Neurocomputing 517 (2023) 106–117

Contents lists available at ScienceDirect

Neurocomputing

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

Unsupervised feature selection via discrete spectral clustering and feature weights

Ronghua Shang [a], Jiarui Kong [a], Lujuan Wang [a], Weitong Zhang [a,*], Chao Wang [b], Yangyang Li [a], Licheng Jiao [a]

[a] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an, Shanxi Province 710071, China
[b] Research Center for Big Data Intelligence, Zhejiang Laboratory, Hangzhou, Zhejiang Province 311121, China

ARTICLE INFO

Article history:
Received 22 December 2021
Revised 24 July 2022
Accepted 24 October 2022
Available online 29 October 2022
Communicated by Zidong Wang

Keywords:
Unsupervised feature selection
Discrete spectral clustering
Feature weights
Orthogonal regression

ABSTRACT

Most of the existing unsupervised feature selection methods learn the cluster structure through spectral clustering, and then use various regression models to introduce the data matrix into the indicator matrix to obtain feature selection matrix. In these methods, the clustering indicator matrix is usually continuous value, which is not the best choice for the matrix in terms of its supervising role in feature selection. Based on this, unsupervised feature selection via discrete spectral clustering and feature weights (FSDSC) is proposed in this paper. First, FSDSC integrates regression model and spectral clustering in a unified framework for feature selection, and introduces a feature weight matrix, which intuitively expresses the importance of each feature with its diagonal elements. Compared with the common feature selection matrix that requires constraints such as sparse regular items, the appearance of the feature weight matrix reduces the complexity of the model and simplifies the calculation process of feature evaluation. Secondly, for the value of the indicators matrix, the spectral clustering is improved to obtain a discrete clustering indicator matrix, which provides clearer guidance information for feature selection. Finally, in order to avoid trivial solutions, the transformation matrix is constrained by orthogonal constraint. The combination of the orthogonal regression model and spectral clustering enables the algorithm to perform feature selection and manifold information learning at the same time, thereby preserving the local geometric structure of data. Compared with other excellent unsupervised feature selection algorithms, the experimental results prove the effectiveness of the proposed algorithm.

1. Introduction

With the rapid development of information technology, a large amount of high-dimensional data has been generated in various fields. These high-dimensional data always contain some noise and redundant features, which increase the difficulty of data processing [1]. The research in the fields of machine learning, image processing, data mining [2], and pattern recognition [3] is used to process the high-dimensional data. It is necessary to reduce the dimensionality of high-dimensional data [4]. Dimension reduction can not only decrease the time cost of calculation and improve calculation efficiency, but also reduce the space pressure of calculation. Feature extraction [5] and feature selection [6] are two common dimensionality reduction methods. Feature extraction converts all features into fewer new features to replace the original features [7]. Feature selection selects some representative features to form a subset according to a certain standard, to obtain a compressed data representation [8]. In addition, it can retain the semantic information of data [9] and has stronger interpretability [10]. Nowadays, feature selection is constantly developing [11]. However, most of the high-dimensional data in real life are unlabeled, which is difficult for direct feature selection. Therefore, how to obtain pseudo-labels that are closer to the real labels is a challenging research. This paper proposes an unsupervised feature selection method via discrete spectral clustering and feature weights (FSDSC), which uses a discrete clustering indicator matrix as a pseudo-label to provide clearer discriminative information for feature selection. At the same time, the feature subset is extracted based on the feature weight matrix. Compared with traditional methods that need to impose constraints on the feature selection matrix, such as sparse regularization item, the feature weight matrix reduces the complexity of model and reduces the calculation amount of feature evaluation. Specifically, FSDSC integrates regression models and spectral clustering in a unified framework, introduces a feature weight matrix in the framework, and then

* Corresponding author.
E-mail address: wtzhang_1@xidian.edu.cn (W. Zhang).

https://doi.org/10.1016/j.neucom.2022.10.053
0925-2312/

performs feature selection through this framework. The matrix is a diagonal matrix, each diagonal element of which intuitively represents the weight of each feature, so that the algorithm can easily select a subset of features. Orthogonal constraint is imposed on the transformation matrix. Compared with least square regression, orthogonal regression model can preserve more discriminative information and avoid trivial solutions. In addition, FSDSC improves the spectral clustering method to obtain a discrete indicator matrix, which provides more accurate guidance information for feature selection. The rest of this paper is organized as follows. Section 2 is mainly about the introduction of some related work. Section 3 introduces the proposed model, optimization method and convergence analysis in detail. Section 4 presents the experimental results and comparative analysis of FSDSC and the compared algorithms on the same datasets. The conclusion is summarized in Section 5.

2. Related work

At present, feature selection methods can be divided into supervised feature selection, semi-supervised feature selection, and unsupervised feature selection according to whether label information is needed [12]. Supervised methods use the correlation between sample and label information to select discriminative features, which are beneficial to the classification of samples [13]. Some labels are needed in the semi-supervised methods [14]. Most of these methods combine labeled and unlabeled data, and select feature subset by constructing a similarity matrix [15]. However, most real datasets lack label information, and the cost of labeling large-scale data is high. Therefore, unsupervised methods are very important which select important features based on the inherent information of data.

Unsupervised feature selection methods can be divided into filter unsupervised feature selection [16], wrapper unsupervised feature selection [17] and embedded unsupervised feature selection [18] according to the search strategy. Filter methods utilize the statistical characteristics or inherent attributes of data to rank the importance of features, instead of relying on other algorithms, and then select the top-ranked features to form a subset [16]. Filter methods can remove part of the noise to make search easier, but the performance of which is relatively poor. Wrapper methods use specific learning algorithms to evaluate the generated feature subsets. Compared with the former, the computational cost of these methods is higher. When processing a larger dataset, the computational complexity of which increases exponentially [19]. Therefore, the wrapper methods are not suitable for large-scale data dimensionality reduction. Embedded methods combine the advantages of filter and wrapper methods, which aim to select a better feature subset at a lower computational cost [20]. These methods combine the search and training processes to explore the correlation between features [21]. Because of the good performance for feature selection, embedded methods have received extensive attention [22]. At the same time, how to get a clustering indicator matrix in the feature selection has become a problem [23].

In order to solve this problem, the correlation between features has been explored as label information to guide feature selection, and many criteria for evaluating feature correlation have been proposed [22]. Among them, a common criterion is to find the clustering indicator through clustering algorithms and turn the problem into a supervised problem. Based on this, unsupervised feature selection can be divided into two types. One type is to find clustering metrics first, and then feature selection is performed. Moreover, these two steps are repeated until a certain conditional condition is met. For example, multi-cluster feature selection (MCFS) proposed by Cai et al. used the two-step strategy. First, clustering indicators were obtained by spectral clustering, and then the indicator matrix was used to perform feature selection. Thus, the cluster structure of data was preserved [24]. Another type of unsupervised feature selection adopts an integrated form, which searches for clustering indicators while performing feature selection. For example, Hou et al. proposed joint embedding learning and sparse regression for feature selection (JELSR), which adopted a single-step strategy to combine low-dimensional spectral embedding learning and sparse regression into an objective function. JELSR effectively improved the performance of feature selection [25]. Li et al. proposed generalized uncorrelated regression with adaptive graph for unsupervised feature selection (URAFS), which combined spectral clustering and generalized uncorrelated regression in a unified framework. The construction and optimization of graph were adaptively realized. Compared with the former, the latter performed clustering learning and feature evaluation simultaneously in the optimization process, which improved the learning performance of the algorithm [26].

Cluster analysis is a common auxiliary method for unsupervised feature selection, which groups samples according to the inherent attributes of data. Studies have shown that the distribution of high-dimensional data is not a uniform linear distribution [27]. On the contrary, the distribution is often sparse. Therefore, high-dimensional data contains a lot of local information, which is beneficial to explore the internal structure of data and improve the learning performance of the algorithm for nonlinear structures [28]. Based on local information, many manifold learning algorithms have been proposed, such as Laplacian Eigenmap (LE) [29], Locality Preserving Projections (LPP) [30], and Local Linear Embedding (LLE) [31] and so on. The spectral graph theory has shown good performance when it was applied to describe the manifold structure of data [32]. Specifically, spectral clustering learnt the geometric structure information of data through the spectral graph theory, and obtained a good clustering result [33]. Later, spectral clustering was applied to unsupervised feature selection, and the clustering result was used as a pseudo-label for the unsupervised problem, so that the problem became supervised. Nonnegative discriminative feature selection (NDFS) proposed by Li et al. utilized spectral clustering to learn the clustering indicators of the input samples and performed feature selection at the same time [34]. The joint learning of clustering label and feature selection matrix enabled NDFS to select discriminative features. Wang et al. embedded clustering algorithm into feature selection through sparse learning, then proposed embedded unsupervised feature selection (EUFS), and proved the effectiveness of EUFS with experimental results [35]. Zhu et al. embeded a block regularizer in the multiview multilabel (MVML) learning for multiview image classification (MVML) framework to perform view selection and select information views. Moreover, feature selection was performed to select information features from the information view [36]. Zhu et al. proposed a low-rank sparse subspace clustering algorithm (LSS) by dynamically learning an affinity matrix from a low-dimensional space of raw data [37]. Zhu et al. proposed a one-step multi-view spectral clustering (OMSC) by outputting a common affinity matrix as the final clustering result [38]. Zhu et al. embeded graph regularization into a joint sparse regression framework, used dictionary learning to generate the basis of the dataset, and used the proposed method to map the original data into the basis space to generate new representations. Moreover, robust joint graph sparse coding for unsupervised spectral feature selection (JGSC) was proposed [39]. Zhu et al. used spectral clustering to preserve the local and global structure of features and samples. In addition, they proposed local and global structure preservation for robust unsupervised spectral feature selection [40]. The method represented each feature with

other features to preserve the local structure of the feature, and low-rank constraint was imposed on the weight matrix to preserve the global structure between samples and features. Zheng et al. added a self-paced regularization model to the sparse feature selection model to reduce the effect of outliers on feature selection, and proposed unsupervised feature selection by self-paced learning regularization [41].

The models of the above feature selection algorithms are least squares regression. Their goal is to find a projection matrix $\mathbf{W}$ as the feature selection matrix and minimize the square error function [42]. However, this may make the model more complicated. Different from traditional methods, supervised feature selection with orthogonal regression and feature weighting (FSOR) proposed by Wu et al. introduced a feature weight matrix for the importance of features sort [43]. The scale factor in the feature weight matrix represented the level or proportion of each feature, which was used to minimize the vertical distance from the data point to the fitting function. Therefore, the scale factor can be used to evaluate the importance of features. Later, Xu et al. proposed a two-stage feature selection method based on FSOR and mRMR [44], called feature selection under orthogonal regression with redundancy minimizing (ORMR) [45]. These two methods can intuitively represent the importance of each feature and reduce the model complexity. However, both of them are not suitable for supervised feature selection. Meanwhile, considering that the matrix plays a supervisory role in feature selection, it is not the best choice for the indicator matrix. On the contrary, discrete variables have clearer guidance information. The real class labels are each independent and discrete. Therefore, discrete clustering indicator is closer to the real class labels than continuous clustering indicator. In addition, discrete clustering indicator can better characterize the independence between labels. Therefore, in this paper, a discrete clustering index matrix is introduced and the importance of each feature is directly represented with a feature weight matrix. Then, unsupervised feature selection via discrete spectral clustering and feature weights (FSDSC) is proposed. The main contributions are summarized as follows:

1) The feature weight matrix is used to directly express the importance of each feature. This matrix reduces the complexity of the model, simplifies the calculation process of feature evaluation, and makes it easier to select a suitable feature subset. 2) By improved spectral clustering, a discrete clustering indicator matrix is obtained, so as to realize the supervising function of the indicator matrix in feature selection. 3) The combination of orthogonal regression and spectral clustering enables the simultaneous realization of feature selection and manifold information learning, thereby preserving the local geometric structure of data. At the same time, orthogonal constraint also avoids trivial solutions.

3. The proposed method

In this section, FSDSC in detail is introduced, which is mainly composed of two parts: the orthogonal regression model with feature weights and discrete spectral clustering. In the proposed algorithm, the dataset is represented by a matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, where n represents the number of samples, and d represents the dimension of samples, that is, the number of features. In addition, let c be the number of categories, l be the number of selected features, and $l \ll d$.

3.1. Uncorrelated feature selection via sparse latent representation

Due to the lack of label information, unsupervised feature selection is regarded as a difficult problem. According to the learned clustering indicator, unsupervised feature selection can be regarded as a regression model like a supervised method. Therefore, in the past few years, regression models have been widely used in unsupervised feature selection [46]. Define the input dataset as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \in \mathbb{R}^{d\times n}$, where $\mathbf{x}_i$ represents the ith sample. The classic regression model can be expressed as:

$$\min_{\mathbf{W},\mathbf{F},\mathbf{b}} ||\mathbf{X}^T\mathbf{W} + \mathbf{1}_n\mathbf{b}^T - \mathbf{F}||_F^2, \tag{1}$$

where, $\mathbf{W} \in \mathbb{R}^{d\times c}$ represents the subspace, $\mathbf{b} \in \mathbb{R}^{c\times 1}$ represents the deviation, $\mathbf{F} \in \mathbb{R}^{n\times c}$ represents the pseudo-label matrix, that is, the learned cluster structure. By adding orthogonal constraint in (1), the statistical characteristics of the input data are preserved while avoiding trivial solutions. In addition, the orthogonal constraint is realized with the manifold structure, so the geometric structure of data is preserved during the projection process [47]. The orthogonal regression model is defined as:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{F},\mathbf{b}} ||\mathbf{X}^T\mathbf{W} + \mathbf{1}_n\mathbf{b}^T - \mathbf{F}||_F^2, \\ &s.t.\ \mathbf{W}^T\mathbf{W} = \mathbf{I}_c. \end{aligned} \tag{2}$$

A feature weight in the orthogonal regression model (2) is introduced to measure the importance of all features. Define the diagonal matrix $\Phi \in \mathbb{R}^{d\times d}$ as the feature weight matrix, and its diagonal is the vector $\boldsymbol{\varphi}$. The scale factor $\boldsymbol{\varphi}_j \geqslant 0 (1 \leqslant j \leqslant d)$ represents the weight of the jth feature, so $\boldsymbol{\varphi}^T\mathbf{1}_d = 1$. Specifically, the orthogonal regression model with feature weights is as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{F},\boldsymbol{\varphi},\mathbf{b}} ||\mathbf{X}^T\Phi\mathbf{W} + \mathbf{1}_n\mathbf{b}^T - \mathbf{F}||_F^2, \\ &s.t.\ \mathbf{W}^T\mathbf{W} = \mathbf{I}_c, \boldsymbol{\varphi}^T\mathbf{1}_d = 1, \boldsymbol{\varphi} \geqslant 0, \end{aligned} \tag{3}$$

$\boldsymbol{\varphi}$ in (3) is used to evaluate the importance of d features, and rank the values of scale factors in $\boldsymbol{\varphi}$. The k most important features are selected based on the k largest values.

3.2. Discrete spectral clustering

In (3), $\mathbf{F}$ is unknown and needs to be learned through clustering structure learning. Many studies on graph spectral theory have found that it can effectively retain the local structure of data through the nearest neighbor graph [48], thereby obtaining discriminative information to improve the accuracy of feature selection [49]. Therefore, generalized spectral clustering is utilized to learn the geometric structure of data, thereby obtaining a clustering indicator matrix to guide feature selection. Define $\mathbf{F}$ to represent the clustering structure of dataset in the c dimensional subspace. If the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar in the original space, then $\mathbf{f}_i$ and $\mathbf{f}_j$ are also similar in the subspace. The Gaussian function is used to measure the similarity between two samples. The definition of similarity matrix is defined as follows:

$$\mathbf{S}_{ij} = \begin{cases} \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2/2\sigma^2\right) & if\ \mathbf{x}_i \in N(\mathbf{x}_i), \\ & \text{or } \mathbf{x}_j \in N(\mathbf{x}_j), \\ 0 & \text{otherwise}, \end{cases} \tag{4}$$

where, $i, j = 1, 2, \ldots, n, N(\mathbf{x}_i)$ represents the set of k-nearest neighbors of the sample $\mathbf{x}_i$. Based on (4), the degree matrix $\mathbf{D}$ and the Laplacian matrix $\mathbf{L}$ can be obtained, where $\mathbf{L} = \mathbf{D}\text{-}\mathbf{S}$. The objective function of generalized spectral clustering is expressed as follows:

$$\begin{aligned} &\min_{\mathbf{F}} \sum_{i=1}^{n}\sum_{j=1}^{n} ||\mathbf{f}_i - \mathbf{f}_j||^2\mathbf{S}_{ij} = Tr\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right), \\ &s.t.\ \mathbf{F}^T\mathbf{D}\mathbf{F} = \mathbf{I}_c, \end{aligned} \tag{5}$$

where, in order to ensure that the problem (5) is solvable, the constraint $\mathbf{F}^T\mathbf{D}\mathbf{F} = \mathbf{I}_c$ is introduced. Considering that the indicator

matrix F will provide guidance information in the regression model, a discrete clustering indicator matrix is more suitable than a continuous one. In order to obtain an ideal matrix, discrete constraint is added in (5) to improve the effect of generalized spectral clustering [50].

$$\begin{aligned} &\min_{\mathbf{Y},\mathbf{F}} Tr\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right), \\ &s.t.\ \mathbf{Y}\in Ind, \mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}, \end{aligned} \tag{6}$$

where, *Ind* indicates that $\mathbf{Y}$ is a binary matrix composed of 0 and 1, that is $\mathbf{Y}\in\{0,1\}^{n\times c}$. At the same time, the matrix $\mathbf{Y}$ satisfies $\mathbf{Y}\mathbf{1}_c=\mathbf{1}_n$, that is, only one element in each row of $\mathbf{Y}$ is 1, and the other elements are 0. $\mathbf{y}_{ij}$ = 1 means that the ith sample is assigned to the jth class. $\mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}$ means that $\mathbf{F}$ is obtained by scaling of $\mathbf{Y}$, so the matrix $\mathbf{F}$ is discrete and satisfies the constraint $\mathbf{F}^T\mathbf{D}\mathbf{F}=\mathbf{F}_c$.

3.3. Objective function

Most unsupervised feature selection methods obtain the indicator matrix through spectral clustering learning, and then use various regression models to introduce the data matrix into the indicator matrix to obtain the feature selection matrix [22]. Similar to traditional methods, discrete spectral clustering is used to learn the clustering structure of data and obtain a discrete clustering indicator matrix. At the same time, the matrix is used as a pseudo-label of the orthogonal regression model to turn the problem into supervised. Specifically, by combining (3) and (6), the following objective function is obtained:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{F},\mathbf{Y},\boldsymbol{\varphi},\mathbf{b}} ||\mathbf{X}^T\Phi\mathbf{W}+\mathbf{1}_n\mathbf{b}^T-\mathbf{F}||_F^2+\alpha Tr\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right), \\ &s.t.\ \mathbf{W}^T\mathbf{W}=\mathbf{I}_c, \boldsymbol{\varphi}^T\mathbf{1}_d=1,\ \boldsymbol{\varphi}\geqslant 0, \\ &\quad \mathbf{Y}\in Ind, \mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}, \end{aligned} \tag{7}$$

where, α is the balance parameter, which is used to control the weight of the second item. It is worth noting that the deviation $\mathbf{b}$ is not restricted in (7). Therefore, (7) can be simplified by the extreme condition of $\mathbf{b}$, and obtain the Lagrangian equation for $\mathbf{b}$:

$$L(\mathbf{b})=||\mathbf{X}^T\Phi\mathbf{W}+\mathbf{1}_n\mathbf{b}^T-\mathbf{F}||_F^2+\Re(\mathbf{W},\mathbf{F},\mathbf{Y},\boldsymbol{\varphi}), \tag{8}$$

where, $\Re(\mathbf{W},\mathbf{F},\mathbf{Y},\varphi)$ represents a term in the Lagrange equation that has nothing to do with $\mathbf{b}$. According to the Karush–Kuhn–Tucker (KKT) theory [51], when $\partial L(\mathbf{b})/\partial\mathbf{b}=0$, the solution of $\mathbf{b}$ in the objective function (7) is obtained,

$$\frac{\partial||\mathbf{X}^T\Phi\mathbf{W}+\mathbf{1}_n\mathbf{b}^T-\mathbf{F}||_F^2}{\partial\mathbf{b}}=0, \tag{9}$$

So the value of $\mathbf{b}$ is expressed as:

$$\mathbf{b}=\frac{1}{n}\left(\mathbf{F}^T-\mathbf{W}^T\Phi\mathbf{X}\right)\mathbf{1}_n, \tag{10}$$

According to (10), the objective function (7) can be simplified as

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{F},\mathbf{Y},\boldsymbol{\varphi}} ||\mathbf{H}\left(\mathbf{X}^T\Phi\mathbf{W}-\mathbf{F}\right)||_F^2+\alpha Tr\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right), \\ &s.t.\ \mathbf{W}^T\mathbf{W}=\mathbf{I}_c, \boldsymbol{\varphi}^T\mathbf{1}_d=1, \boldsymbol{\varphi}\geqslant 0, \\ &\quad \mathbf{Y}\in Ind, \mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}, \end{aligned} \tag{11}$$

where, $\mathbf{H}=\mathbf{I}_n-(1/n)\mathbf{1}_n\mathbf{1}_n^T$. The final objective function is as shown in (11). Next, how to solve (11) is introduced.

3.4. Optimization process

The objective function (11) involves four variables $\mathbf{W},\mathbf{F},\mathbf{Y}$ and $\boldsymbol{\varphi}$, where the variable $\mathbf{F}$ is the scaling of $\mathbf{Y}$. Therefore, the optimization of $\mathbf{F}$ and $\mathbf{Y}$ is same. In this paper, an alternate optimization strategy is used to solve the objective function (11), and each variable is solved iteratively according to the following steps: A. Update $\mathbf{W}$ When the variables $\mathbf{F},\mathbf{Y}$ and $\boldsymbol{\varphi}$ are fixed, solving $\mathbf{W}$ in function (11) is equivalent to solving the following problem:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_c} ||\mathbf{H}\left(\mathbf{X}^T\Phi\mathbf{W}-\mathbf{F}\right)||_F^2, \tag{12}$$

For any matrix $\mathbf{M}$, $||\mathbf{M}||_F^2=Tr\left(\mathbf{M}\mathbf{M}^T\right)$, then (12) satisfies the following process:

$$\begin{aligned} &\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_c} ||\mathbf{H}\left(\mathbf{X}^T\Phi\mathbf{W}-\mathbf{F}\right)||_F^2, \\ &\Longleftrightarrow \min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_c} Tr\left(\mathbf{W}^T\Phi\mathbf{X}\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}-2\mathbf{W}^T\Phi\mathbf{X}\mathbf{H}\mathbf{F}\right), \\ &\Longleftrightarrow \min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_c} Tr\left(\mathbf{W}^T\mathbf{A}\mathbf{W}-2\mathbf{W}^T\mathbf{B}\right), \end{aligned} \tag{13}$$

Algorithm 1: GPI method.

Input: Symmetric matrix $\mathbf{A}\in\mathbb{R}^{d\times d}$, matrix $\mathbf{B}\in\mathbb{R}^{d\times c}$;
Initialization: Randomly initialize the matrix $\mathbf{W}\in\mathbb{R}^{d\times c}$ which satisfies $\mathbf{W}^T\mathbf{W}=\mathbf{I}_c$. Initialize the parameter ν, so that $\widetilde{\mathbf{A}}=\nu\mathbf{I}_d-\mathbf{A}$ is a positive definite matrix;
Repeat:
1. Update $\mathbf{R}\leftarrow\widetilde{\mathbf{A}}\mathbf{W}+2\mathbf{B}$;
2. Calculate $\mathbf{U}\mathbf{G}\mathbf{V}^T=\mathbf{R}$ according to the compressed SVD method of $\mathbf{R}$;
3. Update $\mathbf{W}\leftarrow\mathbf{U}\mathbf{V}^T$;

Until convergence;
Output: $W\in\mathbb{R}^{d\times c}$.

where $\mathbf{B}=\Phi\mathbf{X}\mathbf{H}\mathbf{X}^{\mathbf{T}}\Phi, \mathbf{B}=\Phi\mathbf{X}\mathbf{H}\mathbf{F}$. Function (13) is the same as the quadratic problem of Stiefel manifold (QPSM) [52], which can be solved by the generalized power iteration (GPI) method proposed by Nie et al. [53]. Algorithm 1 introduces the details of GPI. B. Update $\mathbf{F},\mathbf{Y}$ In (11), the process of solving variables $\mathbf{F}$ and $\mathbf{Y}$ is same. Therefore, when the variables $\mathbf{W}$ and $\boldsymbol{\varphi}$ are fixed, (11) is simplified as follows:

$$\min_{\mathbf{Y}\in Ind,\mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}} ||\mathbf{H}\left(\mathbf{X}^T\Phi\mathbf{W}-\mathbf{F}\right)||_F^2+\alpha Tr\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right), \tag{14}$$

Since $\mathbf{F}^T\mathbf{D}\mathbf{F}=\mathbf{I}_c$ and $\mathbf{L}=\mathbf{D}$-$\mathbf{S}$, (14) can be rewritten as

$$\begin{aligned} \min_{\mathbf{Y}\in Ind,\mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}} &Tr\left(\mathbf{F}^T\mathbf{H}\mathbf{F}\right)-2Tr\left(\mathbf{F}^T\mathbf{H}\mathbf{X}^0T\Phi\mathbf{W}\right) \\ &-\alpha Tr\left(\mathbf{F}^T\mathbf{S}\mathbf{F}\right), \end{aligned} \tag{15}$$

(15) is further transformed into

$$\begin{aligned} &\max_{\mathbf{Y}\in Ind,\mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}} 2Tr\left(\mathbf{F}^T\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}\right)-Tr\left(\mathbf{F}^T\mathbf{H}\mathbf{F}\right) \\ &\qquad +\alpha Tr\left(\mathbf{F}^T\mathbf{S}\mathbf{F}\right), \\ &\Longleftrightarrow \max_{\mathbf{Y}\in Ind,\mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}} Tr\left(\mathbf{F}^T\mathbf{P}\mathbf{F}\right)+2Tr\left(\mathbf{F}^T\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}\right), \end{aligned} \tag{16}$$

where, $\mathbf{P}=\alpha\mathbf{S}-\mathbf{H}+\lambda\mathbf{I}_n$, and the value of λ is larger to ensure that matrix $\mathbf{P}$ is a positive semi-definite matrix. The process of solving problem (16) is mainly divided into two steps: 1) Define $\mathbf{Q}=\mathbf{PF}+\mathbf{HX}^T\Phi\mathbf{W}$. 2) Eq. (16) is equivalent to the following function:

$$\max_{\mathbf{Y}\in Ind,\mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{DY}\right)^{-1/2}} Tr\left(\mathbf{F}^T\mathbf{Q}\right), \tag{17}$$

The above two steps are performed iteratively until the function (14) converges, and the solution of (14) is completed. The most critical step in the above is to solve the problem (17). Here, we learn from the method proposed by Zhang et al. [50]. According to $\mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{DY}\right)^{-1/2}$, (17) is equivalent to

$$\max_{\mathbf{Y}\in Ind}\sum_{j=1}^{c}\frac{\mathbf{y}_j^T\mathbf{q}_j}{\sqrt{\mathbf{y}_j^T\mathbf{Dy}_j}}, \tag{18}$$

where, $\mathbf{y}_j$ and $\mathbf{q}_j$ represent the jth column of $\mathbf{Y}$ and $\mathbf{Q}$ respectively. Since all rows of $\mathbf{Y}$ are involved, one row is solved firstly, the remaining rows are fixed. Then update row by row. Assume that the solution $\overline{\mathbf{Y}}$ for one iteration has been obtained, when solving the ith row ($\mathbf{y}^i$) of $\mathbf{Y}$, only the increment Δ_{ij} of the objective function (18) need to be considered when $\overline{y}_{ij}$ changes from 0 to 1. The increment Δ_{ij} determines $\mathbf{y}_{ij}$, which is calculated as

Algorithm 2: Algorithm to Solve Problem (16).

Input: Data matrix $\mathbf{X}\in\mathbb{R}^{d\times n}$, parameters α, similarity matrix $\mathbf{S}\in\mathbb{R}^{n\times n}$, feature weight matrix $\Phi\in\mathbb{R}^{d\times d}$, projection matrix $\mathbf{W}\in\mathbb{R}^{d\times c}$;

Initialization: Randomly initialize the matrix $\mathbf{Y}\in\{0,1\}^{n\times c}$ satisfy $\mathbf{Y1}_c=\mathbf{i}_n$, and $\mathbf{F}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{DY}\right)^{-1/2}$, $\mathbf{H}=\mathbf{I}_n-(1/n)\mathbf{1}_n\mathbf{1}_n^T$, the initial value of the parameter λ is large enough to ensure that $\mathbf{P}=\alpha\mathbf{S}-\mathbf{H}+\lambda\mathbf{I}_n$ is a positive semi-definite matrix;

Repeat:

1. Update $\mathbf{Q}=\mathbf{PF}+\mathbf{HX}^T\Phi\mathbf{W}$;
2. Update $\mathbf{Y}$ according to (20);
3. Update $\mathbf{F}\leftarrow\mathbf{Y}\left(\mathbf{Y}^T\mathbf{DY}\right)^{-1/2}$;

Until convergence;

Output: Matrix $\mathbf{F}$ and $\mathbf{Y}$.

$$\Delta_{ij}=\frac{\overline{\mathbf{y}}_j^T\mathbf{q}_j+q_{ij}\left(1-\overline{y}_{ij}\right)}{\sqrt{\overline{\mathbf{y}}_j^T\mathbf{D}\overline{\mathbf{y}}_j+d_{ii}\left(1-\overline{y}_{ij}\right)}}-\frac{\overline{\mathbf{y}}_j^T\mathbf{q}_j-\overline{y}_{ij}q_{ij}}{\sqrt{\overline{\mathbf{y}}_j^T\mathbf{D}\overline{\mathbf{y}}_j-d_{ii}\overline{y}_{ij}}}, \tag{19}$$

where $d_{ii}=1$ represents the ith element on the diagonal of matrix $\mathbf{D}$. According to Δ_{ij}, $\mathbf{y}_{ij}$ can be determined:

$$y_{ij}=<j=\arg\max_{j'\in[1,c]}\Delta_{ij'}>, \tag{20}$$

where, $\mathbf{y}_{ij}$ when the parameter in the $<\bullet>$ is true, otherwise $\mathbf{y}_{ij}=0$. In other words, in each row of matrix $\mathbf{Y}$, when function (18) is the largest, the corresponding element is 1, and the remaining elements are 0. Algorithm 2 describes the steps for solving (16) in detail. C. Update $\boldsymbol{\varphi}$ When the variables $\mathbf{W},\mathbf{F}$ and $\mathbf{Y}$ are fixed, function (11) is equivalent to:

$$\begin{aligned}&\min_{\varphi^T\mathbf{1}_d=1,\varphi\geqslant 0}||\mathbf{H}\left(\mathbf{X}^T\Phi\mathbf{W}-\mathbf{F}\right)||_F^2,\\ \Longleftrightarrow&\min_{\varphi^T\mathbf{1}_d=1,\varphi\geqslant 0}Tr\left(\Phi\mathbf{XHX}^T\Phi\mathbf{WW}^T-2\Phi\mathbf{XHFW}^T\right).\end{aligned} \tag{21}$$

Lemma 1. *If $\mathbf{O}$ is a diagonal matrix, then $Tr(\mathbf{OZOM})=\mathbf{O}^T\left(\mathbf{Z}^T\circ\mathbf{M}\right)\mathbf{O}$.*

Prove:

$$\begin{aligned}&Tr(\mathbf{OZOM})=\mathbf{O}^T diag(\mathbf{ZOM})=\mathbf{O}^T vec\left\{\mathbf{z}_i^T\mathbf{Om}_i\right\},\\&=\mathbf{O}^T vec\left\{(\mathbf{z}_i\circ\mathbf{m}_i)^T\mathbf{O}\right\}=\mathbf{O}^T\left(\mathbf{Z}^T\circ\mathbf{M}\right)^T\mathbf{O},\\&=\mathbf{O}^T\left(\mathbf{Z}^T\circ\mathbf{M}\right)\mathbf{O}.\end{aligned} \tag{22}$$

According to Lemma 1, (21 becomes

$$\begin{aligned}&\min_{\boldsymbol{\varphi}^T\mathbf{1}_d=1,\boldsymbol{\varphi}\geqslant 0}\boldsymbol{\varphi}^T\left(\mathbf{XHX}^T\right)\circ\left(\mathbf{WW}^T\right)\boldsymbol{\varphi}\\&\qquad-\boldsymbol{\varphi}^T diag\left(2\mathbf{XHFW}^T\right),\\\Longleftrightarrow&\min_{\boldsymbol{\varphi}^T\mathbf{1}_d=1,\boldsymbol{\varphi}\geqslant 0}\boldsymbol{\varphi}^T\mathbf{G}\boldsymbol{\varphi}-\boldsymbol{\varphi}^T\eta,\end{aligned} \tag{23}$$

where, $\mathbf{G}=\left(\mathbf{XHX}^T\right)\circ\left(\mathbf{WW}^T\right),\eta=diag\left(2\mathbf{XHFW}^T\right)$. Let $\mu=\varphi$, then (23) becomes

$$\min_{\boldsymbol{\varphi}^T\mathbf{1}_d=1,\boldsymbol{\mu}\geqslant 0,\boldsymbol{\mu}=\boldsymbol{\varphi}}\boldsymbol{\varphi}^T\mathbf{G}\boldsymbol{\varphi}-\boldsymbol{\varphi}^T\boldsymbol{\eta}. \tag{24}$$

The augmented Lagrangian function method (ALM) is introduced to solve the constraint minimization problem (24), which decomposes the problem into multiple sub-problems [54]. The augmented Lagrangian equation of (24) is:

$$\begin{aligned}L(\boldsymbol{\varphi},\boldsymbol{\mu},\gamma,\boldsymbol{\beta}_1,\beta_2)&=\boldsymbol{\varphi}^T\mathbf{G}\boldsymbol{\varphi}-\boldsymbol{\varphi}^T\boldsymbol{\eta}+\tfrac{\gamma}{2}||\boldsymbol{\varphi}-\boldsymbol{\mu}+\tfrac{1}{\gamma}\boldsymbol{\beta}_1||_F^2\\&\quad+\tfrac{\gamma}{2}\left(\boldsymbol{\varphi}^T\mathbf{1}_d-1+\tfrac{1}{\gamma}\beta_2\right)^2,\\ s.t.\ \boldsymbol{\mu}&\geqslant 0,\end{aligned} \tag{25}$$

where β_1 indicates the column vector, and γ indicates the Lagrangian multiplier. When μ fixed, (25) is equivalent to

$$\min_{\boldsymbol{\varphi}}\tfrac{1}{2}\boldsymbol{\varphi}^T\boldsymbol{\Omega}\boldsymbol{\varphi}-\boldsymbol{\varphi}^T\boldsymbol{\delta}, \tag{26}$$

where, $\Omega=2\mathbf{G}+\gamma\mathbf{I}_d+\gamma\mathbf{1}_d\mathbf{1}_d^T,\boldsymbol{\delta}=\boldsymbol{\eta}+\gamma\boldsymbol{\mu}+\gamma\mathbf{1}_d-\beta_2\mathbf{1}_d-\boldsymbol{\beta}_1+\boldsymbol{\eta}$. According to (26), $\hat{\boldsymbol{\varphi}}=\Omega^{-1}\boldsymbol{\delta}$. Similarly, when φ fixed, (25) is equivalent to

$$\min_{\boldsymbol{\mu}\geqslant 0}||\boldsymbol{\mu}-\left(\boldsymbol{\varphi}+\tfrac{1}{\gamma}\boldsymbol{\beta}_1\right)||^2. \tag{27}$$

From (27), $\hat{\boldsymbol{\mu}}=\max(\hat{\boldsymbol{\varphi}}+\frac{1}{\gamma}\boldsymbol{\beta}_1,0)$. According to the ALM, $\boldsymbol{\beta}_1=\boldsymbol{\beta}_1+\gamma(\hat{\boldsymbol{\varphi}}-\hat{\boldsymbol{\mu}}),\beta_2=\beta_2+\gamma(\hat{\boldsymbol{\varphi}}^T\mathbf{1}_d-1),\gamma=\rho\gamma$. Algorithm 3 introduces the details of solving problem (23).

Algorithm 3: Algorithm to Solve Problem (23).

Input: Data matrix $\mathbf{X}\in\mathbb{R}^{d\times n}$;

Initialization: Initialize $\boldsymbol{\varphi}_i=1/d,\boldsymbol{\mu}=\boldsymbol{\varphi},\boldsymbol{\beta}_1=(0,0,\cdots,0)^T\in\mathbb{R}^{d\times 1},\beta_2=0,\gamma>0,\rho>1$;

Repeat:

1. Update $\Omega=2\mathbf{G}+\gamma\mathbf{I}_d+\gamma\mathbf{1}_d\mathbf{1}_d^T$, $\boldsymbol{\delta}=\boldsymbol{\eta}+\gamma\boldsymbol{\mu}+\gamma\mathbf{1}_d-\beta_2\mathbf{1}_d-\boldsymbol{\beta}_1+\boldsymbol{\eta}$;
2. Update $\hat{\boldsymbol{\varphi}}=\Omega^{-1}\boldsymbol{\delta}$;
3. Update $\hat{\boldsymbol{\mu}}=\max\left(\hat{\boldsymbol{\varphi}}+\frac{1}{\gamma}\boldsymbol{\beta}_1,0\right)$;
4. Update $\boldsymbol{\beta}_1=\boldsymbol{\beta}_1+\gamma(\hat{\boldsymbol{\varphi}}-\hat{\boldsymbol{\mu}})$;
5. Update $\beta_2=\beta_2+\gamma(\hat{\boldsymbol{\varphi}}^T\mathbf{1}_d-1)$;
6. Update $\gamma=\rho\gamma$;

Until convergence;

Output: $\hat{\boldsymbol{\varphi}}$

In summary, first, the objective function (11) is solved by alternately optimizing variables $\mathbf{W}, \mathbf{F}, \mathbf{Y}$ and $\boldsymbol{\varphi}$, then sort the features in descending order according to the variable $\boldsymbol{\varphi}$, and select the first l features to form a new dataset. So far, feature selection of the original dataset is completed. Algorithm 4 summarizes the process of the proposed FSDSC.

3.5. Computational and space complexity analysis

In the update process in the previous section, n is the number of samples, and d represents the number of features. In addition, let c be the number of categories. The complexity when updating $\mathbf{W}$ is $O(n^2c + nc^2 + n^3)$. When updating $\mathbf{F}$, the complexity is $O(nc + nc^2 + c^2)$. The complexity of updating $\mathbf{Y}$ is $O(n^2)$ and the complexity of $\boldsymbol{\varphi}$ updating is $O\left(d^2\right)$. The total computational complexity is $O\left(n^2c + nc^2 + n^3 + nc + c^2 + n^2 + d^2\right)$ at each iteration. When the number of iterations is N_{Iter}, the total computational complexity is $O\left(N_{Iter}\left(n^2c + nc^2 + n^3 + nc + c^2 + n^2 + d^2\right)\right)$. Since in practical applications, $c \ll n$, and $n \geqslant d$ or $n \leqslant d$, the total computational complexity of FSDSC is $O\left(N_{Iter}\left(n^3 + n^2 + d^2\right)\right)$. When updating the objective function of FSDSC, the required space complexity for some matrices is $O\left(n^2 + d^2 + nd + nc + dc\right)$, and the space complexity for defining variables is $O(n^2 + c^2)$. Similar to the case of computational complexity, the total space complexity is $O\left(n^2 + d^2\right)$.

Algorithm 4: the procedure of FSDSC.

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, parameters α, the maximum number of iterations N_{Iter}, the Gaussian scale parameters σ, the number of selected features l; Construct the similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$, calculate the diagonal matrix $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$ and the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{S}$;

Initialization: Initialize $iter = 0$, $\mathbf{H} = \mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n^T$, $\mathbf{Y} \in \{0,1\}^{n\times c}(\mathbf{Y}\mathbf{1}_c = \mathbf{I}_n)$, $\mathbf{F} = \left(\mathbf{Y}^T\mathbf{D}\mathbf{Y}\right)^{-1/2}$, $\mathbf{W} = rand(d,c)$, $\boldsymbol{\varphi}_i = 1/d (1 \leqslant i \leqslant d)$;

Repeat:

1. Update $\mathbf{W}$ according to the GPI method;
2. Update $\mathbf{F}$, $\mathbf{Y}$ according to Algorithm 2;
3. Update $\boldsymbol{\varphi}$ according to Algorithm 3;

Until convergence;

Output: Calculate the weights of all features of $\mathbf{X}$ according to φ, sort them in descending order, and select the first l features to form a new data $\mathbf{X}_{new}$ matrix; the index of the selected feature, the new dataset $\mathbf{X}_{new}$.

3.6. Convergence analysis

FSDSC solves the objective function (11) by alternating iterative method. Therefore, it is necessary to prove that the objective function (11) is convergent under the update rules of variables $\mathbf{W}, \mathbf{F}, \mathbf{y}$ and $\boldsymbol{\varphi}$. In the above analysis, the GPI and ALM methods are introduced to solve the variables $\mathbf{W}$ and $\boldsymbol{\varphi}$ respectively, and the convergence of these two methods has been proved. Next, the convergence of Algorithm 2 will be analyzed. Define t to represent the tth iteration, according to function (17),

$$Tr\left(\mathbf{F}_{t+1}^T\mathbf{Q}_t\right) \geqslant Tr\left(\mathbf{F}_t^T\mathbf{Q}_t\right). \tag{28}$$

Since $\mathbf{Q} = \mathbf{P}\mathbf{F} + \mathbf{H}\mathbf{X}^T\Phi\mathbf{W}$,

$$\begin{aligned} &Tr\left(\mathbf{F}_{t+1}^T\mathbf{P}\mathbf{F}_t\right) + Tr\left(\mathbf{F}_{t+1}^T\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}\right) \\ &\geqslant Tr\left(\mathbf{F}_t^T\mathbf{P}\mathbf{F}_t\right) + Tr\left(\mathbf{F}_t^T\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}\right), \end{aligned} \tag{29}$$

where, $\mathbf{P}$ is a positive semi-definite matrix, and $\mathbf{P}$ can be expressed as $\mathbf{P} = \Sigma^T\Sigma$ by Cholesky decomposition. Therefore, (29) is rewritten as

$$\begin{aligned} &Tr\left(\mathbf{F}_{t+1}^T\Sigma^T\Sigma\mathbf{F}_t\right) + Tr\left(\mathbf{F}_{t+1}^T\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}\right) \\ &\geqslant Tr\left(\mathbf{F}_t^T\Sigma^T\Sigma\mathbf{F}_t\right) + Tr\left(\mathbf{F}_t^T\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}\right). \end{aligned} \tag{30}$$

Because the inequality $||\Sigma\mathbf{F}_{t+1} - \Sigma\mathbf{F}_t||_F^2 \geqslant 0$ holds, so

$$\begin{aligned} &Tr\left(\mathbf{F}_{t+1}^T\Sigma^T\Sigma\mathbf{F}_{t+1}\right) - 2Tr\left(\mathbf{F}_{t+1}^T\Sigma^T\Sigma\mathbf{F}_t\right) \\ &+Tr\left(\mathbf{F}_t^T\Sigma^T\Sigma\mathbf{F}_t\right) \geqslant 0. \end{aligned} \tag{31}$$

(30) is multiplied by 2 and combined with (31),

$$\begin{aligned} &Tr\left(\mathbf{F}_{t+1}^T\mathbf{P}\mathbf{F}_{t+1}\right) + 2Tr\left(\mathbf{F}_{t+1}^T\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}\right) \\ &\geqslant Tr\left(\mathbf{F}_t^T\mathbf{P}\mathbf{F}_t\right) + 2Tr\left(\mathbf{F}_t^T\mathbf{H}\mathbf{X}^T\Phi\mathbf{W}\right). \end{aligned} \tag{32}$$

It can be seen from (32) that in Algorithm 2, the value of function (16) is monotonous and non-decreasing, that is, the value of objective function (14) is monotonous and non-increasing. Therefore, Algorithm 2 is convergent. Therefore, the objective function (11) of the FSDSC algorithm is convergent under the update rules of variables $\mathbf{W}, \mathbf{F}, \mathbf{Y}$ and $\boldsymbol{\varphi}$.

4. Simulation results and the analyses

In this section, the experiments are conducted to verify the effectiveness of FSDSC. Specifically, first, feature selection is performed on the same datasets through FSDSC and the compared algorithms, the selected l features are recombined into a new dataset, and then k-means method [55] is used to cluster the new dataset. The performance of FSDSC is evaluated by analyzing the clustering effect. In addition, the parameter sensitivity analysis and convergence study are conducted. Before showing the results, the details of experiments are introduced.

4.1. Datasets and the compared algorithms

This experiment uses 11 datasets, including Yale, COIL20, AT&T, Jaffe, Umist, Orl64, Optdigit, Yale64, USPS, Orlraws and PIE32. Table 1 summarizes the specific information of these datasets.

Five unsupervised feature selection algorithms and Baseline are compared to verify the effectiveness of FSDSC. (1) **Baseline**: All features are selected and clustered by k-means method. (2) **MCFS**: A two-step strategy is adopted. First, the low-dimensional subspace index matrix is obtained through spectral embedding learning, and then the regression coefficient matrix is obtained based on the l_1-*norm* constrained sparse regression model, thereby completing feature selection [24]. (3) **JELSR**: Construct a framework for unsupervised feature selection, which combines low-dimensional embedding learning and sparse regression to preserve the local structure of data [25]. (4) **SOGFS**: Feature selection and local structure learning are performed at the same time, so that the algorithm

Table 1
Characteristics of datasets.

Datasets	Instance	Feature	Class
Yale	165	1024	15
Jaffe	213	676	10
Umist	575	644	20
COIL20	1440	1024	20
Orl64	400	4096	40
AT&T	400	10304	40
Optdigit	3823	64	10
Yale64	165	4096	15
USPS	9298	256	10
Orlraws	100	10304	10
PIE32	11554	1024	68

can adaptively determine the similarity matrix. In addition, the constraints imposed on the similarity matrix are used to ensure that SOGFS can select more valuable features [56]. (5) **URAFS**: A generalized uncorrelated regression model (GURM) is proposed to find irrelevant but discriminative features. The graph regularization term based on maximum entropy is incorporated into the GURM model to embed the local geometric structure of data in the manifold learning [26]. (6) **Zhou's**: many basic graphs are constructed, and adaptive consensus graphs are learned through these basic graphs, which are used to characterize the inherent structure of data. In order to promote structural learning and feature selection, this method integrates them into a unified framework [57].

4.2. Evaluation metrics

In order to evaluate the clustering results of all algorithms, two popular metrics are chosen: clustering accuracy (ACC) [27] and normalized mutual information (NMI) [53]. The higher the values of ACC and NMI are, the better the clustering result. Therefore, ACC and NMI can reflect the feature selection effectiveness of all algorithms. ACC is defined as:

$$\mathrm{ACC} = \frac{1}{n}\sum_{i=1}^{n}\delta(c_i, map(g_i)) \tag{33}$$

where c_i and g_i respectively denote the clustering label and the true label of $\mathbf{x}_i$. $map(\cdot)$ is an optimal mapping function, which utilizes Hungarian method [58] to match clustering labels with true labels. $\delta(c_i, g_i)$ is an indicator function, if $c_i = g_i, \delta(c_i, g_i) = 1$, otherwise, $\delta(c_i, g_i) = 0$. NMI is defined as:

$$\mathrm{NMI} = \frac{MI\left(C,\widetilde{C}\right)}{max\left(H(C),H\left(\widetilde{C}\right)\right)} \tag{34}$$

where C and $\widetilde{C}$ respectively represent the clustering labels and the true labels. $H(C)$ is the entropy of C, and $H\left(\widetilde{C}\right)$ is the entropy of $\widetilde{C}$. $MI\left(C,\widetilde{C}\right)$ is the information entropy between C and $\widetilde{C}$.

$$MI\left(C,\widetilde{C}\right) = \sum_{c_i\in C,\widetilde{c_j}\in\widetilde{C}} p(c_i,\widetilde{c_j}) log\frac{p\left(c_i,\widetilde{c_j}\right)}{p(c_i)p\left(\widetilde{c_j}\right)} \tag{35}$$

where $p(c_i)$ and $p(\widetilde{c_j})$ respectively indicate the probabilities that a sample belongs to the clusters c_i and $\widetilde{c_j}$. $p(c_i,\widetilde{c_j})$ is the joint probability that a sample simultaneously belongs to the clusters c_i and $\widetilde{c_j}$. It is worth noting that ACC and NMI are two different evaluation metrics for clustering results. ACC reflects the accuracy of the clustering result, while NMI reflects the consistency between the clustering result and the true label. For clustering results on the same dataset, ACC and NMI may not reach the highest at the same time.

4.3. Experimental settings

The value ranges of some parameters in the experiment are set as follows. In FSDSC, the maximum number of iterations $N_{Iter} = 50$, the Gaussian scale parameter $\sigma = 1e+2$, the nearest neighbor parameter $k = 5$, the parameter $\gamma = 1, \rho = 1.5$ in Algorithm 3, and the value range of the balance parameter α is $\{10^8, 10^7, 10^6, 10^5, 10^4, 10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}\}$. For other the compared methods, the corresponding parameters is adjusted according to the suggestions in the paper. The feature selection parameter l is in the range of {20, 40, 60, 80, 100, 120, 140, 160, 180, 200}. Since the results of the k-means clustering method are dependent on initialization, the clustering experiments are implemented 20 runs independently. And the mean values of ACC and NMI are taken as the final result, respectively.

4.4. Clustering results and analysis

Table 2 shows the average and standard deviation of clustering accuracy (ACC) for feature selection of FSDSC and the compared algorithms on different datasets. Table 3 summarizes the average and standard deviation of the normalized mutual information (NMI) for the same experiment. In both tables, the best value is highlighted in bold and the second best value is underlined.

It can be seen from Table 2 that the ACC of FSDSC on 11 datasets is better than the other compared algorithms. The ACC of FSDSC exceeds Baseline on 10 datasets. It can be seen from Table 3 that in most datasets, the NMI of FSDSC is better than the baseline and the compared algorithms. On Jaffe dataset, the NMI of FSDSC is slightly smaller than Zhou's, but better than the other compared algorithms. The results in Tables 2 and 3 verify the performance of

Table 2
Clustering accuracy of different algorithms on 11 datasets (ACC ± STD%).

Datasets	Baseline	MCFS	JELSR	SOGFS	URAFS	Zhou's	FSDSC
Yale	40.85 ± 2.99	41.78 ± 2.33	42.21 ± 3.16	44.27 ± 3.01	36.36 ± 2.21	47.03 ± 2.31	**49.54** ± 3.20
Jaffe	**88.08** ± 5.77	83.70 ± 3.35	85.14 ± 4.93	83.19 ± 5.39	85.91 ± 4.78	82.13 ± 3.87	87.61 ± 4.95
Umist	43.08 ± 2.24	50.68 ± 3.68	49.50 ± 3.18	46.02 ± 2.50	48.62 ± 3.08	50.36 ± 2.98	**51.78** ± 2.42
COIL20	64.35 ± 3.83	65.15 ± 3.00	65.37 ± 2.27	58.68 ± 2.39	63.45 ± 2.58	65.83 ± 2.26	**67.25** ± 2.76
Orl64	53.53 ± 2.83	53.62 ± 2.77	53.46 ± 2.39	50.50 ± 2.11	47.98 ± 2.05	51.80 ± 2.43	**55.22** ± 2.46
AT&T	60.96 ± 3.30	63.61 ± 2.65	60.30 ± 2.66	57.47 ± 3.30	58.11 ± 2.99	60.05 ± 3.15	**65.02** ± 2.61
Optdigit	80.29 ± 0.56	79.53 ± 0.61	81.53 ± 0.63	82.12 ± 1.03	81.02 ± 0.39	82.24 ± 1.65	**84.13** ± 1.00
Yale64	50.44 ± 4.01	48.76 ± 2.97	50.57 ± 3.48	46.25 ± 2.59	44.90 ± 2.37	53.33 ± 2.68	**55.06** ± 2.87
USPS	66.26 ± 1.93	65.06 ± 4.75	52.31 ± 1.77	66.62 ± 0.29	66.24 ± 1.75	59.66 ± 1.28	**68.61** ± 0.09
Orlraws	73.65 ± 7.06	75.15 ± 3.95	74.55 ± 4.24	76.38 ± 3.68	75.95 ± 4.45	76.95 ± 3.82	**78.15** ± 4.52
PIE32	7.54 ± 0.23	8.08 ± 0.22	7.40 ± 0.22	8.10 ± 0.21	8.08 ± 0.23	8.01 ± 0.17	**8.285** ± 0.22

Table 3
Normalized mutual information of different algorithms on 11 datasets (NMI ± STD%).

Datasets	Baseline	MCFS	JELSR	SOGFS	URAFS	Zhou's	FSDSC
Yale	46.95 ± 2.37	47.79 ± 2.39	48.47 ± 2.00	50.63 ± 2.44	43.12 ± 2.34	53.92 ± 2.68	**56.34** ± 3.17
Jaffe	**88.84** ± 3.62	84.85 ± 3.27	85.14 ± 3.12	85.56 ± 3.41	84.98 ± 3.19	87.91 ± 3.85	86.77 ± 3.31
Umist	64.58 ± 1.51	67.97 ± 1.91	67.78 ± 1.27	64.63 ± 1.49	66.89 ± 1.69	67.99 ± 1.56	**69.14** ± 1.27
COIL20	76.35 ± 1.78	75.02 ± 1.47	75.48 ± 1.39	71.51 ± 1.28	74.25 ± 1.36	75.80 ± 1.85	**77.34** ± 1.22
Orl64	73.33 ± 1.53	73.02 ± 1.16	72.34 ± 1.43	70.12 ± 1.37	68.95 ± 1.30	72.39 ± 1.67	**74.35** ± 1.40
AT&T	79.96 ± 1.37	80.99 ± 1.56	78.69 ± 1.33	77.06 ± 1.54	76.65 ± 1.39	79.06 ± 1.54	**82.43** ± 1.42
Optdigit	75.75 ± 0.33	74.85 ± 0.42	74.66 ± 0.17	74.92 ± 0.41	74.18 ± 0.27	74.25 ± 0.96	**76.12** ± 0.36
Yale64	55.81 ± 3.04	54.68 ± 2.46	54.57 ± 1.96	49.61 ± 1.96	49.80 ± 1.68	58.70 ± 2.45	**61.21** ± 3.04
USPS	61.13 ± 0.86	58.89 ± 2.03	48.70 ± 0.09	60.83 ± 0.21	61.83 ± 0.50	57.40 ± 0.36	**62.35** ± 0.04
Orlraws	79.87 ± 5.31	82.52 ± 3.07	81.45 ± 2.93	80.40 ± 0.42	80.97 ± 3.62	81.99 ± 2.08	**84.44** ± 0.42
PIE32	18.90 ± 0.29	20.67 ± 0.20	20.51 ± 0.24	19.73 ± 0.21	20.42 ± 0.28	21.02 ± 0.24	**21.61** ± 0.17

FSDSC is better. FSDSC obtains a high-quality clustering indicator matrix through discrete spectral clustering, retains the geometric information of data, and more accurately guides feature selection, thereby reducing the dimensionality of data and the amount of calculation for subsequent processing. In order to study the effect of the number of selected features on the proposed algorithm, this experiment shows the clustering performance of FSDSC and the compared algorithms when different numbers of features are selected. Fig. 1 shows ACC of each algorithm on different datasets. Fig. 2 shows the NMI for the same experiment. In Fig. 1 and Fig. 2, the abscissa represents the number of selected features, the ordinate in Fig. 1 represents ACC, and the ordinate in Fig. 2 represents NMI.

In Fig. 1, the Baseline and six feature selection methods are represented by 7 different color curves, where the red curve represents the proposed FSDSC. As can be seen from Fig. 1, on Yale, AT&T, Optdigit, Yale64 and PIE32 datasets, the red curves are always above other curves. This shows that the ACC of FSDSC is better than the compared algorithms on these datasets. On Jaffe, Umist, COIL20, Orl64, and USPS datasets, most points of the red curves are higher than other curves, and the entire curves are above most of other curves. On the dataset Orlraws, when the number of features is 20, 40 and 60, the red curve is under the other curves, but then the red curve is completely above the other curves. Overall, Fig. 1 illustrates that the effect of FSDSC is better than the compared algorithms.

As shown in Fig. 2, on datasets Yale, Umist, AT&T, Optdigit, Yale64 and PIE32, the red curves of FSDSC are above the curves of the compared algorithms. Especially on datasets Yale and Yale64, the red curves have obvious advantages. This shows that on these datasets, the NMI of FSDSC is better than the compared algorithms. On the COIL20 and Orl64 datasets, most points of the red curves are higher than other curves, and the highest point is above other curves. On datasets Jaffe and Orlraws, the red curves

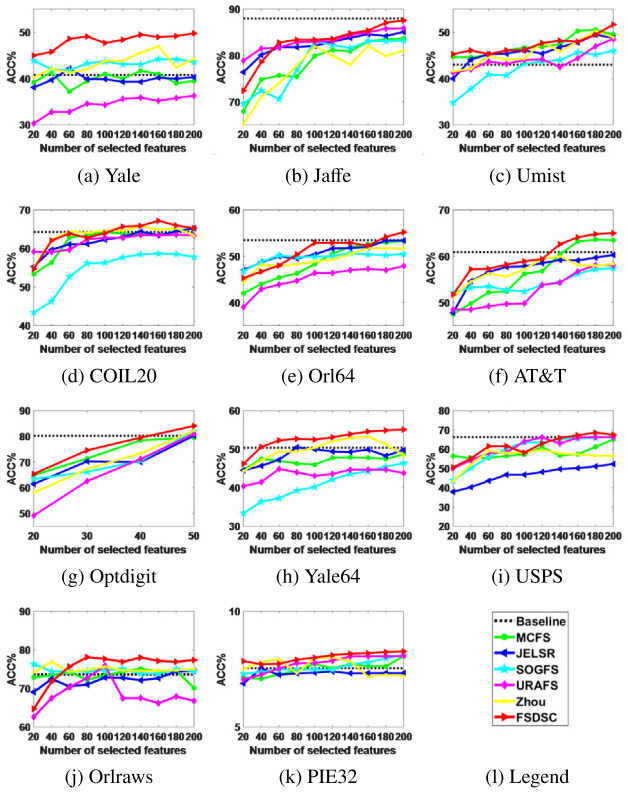


Fig. 1. The ACC of all algorithms for selecting different numbers of features on the 11 datasets.

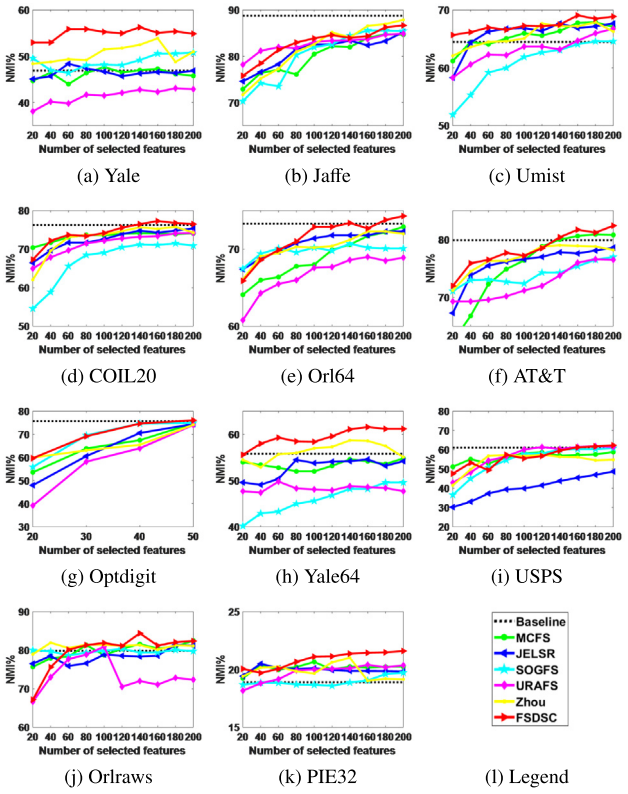


Fig. 2. The NMI of all algorithms for selecting different numbers of features on the 11 datasets.

of FSDSC are above most of other curves. It can be found on the dataset USPS that the low position of the red curve is mostly the highest point of other curves. And the highest point of the red curve is above the highest point of all the curves. Overall, the effect of the clustering experiment of FSDSC is better than the compared algorithms. In summary, compared with other methods, FSDSC performs better in feature selection.

4.5. Wilcoxon rank sum test

In Section 4.4, the clustering results of FSDSC and the compared algorithms are presented. In order to more intuitively illustrate that the clustering effect of FSDSC is significantly improved compared with the compared algorithms, the Wilcoxon rank sum test [59] is performed at a significance level of 0.05. The specific method is to repeat the clustering experiment 20 times for each algorithm, and the p value and h value are obtained. In addition, in this experiment, the *mean* value of 20 times is also displayed in the tables. In the Wilcoxon rank sum test, h has two values. $h = 0$ means $p \geqslant 0.05$, indicating that the null hypothesis cannot be rejected at the 5% significance level. When $h = 1$, it means $p \leqslant 0.05$, which means that the null hypothesis is rejected at the 5% significance level. In short, when $h = 0$, it means that the differences between FSDSC and the compared algorithms are not obvious. When $h = 1$, it means that there are significant differences between FSDSC and the compared algorithms. Tables 4 and 5 show the statistical results of the clustering results of the FSDSC and the compared algorithms.

As can be seen from Table 4, in the rank sum test of FSDSC and the compared algorithms, the h values are almost 1, and the p values are also very small. This shows that the ACC of FSDSC is significantly improved compared with other algorithms. On the dataset Umist, the h value calculated with FSDSC and MCFS is 0. This shows that, compared with MCFS, FSDSC has no obvious improvement. Similarly, on the dataset Orlraws, FSDSC is calculated with SOGFS and Zhou's, and the h value is 0. It shows that on this dataset, FSDSC is not significantly higher than that with SOGFS and Zhou's. Overall, the improvement in clustering accuracy of FSDSC is significant.

Table 5 reflects the improvement of the NMI value of FSDSC and the compared algorithms. It is not difficult to find from Table 5 that, except on individual dataset such as dataset Optdigit, the rank sum test is $h = 1$. This shows that on dataset Optdigit, the differences between FSDSC and JELSR are not obvious. In addition, on the dataset Orlraws, the h value calculated with MCFS and JELSR is 0. This shows that there is no obvious difference between FSDSC and the above two algorithms. However, on the dataset Orlraws, compared with other algorithms, $h = 1$ is in most cases. This shows that FSDSC is still significantly better than the compared algorithms. In most cases, $h = 1$ and the p-value is extremely small. Therefore, the effect of improving the NMI value of FSDSC is also obvious.

4.6. Parameter sensitivity analysis

The parameters of FSDSC include parameter α, feature selection parameter l, and Gaussian scale parameter σ. The sensitivity of parameter α is discussed here. Other parameters are fixed and the changes of ACC and NMI are given when the parameter α changes. The value range of parameter α is $\{10^8, 10^7, 10^6, 10^5, 10^4, 10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}10^{-3}\}$. Fig. 3 shows the changes of ACC and NMI on 11 datasets under different values of α. In Fig. 3, the ordinate represents the clustering performance of FSDSC, and the abscissa represents the value of parameter α.

It can be seen from Fig. 3 that the ACC and NMI of FSDSC on most datasets fluctuate little when parameter α changes. Especially

Table 4
Rank sum test statistics for FSDSC and each the compared algorithm (ACC).

Datasets	FSDSC		MCFS			JELSR			SOGFS			URAFS			Zhou's	
	Mean	Mean	p	h	Mean	p	h	Mean	p	h	Mean	p	h	Mean	p	h
Yale	**49.54**	41.78	1.7e−10	1	48.47	1.6e−11	1	44.27	1.2e−08	1	36.36	6.5e−14	1	47.03	2.5e−05	1
Jaffe	**87.91**	83.70	2.1e−04	1	85.14	0.0078	1	83.19	2.3e−06	1	85.91	0.0278	1	82.13	1.7e−08	1
Umist	**51.78**	50.68	0.0774	0	67.78	4.7e−07	1	46.02	4.4e−13	1	48.62	4.5e−09	1	50.36	6.5e−08	1
COIL20	**67.25**	65.15	0.0169	1	75.48	3.7e−07	1	58.68	2.1e−14	1	63.45	5.1e−09	1	65.83	0.0025	1
Orl64	**55.22**	53.62	7.3e−04	1	72.34	1.2e−04	1	50.50	2.1e−13	1	47.98	2.4e−14	1	51.80	4.6e−10	1
AT&T	**65.02**	63.61	1.4e−14	1	78.69	1.9e−12	1	57.47	1.4e−14	1	58.11	7.2e−13	1	60.05	3.6e−14	1
Optdigit	**84.13**	79.53	3.8e−10	1	74.66	0.0465	1	82.12	2.1e−8	1	81.02	7.2e−10	1	82.24	1.7e−05	1
Yale64	**55.06**	48.76	6.4e−11	1	54.57	1.4e−08	1	46.45	1.5e−14	1	44.90	1.4e−14	1	53.33	6.2e−04	1
USPS	**68.81**	65.06	2.5e−13	1	52.31	1.3e−04	1	66.62	2.5e−13	1	66.24	2.3e−13	1	59.66	1.3e−14	1
Orlraws	**78.15**	75.15	0.0051	1	74.55	0.0035	1	76.38	0.1269	0	75.95	0.0139	1	76.95	0.0158	0
PIE32	**8.28**	8.08	3.0e−05	1	7.40	1.5e−14	1	8.1	0.0177	1	8.08	2.2e−05	1	8.01	1.8e−04	1

Table 5
Rank sum test statistics for FSDSC and each the compared algorithm (NMI).

Datasets	FSDSC	MCFS			JELSR			SOGFS			URAFS			Zhou's		
	Mean	Mean	p	h	Mean	p	h	Mean	p	h	Mean	p	h	Mean	p	h
Yale	**56.34**	47.79	3.5e−13	1	48.47	2.4e−13	1	50.63	4.8e−11	1	43.12	4.7e−14	1	53.92	1.3e−06	1
Jaffe	86.77	84.85	2.7e−08	1	85.14	1.5e−07	1	85.56	3.8e−08	1	84.98	6.1e−07	1	**87.91**	0.1644	0
Umist	**69.14**	67.97	3.5e−07	1	67.78	1.4e−11	1	64.63	1.4e−14	1	66.89	2.2e−11	1	67.99	9.7e−11	1
COIL20	**77.34**	75.02	8.4e−08	1	75.48	1.5e−12	1	71.51	2.1e−14	1	74.25	1.6e−12	1	75.80	5.0e−06	1
Orl64	**74.35**	73.02	2.3e−06	1	72.34	7.2e−09	1	70.12	1.4e−14	1	68.95	1.4e−14	1	72.39	2.9e−12	1
AT&T	**82.43**	80.99	1.4e−14	1	78.69	2.4e−14	1	77.06	1.4e−14	1	76.65	1.1e−13	1	79.06	2.0e−14	1
Optdigit	**76.12**	74.85	1.8e−12	1	74.66	0.0528	0	74.92	1.3e−05	1	74.18	1.3e−10	1	74.25	7.4e−04	1
Yale64	**61.21**	54.68	3.3e−14	1	54.57	2.4e−14	1	49.61	1.7e−14	1	49.80	1.4e−14	1	58.70	2.0e−07	1
USPS	**62.35**	58.89	1.3e−14	1	48.70	1.3e−14	1	60.83	1.3e−14	1	61.83	1.3e−14	1	57.40	1.2e−14	1
Orlraws	**84.44**	82.52	0.5801	0	81.45	0.1113	0	80.40	4.7e−04	1	80.97	0.0164	1	81.99	0.1272	1
PIE32	**21.61**	20.67	1.4e−14	1	20.51	1.4e−14	1	19.73	5.4e−19	1	20.42	1.4e−14	1	21.02	1.1e−13	1

on Umist, COIL20, Orl64, AT&T, USPS and PIE32 datasets, the performances of FSDSC are stable. On Jaffe dataset, with the gradual increase of parameter α, the values of ACC and NMI are also slowly increase, but the magnitude of the change is small. Moreover, it can be found from Fig. 3 that in the parameter range $\{10^{-1}, 10^{-2}, 10^{-3}\}$, the polyline part is already straight. Indicating that within this parameter range, the ACC value and the NMI value do not follow changes with the parameters. On most datasets, the ACC and NMI values in the parameter range $\{10^{-1}, 10^{-2}, 10^{-3}\}$ are slightly lower than those in the other intervals. In general, FSDSC is not sensitive to the parameter α.

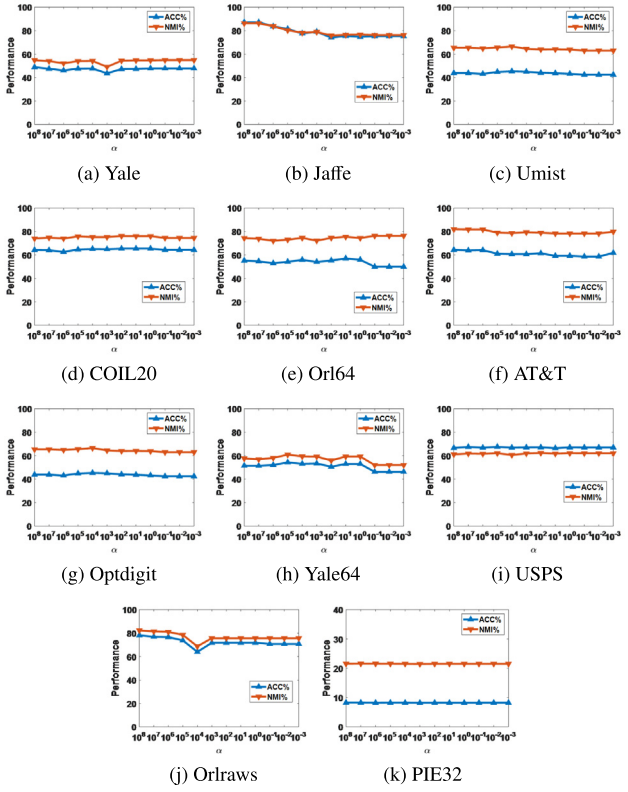


Fig. 3. The ACC and NMI of FSDSC on the 11 datasets under different α.

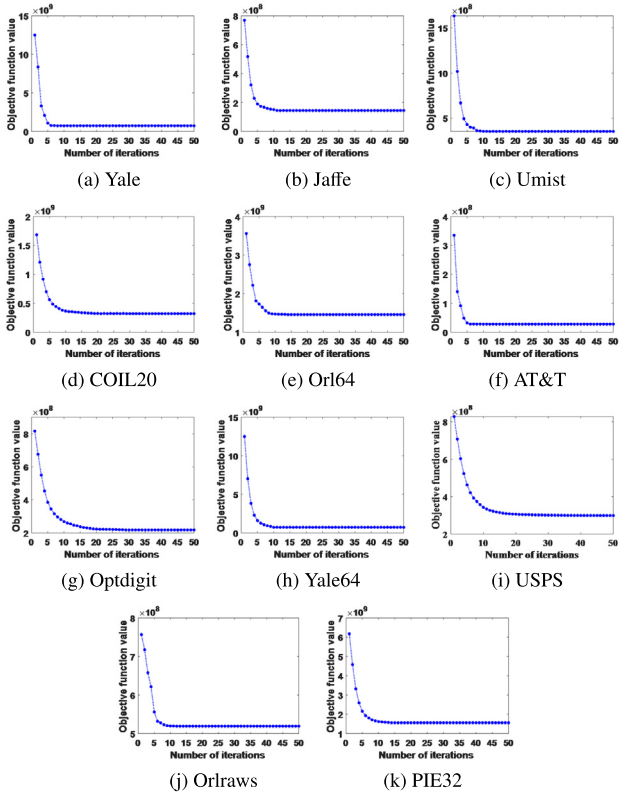


Fig. 4. The convergence curves of FSDSC on 11 datasets.

4.7. Convergence study

The convergence analysis of FSDSC has been given in Section 3.6. Here, the convergence curves of FSDSC on different datasets are shown to visually prove the convergence of the proposed algorithm.

The horizontal axis represents the number of iterations, and the vertical axis represents the value of the objective function. It can be seen from Fig. 4 that on most datasets, as the number of iterations increases, the value of objective function reaches a fixed value and converges around the twentieth generation. On the Optdigit dataset, the convergence speed of FSDSC is slower than other cases. At the twentieth iteration, the objective function begins to converge. Therefore, N_{Iter} = 50 is feasible. Fig. 4 verifies the convergence of FSDSC.

5. Conclusions

This paper proposes unsupervised feature selection via discrete spectral clustering and feature weights (FSDSC), which combines regression models and spectral clustering to form a unified feature selection framework. On this basis, FSDSC introduces a feature weight matrix to express the importance of features, which simplifies the process of feature selection. Second, FSDSC obtains a discrete clustering indicator matrix by imposing discrete constraint on spectral clustering, thereby providing clearer guidance information. In addition, this paper imposes orthogonal constraint on the regression model to avoid the trivial solutions. Moreover, the orthogonal constraint and manifold learning are executed at the same time, which preserve the local geometric structure of data better. In the process of optimization, this paper uses an alternate iteration method to solve each variable in the objective function separately. Finally, on the eleven datasets, the compared experiments with Baseline, MCFS, JELSR, SOGFS, URAFS and Zhou's algorithms have verified the effectiveness of the FSDSC algorithm. It can be seen from the clustering experiment that the FSDSC algorithm has relatively good clustering results on most datasets, and can reach the optimal value. Through the line chart, it can be found that the curves representing FSDSC are basically above the curves of other algorithms. This also reflects the effectiveness of FSDSC. It also verifies that the introduction of discrete clustering index is effective. From the parameter sensitivity experiment, we can find that the clustering results of FSDSC do not change much for the changes of parameters, which verifies the robustness of the algorithm in this paper. It can also be explained that there are fewer redundant solutions in the features selected by FSDSC, and orthogonal constraint also plays a role in avoiding trivial solutions. Through convergence analysis and testing, parameter sensitivity and convergence analysis experiments respectively verify the robustness and convergence of the algorithm. In general, through these experiments, it can be found that FSDSC is superior to other the compared algorithms. In the FSDSC algorithm, we focus on the exploration of data structure information and the better integration of spectral clustering methods in feature selection, ignoring the noise information that originally exists in the data, and these noises may affect the performance of the algorithm. Therefore, in

future research, more attention will be paid to the processing of noise in FSDSC.

CRediT authorship contribution statement

Ronghua Shang: Conceptualization, Methodology, Writing – review & editing. **Jiarui Kong:** Methodology, Data curation, Software. **Lujuan Wang:** Methodology, Data curation, Writing – original draft. **Weitong Zhang:** Software. **Chao Wang:** Conceptualization. **Licheng Jiao:** Conceptualization, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

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 Natural Science Foundation of China under Grant Nos. 62176200 and 61871306, the Natural Science Basic Research Program of Shaanxi under Grant No.2022JC-45 and the Open Research Projects of Zhejiang Lab under Grant 2021KG0AB03, the National Key R&D Program of China and the Guangdong Provincial Key Laboratory under Grant No. 2020B121201001.

References

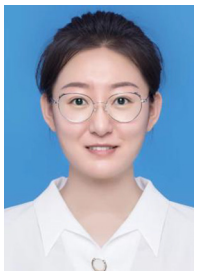
[1] R. Shang, W. Wang, R. Stolkin, L. Jiao, Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection, IEEE Trans. Cybern. 48 (2) (2017) 793–806.
[2] S. Wang, J. Chen, W. Guo, G. Liu, Structured learning for unsupervised feature selection with high-order matrix factorization, Expert Syst. Appl. 140 (2020).
[3] Y. Liu, D. Ye, W. Li, H. Wang, Y. Gao, Robust neighborhood embedding for unsupervised feature selection, Knowl.-based Syst. 193 (2020).
[4] S. Yi, Z. He, X.-Y. Jing, Y. Li, Y.-M. Cheung, F. Nie, Adaptive weighted sparse principal component analysis for robust unsupervised feature selection, IEEE Trans. Neural Networks Learn. Syst. 31 (6) (2019) 2153–2163.
[5] G. Zhao, Y. Wu, An efficient kernel-based feature extraction using a pull–push method, Appl. Soft Comput. 96 (2020).
[6] Y. Zhang, Q. Wang, D.-W. Gong, X.-F. Song, Nonnegative laplacian embedding guided subspace learning for unsupervised feature selection, Pattern Recogn. 93 (2019) 337–352.
[7] P. Li, Q.S. Zhang, G.L. Zhang, W. Liu, F.R. Chen, Adaptive s transform for feature extraction in voltage sags, Appl. Soft Comput. 80 (2019) 438–449.
[8] X. Li, M. Chen, Q. Wang, Self-tuned discrimination-aware method for unsupervised feature selection, IEEE Trans. Neural Networks Learn. Syst. 30 (8) (2018) 2275–2284.
[9] M. Luo, F. Nie, X. Chang, Y. Yang, A.G. Hauptmann, Q. Zheng, Adaptive unsupervised feature selection with structure regularization, IEEE Trans. Neural Networks Learn. Syst. 29 (4) (2017) 944–956.
[10] R. Shang, Y. Meng, W. Wang, F. Shang, L. Jiao, Local discriminative based sparse subspace learning for feature selection, Pattern Recogn. 92 (2019) 219–230.
[11] R. Zhang, X. Li, Unsupervised feature selection via data reconstruction and side information, IEEE Trans. Image Process. 29 (2020) 8097–8106.
[12] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint l2,1-norms minimization, Adv. Neural Inf. Process. Syst. 23 (2010).
[13] B. Krishnapuram, A. Harternink, L. Carin, M.A. Figueiredo, A bayesian approach to joint feature selection and classifier design, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1105–1111.
[14] Z. Zhao, H. Liu, Semi-supervised feature selection via spectral analysis, in: Proceedings of the 2007 SIAM international conference on data mining, SIAM, 2007, pp. 641–646.
[15] Q. Cheng, H. Zhou, J. Cheng, The fisher-markov selector: fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data, IEEE Trans. Pattern Anal. Mach. Intell. 33 (6) (2010) 1217–1233.
[16] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, Neurocomputing 300 (2018) 70–79.
[17] M.H. Law, M.A. Figueiredo, A.K. Jain, Simultaneous feature selection and clustering using mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1154–1166.
[18] C. Tang, M. Bian, X. Liu, M. Li, H. Zhou, P. Wang, H. Yin, Unsupervised feature selection via latent representation learning and manifold regularization, Neural Networks 117 (2019) 163–178.
[19] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1–2) (1997) 273–324.
[20] R. Shang, K. Xu, F. Shang, L. Jiao, Sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection, Knowl.-Based Syst. 187 (2020).
[21] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[22] R. Zhang, F. Nie, X. Li, X. Wei, Feature selection with multi-view data: A survey, Inf. Fusion 50 (2019) 158–167.
[23] J.G. Dy, C.E. Brodley, Feature selection for unsupervised learning, J. Mach. Learn. Res. 5 (Aug) (2004) 845–889.
[24] 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, 2010, pp. 333–342.
[25] C. Hou, F. Nie, X. Li, D. Yi, Y. Wu, Joint embedding learning and sparse regression: A framework for unsupervised feature selection, IEEE Trans. Cybern. 44 (6) (2013) 793–804.
[26] X. Li, H. Zhang, R. Zhang, Y. Liu, F. Nie, Generalized uncorrelated regression with adaptive graph for unsupervised feature selection, IEEE Trans. Neural Networks Learn. Syst. 30 (5) (2018) 1587–1595.
[27] X. Fang, Y. Xu, X. Li, Z. Fan, H. Liu, Y. Chen, Locality and similarity preserving embedding for feature selection, Neurocomputing 128 (2014) 304–315.
[28] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, Advances in neural information processing systems 22 (2009).
[29] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Advances in neural information processing systems 14 (2001).
[30] X. He, P. Niyogi, Locality preserving projections, advances in neural information processing systems16, vancouver, British Columbia, Canada, 2003.
[31] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[32] F. Shang, Y. Liu, F. Wang, Learning spectral embedding for semi-supervised clustering, in: 2011 IEEE 11th International Conference on Data Mining, IEEE, 2011, pp. 597–606.
[33] F. Nie, Z. Zeng, I.W. Tsang, D. Xu, C. Zhang, Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering, IEEE Trans. Neural Networks 22 (11) (2011) 1796–1808.
[34] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, Unsupervised feature selection using nonnegative spectral analysis, in: Proceedings of the AAAI conference on artificial intelligence, vol. 26, 2012, pp. 1026–1032.
[35] S. Wang, J. Tang, H. Liu, Embedded unsupervised feature selection, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, 2015.
[36] X. Zhu, X. Li, S. Zhang, Block-row sparse multiview multilabel learning for image classification, IEEE Trans. Cybern. 46 (2) (2015) 450–461.
[37] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, Y. Fang, Low-rank sparse subspace for spectral clustering, IEEE Trans. Knowl. Data Eng. 31 (8) (2018) 1532–1543.
[38] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, P. Zhu, One-step multi-view spectral clustering, IEEE Trans. Knowl. Data Eng. 31 (10) (2018) 2022–2034.
[39] X. Zhu, X. Li, S. Zhang, C. Ju, X. Wu, Robust joint graph sparse coding for unsupervised spectral feature selection, IEEE Trans. Neural Networks Learn. Syst. 28 (6) (2016) 1263–1275.
[40] X. Zhu, S. Zhang, R. Hu, Y. Zhu, et al., Local and global structure preservation for robust unsupervised spectral feature selection, IEEE Trans. Knowl. Data Eng. 30 (3) (2017) 517–529.
[41] W. Zheng, X. Zhu, G. Wen, Y. Zhu, H. Yu, J. Gan, Unsupervised feature selection by self-paced learning regularization, Pattern Recogn. Lett. 132 (2020) 4–11.
[42] T. Strutz, Data fitting and uncertainty: A practical introduction to weighted least squares and beyond, Springer, 2011.
[43] X. Wu, X. Xu, J. Liu, H. Wang, B. Hu, F. Nie, Supervised feature selection with orthogonal regression and feature weighting, IEEE Trans. Neural Networks Learn. Syst. 32 (5) (2020) 1831–1838.
[44] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinf. Comput. Biol. 3 (02) (2005) 185–205.
[45] X. Xu, X. Wu, Feature selection under orthogonal regression with redundancy minimizing, in: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 3457–3461.
[46] X. Li, H. Zhang, R. Zhang, F. Nie, Discriminative and uncorrelated feature selection with constrained spectral analysis in unsupervised learning, IEEE Trans. Image Process. 29 (2019) 2139–2149.
[47] R. Zhang, X. Li, T. Wu, Y. Zhao, Data clustering via uncorrelated ridge regression, IEEE Trans. Neural Networks Learn. Syst. 32 (1) (2020) 450–456.
[48] F. Shang, L. Jiao, F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, Pattern Recogn. 45 (6) (2012) 2237–2250.
[49] J. Ye, Z. Jin, Dual-graph regularized concept factorization for clustering, Neurocomputing 138 (2014) 120–130.
[50] H. Zhang, R. Zhang, F. Nie, X. Li, An efficient framework for unsupervised feature selection, Neurocomputing 366 (2019) 194–207.
[51] 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, 2004, pp. 202–209.
[52] R. Zhang, F. Nie, X. Li, Feature selection under regularized orthogonal least square regression with optimal scaling, Neurocomputing 273 (2018) 547–553.

[53] F. Nie, R. Zhang, X. Li, A generalized power iteration method for solving quadratic problem on the stiefel manifold, Sci. China Inf. Sci. 60 (11) (2017) 1–10.
[54] M.J. Powell, A method for nonlinear constraints in minimization problems, Optimization (1969) 283–298.
[55] A. Rakhlin, A. Caponnetto, Stability of k)means clustering, Advances in neural information processing systems 19 (2006).
[56] F. Nie, W. Zhu, X. Li, Unsupervised feature selection with structured graph optimization, in: Proceedings of the AAAI conference on artificial intelligence, vol. 30, 2016.
[57] P. Zhou, L. Du, X. Li, Y.-D. Shen, Y. Qian, Unsupervised feature selection with adaptive multiple graph learning, Pattern Recogn. 105 (2020).
[58] C.H. Papadimitriou, K. Steiglitz, Combinatorial optimization: algorithms and complexity, Courier Corporation (1998).
[59] J.D. Gibbons, S. Chakraborti, Nonparametric statistical inference, CRC Press, 2014.

Ronghua Shang (M'09) received the B.S. degree in information and computation science and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University in 2003 and 2008, respectively. She is currently a professor with Xidian University. Her current research interests include machine learning, pattern recognition evolutionary computation, image processing, and data mining.

Jiarui Kong received the B.S. degree in college of computer science & engineering from Northwest Normal University, Lanzhou, China. She is currently working toward the master's degree in school of artificial intelligence from Xidian University, Xi'an,China. Her current research interests include machine learning and data mining.

Lujuan Wang received the B.S. degree in School of Computer Science and Technology from Tianjin Polytechnic University, Tianjin, China. Her current research interests include pattern recognition, machine learning.



Weitong Zhang received the B.E. degree in Electronic and Information Engineering from Changchun University of Science and Technology, Changchun, China, in 2013, the M.S. degree in Electronics and Communication Engineering, and the Ph.D. degree in Electronic science and technology from Xidian University, Xi'an, China, in 2017 and 2021. She is currently a lecturer with Xidian University. Her current research interests include complex networks and machine learning.

Chao Wang received the B.S. degree from Lanzhou University in 2016 and the Ph.D. degree from Zhejiang University in 2021. She is currently an assistant research scientist with the Research Center for Big Data Intelligence, Zhejiang Laboratory. Her research interests include spatial data mining and geographic information science.

Yangyang Li (SM'18) received the B.S. and M.S. degrees in computer science and technology, and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2001, 2004, and 2007, respectively. She is currently a Professor with the School of Artificial Intelligence, Xidian University. Her research interests include quantum-inspired evolutionary computation, artificial immune systems, and deep learning.

Licheng Jiao (SM'89) received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. From 1990 to 1991, he was a postdoctoral Fellow in the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, China. Since 1992, Dr. Jiao has been a Professor in the School of Electronic Engineering at Xidian University. Currently, he is the Director of the Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China at Xidian University, Xi'an, China. Dr. Jiao is a Senior Member of IEEE, member of IEEE Xi'an Section Execution Committee and the Chairman of Awards and Recognition Committee, vice board chairperson of Chinese Association of Artificial Intelligence, councilor of Chinese Institute of Electronics, committee member of Chinese Committee of Neural Networks, and expert of Academic Degrees Committee of the State Council. His research interests include image processing, natural computation, machine learning, and intelligent information processing. He has charged of about 40 important scientific research projects, and published more than 20 monographs and a hundred papers in international journals and conferences.