locally linear embedding, science 290 (5500) (2000) 2323–2326.
[37] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Advances in Neural Information Processing Systems, 2002, pp. 585–591.
[38] X. He, P. Niyogi, Locality preserving projections, in: Advances in Neural Information Processing Systems, 2004, pp. 153–160.
[39] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 1151–1157.
[40] Q. Ye, C. Zhao, S. Gao, et al., Weighted twin support vector machines with local information and its application, Neural Netw. 35 (2012) 31–39.
[41] Y. Xu, J. Yu, Y. Zhang, KNN-based weighted rough ν-twin support vector machine, Knowl.-Based Syst. 71 (2014) 303–313.
[42] X. Pan, Y. Luo, Y. Xu, K-nearest neighbor based structural twin support vector machine, Knowl.-Based Syst. 88 (2015) 34–44.
[43] A.K. Das, S. Das, A. Ghosh, Ensemble feature selection using bi-objective genetic algorithm, Knowl.-Based Syst. 123 (2017) 116–127.
[44] Y. Zhang, D. Gong, J. Cheng, Multi-Objective Particle Swarm Optimization Approach for Cost-Based Feature Selection in Classification, in: ACM Transactions on Computational Biology and Bioinformatics (TCBB), 14, (1) IEEE, 2017, pp. 64–75.
[45] Z. Yong, G. Dun-wei, Z. Wan-qiu, Feature selection of unreliable data using an improved multi-objective PSO algorithm, Neurocomputing 171 (2016) 1281–1290.
[46] Y. Xu, Y. Tian, X. Pan, et al., E-ENDPP: a safe feature selection rule for speeding up elastic net, Appl. Intell. 49 (2) (2019) 592–604.
[47] X. Pan, Y. Xu, A safe reinforced feature screening strategy for lasso based on feasible solutions, Inform. Sci. 477 (2019) 132–147.
[48] X. Pan, Z. Yang, Y. Xu, et al., Safe screening rules for accelerating twin support vector machine classification, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2018) 1876–1887.
[49] P. Zhu, W. Zuo, L. Zhang, et al., Unsupervised feature selection by regularized self-representation, Pattern Recognit. 48 (2) (2015) 438–446.
[50] C. Tang, X. Liu, M. Li, et al., Robust unsupervised feature selection via dual self-representation and manifold regularization, Knowl.-Based Syst. 145 (2018) 109–120.
[51] M. Qian, C. Zhai, Robust Unsupervised Feature Selection, IJCAI, 2013, pp. 1621–1627.
[52] Y. Meng, R. Shang, L. Jiao, et al., Feature selection based dual-graph sparse non-negative matrix factorization for local discriminative clustering, Neurocomputing 290 (2018) 87–99.
[53] F. Shang, L.C. Jiao, F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, Pattern Recognit. 45 (6) (2012) 2237–2250.
[54] D. Cai, X. He, J. Han, et al., Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.
[55] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556–562.
[56] W. Xu, Y. Gong, Document clustering by concept factorization, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2004, pp. 202–209.
[57] D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, IEEE Trans. Knowl. Data Eng. 23 (6) (2011) 902–913.
[58] Y. Yang, D. Xu, F. Nie, et al., Image clustering using local discriminant models and global integration, IEEE Trans. Image Process. 19 (10) (2010) 2761–2773.
[59] A. Rakhlin, A. Caponnetto, Stability of K-means clustering, in: Advances in Neural Information Processing Systems, 2007, pp. 1121–1128.
[60] H. Lu, Z. Fu, X. Shu, Non-negative and sparse spectral clustering, Pattern Recognit. 47 (1) (2014) 418–426.
[61] http://featureselection.asu.edu/datasets.php.
[62] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Courier Corporation, 1998.
[63] http://www.vision.caltech.edu/Image_Datasets/Caltech101/.
[64] https://archive.ics.uci.edu/ml/datasets.html.