Non-Negative Spectral Learning and Sparse Regression-Based Dual-Graph Regularized Feature Selection

Ronghua Shang, *Member, IEEE*, Wenbing Wang, Rustam Stolkin, *Member, IEEE*, and Licheng Jiao, *Senior Member, IEEE*

***Abstract*—Feature selection is an important approach for reducing the dimension of high-dimensional data. In recent years, many feature selection algorithms have been proposed, but most of them only exploit information from the data space. They often neglect useful information contained in the feature space, and do not make full use of the characteristics of the data. To overcome this problem, this paper proposes a new unsupervised feature selection algorithm, called non-negative spectral learning and sparse regression-based dual-graph regularized feature selection (NSSRD). NSSRD is based on the feature selection framework of joint embedding learning and sparse regression, but extends this framework by introducing the feature graph. By using low dimensional embedding learning in both data space and feature space, NSSRD simultaneously exploits the geometric information of both spaces. Second, the algorithm uses non-negative constraints to constrain the low-dimensional embedding matrix of both feature space and data space, ensuring that the elements in the matrix are non-negative. Third, NSSRD unifies the embedding matrix of the feature space and the sparse transformation matrix. To ensure the sparsity of the feature array, the sparse transformation matrix is constrained using the $L_{2,1}$-norm. Thus feature selection can obtain accurate discriminative information from these matrices. Finally, NSSRD uses an iterative and alternative updating rule to optimize the objective function, enabling it to select the representative features more quickly and efficiently. This paper explains the objective function, the iterative updating rules and a proof of convergence. Experimental results show that NSSRD is significantly more effective than several other feature selection algorithms from the literature, on a variety of test data.**

***Index Terms*—Dual-graph, feature selection, non-negative spectral learning, sparse regression.**

Manuscript received December 21, 2015; revised November 25, 2016; accepted January 17, 2017. Date of publication March 6, 2017; date of current version January 15, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61371201, and in part by the National Basic Research Program (973 Program) of China under Grant 2013CB329402. This paper was recommended by Associate Editor J. Basak.

R. Shang, W. Wang, and L. Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education, Xidian University, Xi'an 710071, China (e-mail: rhshang@mail.xidian.edu.cn; lwwb19910204@163.com).

R. Stolkin is with the Extreme Robotics Laboratory, University of Birmingham, Birmingham B15 2TT, U.K.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2657007

I. Introduction

DEALING with high-dimensional data is a difficult problem in data mining, pattern recognition, machine learning, and other fields [1]. Often, only a small subset of the features are important or useful in dealing with these data [2], while the vast majority of features are often redundant or artifacts of noise [3] which can interfere with processing of the data. Therefore, it is often necessary to reduce the dimension of high-dimensional data. Feature selection and feature extraction are two main dimension reduction methods [2]–[6]. Feature selection chooses a subset of original features that are representative of the original data. In contrast, feature extraction transforms the original data from a high-dimensional space to a low-dimensional space, by merging the original features into some new types of features to represent the original data. Compared to feature extraction, feature selection preserves the physical meaning of the original data, which is often more convenient during subsequent data analysis [2].

According to the extent to which data label information is utilized, feature selection methods can be broadly categorized into supervised [3], [7], semisupervised [8], [9], and unsupervised [10]–[12]. Supervised feature selection exploits known data labels to obtain discriminant information, and then examines the correlation between the features of each data class, so as to determine the importance of each feature. However, obtaining such label information requires more resources (e.g., human annotation) and class labels may not be available in many problems. Semisupervised feature selection can improve the accuracy of the selection by using only a few data labels. Unsupervised feature selection is performed in the absence of any label information, and determines the importance of each feature only by using the intrinsic information of the dataset.

In many practical applications, the label information of the data is unknown, which makes it important to develop unsupervised feature selection methods. This paper focuses on the problem of unsupervised feature selection. According to the search strategies used, unsupervised feature selection includes three main categories: 1) filter [10], [13], [14]; 2) wrapper [15], [16]; and 3) embedded [17], [18] methods.

In recent years, a variety of new algorithms have been proposed to overcome the shortcomings of conventional feature selection algorithms, that typically ignore the information of the data manifold structure and lack learning mechanisms.

He *et al.* [13] proposed Laplacian score (LapScor) and Cai *et al.* [19] proposed unsupervised feature selection for multicluster data (MCFS). Zhao and Liu [10] and Zhao *et al.* [20] proposed spectral feature selection (SPEC) and minimum redundancy spectral feature selection (MRSF). Although LapScor exploits data manifold information, it lacks a learning mechanism, while SPEC suffers similar drawbacks. In contrast, MCFS and MRSF employ a good learning mechanism, which uses manifold regression and the geometric structure information of the data space. However, MCFS and MRSF both adopt a two-step strategy, which can have difficulties in maximally optimizing the objective function. For this problem, Hou *et al.* [11] proposed joint embedding learning and sparse regression (JELSR). JELSR adopts a single step strategy and integrates manifold learning and sparse regression. It simultaneously optimizes the embedding matrix and sparse transformation matrix, and it demonstrates better performance than previous methods. Yang *et al.* [21] proposed unsupervised maximum margin feature selection via $L_{2,1}$-norm minimization (UMMFSSC). UMMFSSC integrates k-means clustering and feature selection into a coherent framework to select the most discriminative subspace. Nie *et al.* [22] proposed improved minmax cut graph clustering with non-negative relaxation (MinMax Cut). Non-negative MinMax Cut relaxes the constraints of the embedding matrix to non-negative and orthogonal constraints, which makes the embedding matrix more appropriate for the ideal label matrix. Li *et al.* [12] proposed clustering-guided sparse structural learning (CGSSL) for unsupervised feature selection. CGSSL also uses non-negative and orthogonal constraints to constrain the embedding matrix. Therefore, the embedding matrix can be regarded as a scaled cluster indicator matrix, which can provide accurate discriminant information for feature selection. Wang *et al.* [23] proposed unsupervised feature selection via unified trace ratio formulation and k-means clustering (TRACK). TRACK uses an unsupervised trace ratio formulation, which can harness the discriminant power of trace ratio latent dirichlet allocation [24] and select discriminative features. Nie *et al.* [25] proposed unsupervised feature selection with structured graph optimization, which performs local structure learning and feature selection simultaneously. The adaptive learning similarity matrix can provide accurate information for feature selection.

However, the above algorithms only use the geometric and discrimination information of the data space, while ignoring the manifold information of the feature space. Therefore, some potentially useful information is not fully exploited.

In clustering, Cai *et al.* [26] proposed locally consistent concept factorization (LCCF), which uses the manifold information of the data space, and demonstrates better performance than concept factorization [27]. Based on LCCF, Ye and Jin [28] proposed dual-graph regularized concept factorization clustering (GCF). GCF uses the geometric information of both data and feature spaces simultaneously, giving significantly improved performance compared to previous methods. In recent years, a variety of new matrix factorization algorithms have been proposed [29]–[33]. Wang and Gao [34] proposed max–min distance non-negative matrix factorization (NMF), which improves the discriminative ability of NMF by maximizing the distance of between-class pairs while minimizing the distance of within-class pairs. By using the manifold structure information of the data, Cai *et al.* [35] proposed graph regularized NMF (GNMF). Wang *et al.* [36] proposed adaptive GNMF via feature selection (AdapGrNMF$_{\mathrm{FS}}$), and later proposed feature selection and multikernel learning for adaptive GNMF (AGNMF$_{\mathrm{FS}}$) and (AGNMF$_{\mathrm{MK}}$) [37]. These methods construct an adaptive graph by using the results of feature selection or multikernel learning, which avoid the effects of noise or redundant features caused by using a fixed graph. Wang *et al.* [38] proposed multiple GNMF, which uses a linear combination of several graphs with different models and parameters to approximate the intrinsic manifold. This strategy overcomes the difficulties of model selection and parameter adjustment. However, the above NMF-based methods can yield a trivial solution, in which the regular terms go to zero. To guarantee the uniqueness of the solution, Huang *et al.* [39] introduced an orthogonality constraint in the objective function and proposed robust manifold NMF (RMNMF). Based on GNMF, Shang *et al.* [40] introduced the feature graph, and proposed graph dual regularization NMF for co-clustering algorithm (DNMF), which demonstrated better performance than GNMF. The enhanced performance suggests that algorithms which simultaneously use the geometry information of both data and feature spaces perform better than those which only exploit the geometry information of the data space.

Building on the advantages and ideas of the above algorithms, this paper proposes non-negative spectral learning and sparse regression-based dual-graph regularized feature selection (NSSRD). The proposed algorithm uses a JELSR feature selection framework [11]. Similar to CGSSL [12], NSSRD also uses non-negative and orthogonality constraints to constrain the embedding matrix of the data space, which makes the embedding matrix more appropriate for the ideal label matrix, providing accurate discrimination information for feature selection. In contrast to JELSR and CGSSL, NSSRD introduces the feature graph, and uses the geometry information of both data space and feature space simultaneously. In order to use the geometry information of the feature space, NSSRD unifies the embedding matrix of the feature space and the sparse transformation matrix. Via an iterative process, this guides the learning of the sparse transformation matrix. To ensure the non-negative and sparsity of the feature array, NSSRD also adopts non-negative constraint and the $L_{2,1}$-norm to constrain the sparse transformation matrix. In addition, the algorithm uses a single step strategy and combines these terms into an objective function for minimization. We abandon the optimization method of JELSR. Instead, we use the alternating iterative update rule to solve this minimization problem.

Our key contributions are highlighted as follows.

1) In the process of learning, the low-dimensional spectral embedding matrix of data space is constrained by non-negative and orthogonal constraints, which makes it much closer to the ideal label matrix, and provides more accurate information for feature selection.

2) On the basis of the existing data graph, the feature graph is introduced, and the feature space is embedded in a low-dimension space. In the sparse regression stage, the sparse transformation matrix is directly related to the low-dimensional spectral embedding matrix of the feature space. Therefore, in the process of feature selection, our method makes full use of the geometry information of both data space and feature space to guide feature selection.
3) In the proposed algorithm, minimization of the objective function is achieved by using the alternating iterative updating rule, which greatly improves computation speed.

The remainder of this paper is organized as follows. In Section II, we introduce the related work. Our algorithm framework, optimization scheme, and convergence analysis are presented in Section III. In Section IV, we present the results and analysis of experiments comparing NSSRD against five other state-of-the-art algorithms on benchmark datasets. Section V provides concluding remarks.

II. Related Works

In this section, we first summarize some notation and theory related to the proposed algorithm. Next, we provide a brief introduction to several unsupervised feature selection algorithms: LapScor, SPEC, MCFS, MRSF, JELSR, and CGSSL. Finally, we also introduce some NMF algorithms, including NMF, DNMF, GNMF, RMNMF, AGNMF_{FS}, and DNMTF. This related material will facilitate understanding of our proposed algorithm, presented later in this paper.

A. Related Notations

Denote $X = \{\boldsymbol{x}_i \in \Re^d | i = 1, 2, \ldots, n\}$ as the original data, in which $\boldsymbol{x}_i \in \Re^d$ is the ith sample, where d is the dimensionality of original data, and n is the number of samples. We use l to represent the number of the selected features, where $l \leq d$. For an arbitrary matrix $\boldsymbol{A} \in \Re^{e \times f}$, the $L_{r,s}$-norm is defined as

$$\|\boldsymbol{A}\|_{r,s} = \left(\sum_{i=1}^{e} \left(\sum_{j=1}^{f} |A_{ij}|^r \right)^{s/r} \right)^{1/s}. \tag{1}$$

When $r = s = 2$, this becomes the *Frobenius-norm* or L_2-norm. We denote it as $\|\cdot\|_2^2$ in the remainder of this paper.

B. Unsupervised Feature Selection

1) LapScor [13] and SPEC [10]: First, we introduce the LapScor [13] algorithm. LapScor is a classical unsupervised feature selection algorithm, which is based on Eigengraphs [41] and preserving projection [42]. The main purpose of LapScor is to compute the weight of each feature by using the local geometric information of the data space. The higher the weight of the feature, the more important it is. Given sample data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$. Let $\boldsymbol{f}_j = [\boldsymbol{x}_{j1}, \boldsymbol{x}_{j2}, \ldots, \boldsymbol{x}_{jn}]$ represents the jth feature, where $j = 1, 2, \ldots, d$. The main steps of the algorithm are as follows.

Step 1: Construct a k-nearest neighborhood graph G in the data space, where k is the number of nearest neighbors. And calculate the similarity matrix $\boldsymbol{W} \in \Re^{n \times n}$.

Step 2: Define diagonal matrix $\boldsymbol{D}$ as $D_{ii} = \sum_{j=1}^{n} W_{ij}$. Define $\mathbf{1} = [1, 1, \ldots, 1]$ as a vector of all ones. Denote $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, which is known as the graph Laplacian [40]. We use $\hat{\boldsymbol{f}}_j$ to represent the coefficient of the jth feature, which is defined as

$$\hat{\boldsymbol{f}}_j = \boldsymbol{f}_j - \frac{\boldsymbol{f}_j^T \boldsymbol{D} 1}{\mathbf{1}^T \boldsymbol{D} 1} \mathbf{1}. \tag{2}$$

Step 3: The Laplacian score of the jth feature is computed as follows:

$$b_j = \frac{\hat{\boldsymbol{f}}_j^T \boldsymbol{L} \hat{\boldsymbol{f}}_j}{\hat{\boldsymbol{f}}_j^T \boldsymbol{D} \hat{\boldsymbol{f}}_j}. \tag{3}$$

Step 4: According to the value of b_j, select the corresponding l maximum features. This completes the LapScor feature selection process.

The SPEC [10] algorithm is similar to LapScor, and it also needs to construct the nearest neighbor graph and compute the similarity matrix. The difference is that they use different evaluation methods to calculate the feature weight, so SPEC can be considered as an extension of LapScor.

Both SPEC and LapScor use the manifold information of the data space, but these two algorithms lack any learning mechanism, which adversely affects the accuracy with which they can choose the most important features.

2) MCFS [19] and MRSF [20]: Both MCFS [19] and MRSF [20] are the classical feature selection algorithms which incorporate a learning mechanism. These two algorithms use a two-step strategy. At the first step, the Laplacian Eigengraphs [41] method is used to embed the data $\boldsymbol{X} \in \Re^{d \times n}$ into a low-dimensional space, and generate $\boldsymbol{S} \in \Re^{m \times n}$, where m is the dimension of the embedding space, and $m < d$. At the second step, the sparse regression method is used to calculate the regression coefficient, which is used to obtain the weight of each feature. MCFS uses the L_1-norm to constrain the sparse transformation matrix, and its objective function is as follows:

$$\begin{aligned} &\arg \min_{\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{I}_m} \operatorname{Tr}\left(\boldsymbol{S}\boldsymbol{L}\boldsymbol{S}^T\right) \\ &\arg \min_{\boldsymbol{P}} \left\|\boldsymbol{P}^T\boldsymbol{X} - \boldsymbol{S}\right\|_2^2 + \alpha\|\boldsymbol{P}\|_1 \end{aligned} \tag{4}$$

where $\operatorname{Tr}(\cdot)$ is used to denote the trace of a matrix.

For MRSF, the objective function is similar to that of MCFS. The difference is that it uses the $L_{2,1}$-norm to constrain the sparse transformation matrix. Therefore, MRSF solves the following problem:

$$\begin{aligned} &\arg \min_{\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{I}_m} \operatorname{Tr}\left(\boldsymbol{S}\boldsymbol{L}\boldsymbol{S}^T\right) \\ &\arg \min_{\boldsymbol{P}} \left\|\boldsymbol{P}^T\boldsymbol{X} - \boldsymbol{S}\right\|_2^2 + \alpha\|\boldsymbol{P}\|_{2,1}. \end{aligned} \tag{5}$$

A disadvantage of these methods is that the low-dimensional embedding learning and sparse regression steps happen separately, and so are unable to properly interact to achieve an improved feature selection learning effect.

3) JELSR and CGSSL: Both JELSR [11] and CGSSL [12] adopt a single step strategy, which avoids the disadvantages of the two-step strategy of MRSF and MCFS, by combining the objective functions of low-dimensional embedding learning and sparse regression. However, the objective function of JELSR is different to that of CGSSL, and the constraints are not the same. The objective function of JELSR is as follows:

$$\arg \min_{\boldsymbol{P},\boldsymbol{S}\boldsymbol{S}^T=\boldsymbol{I}_m} \operatorname{Tr}\left(\boldsymbol{S}\boldsymbol{L}\boldsymbol{S}^T\right)+\beta\left(\left\|\boldsymbol{P}^T\boldsymbol{X}-\boldsymbol{S}\right\|_2^2+\alpha\|\boldsymbol{P}\|_{2,1}\right). \tag{6}$$

In contrast, the objective function of CGSSL is defined as

$$\begin{aligned}\arg \min_{\boldsymbol{P},\boldsymbol{S},\boldsymbol{Q},\boldsymbol{Z}} & \operatorname{Tr}\left(\boldsymbol{S}\boldsymbol{L}\boldsymbol{S}^T\right)+\alpha\left\|\boldsymbol{P}^T\boldsymbol{X}-\boldsymbol{S}\right\|_2^2+\beta\|\boldsymbol{P}\|_{2,1} \\ & +\lambda\|\boldsymbol{Q}\boldsymbol{Z}-\boldsymbol{P}\|_2^2 \\ \text{s.t.} & \ \boldsymbol{S}\boldsymbol{S}^T=\boldsymbol{I}_m, \boldsymbol{S}\geq 0, \boldsymbol{Q}^T\boldsymbol{Q}=\boldsymbol{I}_r\end{aligned} \tag{7}$$

where $\boldsymbol{Z}\in\Re^{r\times m}$ is weight vector matrix and $\boldsymbol{Q}\in\Re^{d\times r}$ is a linear transformation matrix of low-dimensional subspace. JELSR only uses the orthogonal constraint to constrain the low-dimensional embedding matrix, while CGSSL uses both orthogonal and non-negative constraints. CGSSL uses both original feature and feature in a low-dimensional subspace to predict the clustering indicator matrix, which is used to guide the feature selection. Although CGSSL and JELSR use a joint embedding learning and sparse regression approach, these two algorithms only make use of the data space, and they neglect information from the feature space manifold.

C. Non-Negative Matrix Factorization

1) NMF: NMF is a matrix factorization algorithm [43], [44]. Its main purpose is to factorize the original data matrix into two non-negative matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. Given a data matrix $\boldsymbol{X}\in\Re^{d\times n}$, it can be factorized into $\boldsymbol{U}\in\Re^{d\times k}$ and $\boldsymbol{V}\in\Re^{n\times k}$, where $\boldsymbol{U}$ is the dictionary matrix, $\boldsymbol{V}^T$ is the representation matrix, and $k\leq d, k\leq n, \boldsymbol{X}\approx\boldsymbol{U}\boldsymbol{V}^T$. The objective function of NMF is as follows:

$$\begin{aligned}\arg \min_{\boldsymbol{U},\boldsymbol{V}} & \left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\right\|_2^2 \\ \text{s.t.} & \ \boldsymbol{U}\geq 0, \boldsymbol{V}\geq 0.\end{aligned} \tag{8}$$

2) GNMF, RMNMF, and AGNMF$_{FS}$: GNMF [35], RMNMF [39], and AGNMF$_{\text{FS}}$ [37] are extensions of the original NMF algorithm. According to the Laplacian Eigengraphs [41] method, GNMF, RMNMF, and AGNMF$_{\text{FS}}$ construct the data graph and use the geometry information of the data space. GNMF solves the following problem:

$$\begin{aligned}\arg \min_{\boldsymbol{U},\boldsymbol{V}} & \left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\right\|_2^2+\alpha\operatorname{Tr}\left(\boldsymbol{V}^T\boldsymbol{L}\boldsymbol{V}\right) \\ \text{s.t.} & \ \boldsymbol{U}\geq 0, \boldsymbol{V}\geq 0.\end{aligned} \tag{9}$$

The objective function of RMNMF is defined as

$$\begin{aligned}\arg \min_{\boldsymbol{U},\boldsymbol{V}} & \left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\right\|_{21}+\alpha\operatorname{Tr}\left(\boldsymbol{V}^T\boldsymbol{L}\boldsymbol{V}\right) \\ \text{s.t.} & \ \boldsymbol{V}\geq 0, \boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}.\end{aligned} \tag{10}$$

AGNMF$_{\text{FS}}$ solves the following problem:

$$\begin{aligned}\arg \min_{\boldsymbol{U},\boldsymbol{V}} & \left\|\operatorname{diag}(\mu)\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\right)\right\|_2^2+\alpha\operatorname{Tr}\left(\boldsymbol{V}^T\boldsymbol{L}^{\mu}\boldsymbol{V}\right) \\ \text{s.t.} & \ \boldsymbol{U}\geq 0, \boldsymbol{V}\geq 0, \sum_{i=1}^{d}\mu_i=1, \mu_i\geq 0\end{aligned} \tag{11}$$

where μ is the feature weight vector, and diag$(\cdot)$ is used to denote a diagonal matrix.

3) DNMF and DNMTF: DNMF [40] is an extension of GNMF [35], which uses the idea of the dual-graph, and introduces the feature graph. It makes full use of the geometric information of both data space and feature space to improve the effect of the matrix factorization. The objective function of DNMF is denoted as

$$\begin{aligned}\arg \min_{\boldsymbol{U},\boldsymbol{V}} & \left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\right\|_2^2+\alpha\operatorname{Tr}\left(\boldsymbol{V}^T\boldsymbol{L}_V\boldsymbol{V}\right)+\beta\operatorname{Tr}\left(\boldsymbol{U}^T\boldsymbol{L}_U\boldsymbol{U}\right) \\ \text{s.t.} & \ \boldsymbol{U}\geq 0, \boldsymbol{V}\geq 0\end{aligned} \tag{12}$$

where α and β are two balance parameters. At the same time, the idea of the dual-graph was also used for NMTF, giving the DNMTF [40] algorithm. The objective function of DNMTF is as follows:

$$\begin{aligned}\arg \min_{\boldsymbol{U},\boldsymbol{V}} & \left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\right\|_2^2+\alpha\operatorname{Tr}\left(\boldsymbol{V}^T\boldsymbol{L}_V\boldsymbol{V}\right)+\beta\operatorname{Tr}\left(\boldsymbol{U}^T\boldsymbol{L}_U\boldsymbol{U}\right) \\ \text{s.t.} & \ \boldsymbol{U}\geq 0, \boldsymbol{S}\geq 0, \boldsymbol{V}\geq 0.\end{aligned} \tag{13}$$

III. Algorithm Description

In this section, we introduce our proposed NSSRD algorithm in detail. The framework of NSSRD comprises three main parts: 1) dual-graph non-negative spectral learning; 2) dual-graph sparse regression; and 3) feature selection.

A. Dual-Graph Non-Negative Spectral Learning

Spectral theory has been successfully applied in a number of fields [45]–[49]. Among these, the spectral clustering method uses graph theory to describe the potential data manifold structure, to achieve effective clustering. Using spectral graph theory, high-dimensional data can be embedded into a low-dimensional space, which effectively eliminates redundant features or noise, and facilitates the subsequent analysis. Therefore, this advantage can also be applied to feature selection. In recent years, several researchers have suggested that the manifold information of data is not only distributed in the data space but also in the feature space [40], [50], [51]. According to the method of [40], we construct nearest neighbor graphs in both data space and feature space. We first construct a k-nearest neighbor graph $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ in data space, where $\boldsymbol{V}$ denotes the vertex set $\{\boldsymbol{X}_{:,1},\ldots,\boldsymbol{X}_{:,n}\}$, $\boldsymbol{E}$ denotes the weight of the edge between two points, which represents the similarity of two points. We choose Gaussian function [40] and a parameter free method [52] as weight measures, respectively.

The Gaussian function is defined as follows:

$$\left[\boldsymbol{W}^S\right]_{ij}=\begin{cases}\exp\left(-\left\|\boldsymbol{X}_{:,i}-\boldsymbol{X}_{:,j}\right\|_2^2/\sigma^2\right), & \text{if } \boldsymbol{X}_{:,i}\in N\left(\boldsymbol{X}_{:,j}\right) \\ & \text{or } \boldsymbol{X}_{:,j}\in N\left(\boldsymbol{X}_{:,i}\right) \\ 0, & \text{otherwise}\end{cases} \tag{14}$$

where, $i, j = 1, \ldots, n$. $\boldsymbol{X}_{:,i}$ denotes the ith column of the data matrix, which represents the ith data points. $N(\boldsymbol{X}_{:,i})$ denotes the k-nearest neighborhood set of $\boldsymbol{X}_{:,i}$, and σ is the bandwidth parameter of the Gaussian function.

The parameter free method [52] is defined as follows:

$$\left[\boldsymbol{W}^S\right]_{ij} = \begin{cases} \frac{e_{i,k+1} - e_{i,j}}{k e_{i,k+1} - \sum_{h=1}^{k} e_{i,h}}, & \text{if } \boldsymbol{X}_{:,i} \in N\left(\boldsymbol{X}_{:,j}\right) \\ & \text{or } \boldsymbol{X}_{:,j} \in N\left(\boldsymbol{X}_{:,i}\right) \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where k is the number of neighbors, $e_{i,j} = \|\boldsymbol{X}_{:,i} - \boldsymbol{X}_{:,j}\|_2^2$.

The graph Laplacian matrix of the data graph is $\boldsymbol{L}^S = \boldsymbol{D}^S - \boldsymbol{W}^S$, where $\boldsymbol{D}^S$ is a diagonal matrix, and $[\boldsymbol{D}^S]_{ii} = \sum_j [\boldsymbol{W}^S]_{ij}$.

Similarly, we construct a k-nearest neighbor graph in feature space. The vertex set of the graph is a feature set $\{\boldsymbol{X}_{1,:}^T, \ldots, \boldsymbol{X}_{d,:}^T\}$.

The Gaussian function is defined as follows:

$$\left[\boldsymbol{W}^P\right]_{ij} = \begin{cases} \exp\left(-\left\|\boldsymbol{X}_{i,:}^T - \boldsymbol{X}_{j,:}^T\right\|_2^2 \Big/ \sigma^2\right), & \text{if } \boldsymbol{X}_{i,:}^T \in N\left(\boldsymbol{X}_{j,:}^T\right) \\ & \text{or } \boldsymbol{X}_{j,:}^T \in N\left(\boldsymbol{X}_{i,:}^T\right) \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The parameter free method [52] is defined as follows:

$$\left[\boldsymbol{W}^P\right]_{ij} = \begin{cases} \frac{e_{i,k+1} - e_{i,j}}{k e_{i,k+1} - \sum_{h=1}^{k} e_{i,h}}, & \text{if } \boldsymbol{X}_{i,:}^T \in N\left(\boldsymbol{X}_{j,:}^T\right) \\ & \text{or } \boldsymbol{X}_{j,:}^T \in N\left(\boldsymbol{X}_{i,:}^T\right) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where $e_{i,j} = \|\boldsymbol{X}_{i,:}^T - \boldsymbol{X}_{j,:}^T\|_2^2$, $i, j = 1, \ldots, d$. $\boldsymbol{X}_{i,:}$ denotes the ith row of the data matrix, which represents the ith feature. The graph Laplacian matrix of the feature graph is $\boldsymbol{L}^P = \boldsymbol{D}^P - \boldsymbol{W}^P$, where $\boldsymbol{D}^P$ is a diagonal matrix, and $[\boldsymbol{D}^P]_{ii} = \sum_j [\boldsymbol{W}^P]_{ij}$.

Through (14) and (16), or (15) and (17), we obtain the similarity matrix and Laplacian matrix of the data space and the feature space. Next, we use these matrices to carry out dual-graph non-negative spectral learning, which means that we need to embed the data from the high-dimensional data and feature spaces into low-dimensional spaces. More specifically, we transform the original data $\boldsymbol{X}_{:,i} \in \Re^d$ and $\boldsymbol{X}_{i,:}^T \in \Re^n$ into $\boldsymbol{S}_{:,i} \in \Re^m$ and $\boldsymbol{P}_{:,i}^T \in \Re^m$, where m is the dimension of the embedding space, $m < n$ and $m < d$. $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n] \in \Re^{m \times n}$ and $\boldsymbol{P} = [\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \ldots, \boldsymbol{p}_d^T]^T \in \Re^{d \times m}$ are the low-dimensional spectral embedding matrices of the data space and feature space. Considering that we need to make the low-dimensional spectral embedding matrix of the data space close to the ideal label matrix, we make m equal to the real sample class number c. To generate the low-dimensional spectral embedding matrix $\boldsymbol{S}$, we need to solve the following problem:

$$\arg \min_{\boldsymbol{S}} \frac{1}{2} \sum_{i,j=1}^{n} \left\|\boldsymbol{s}_i - \boldsymbol{s}_j\right\|_2^2 \boldsymbol{W}^S = \mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{L}^S\boldsymbol{S}^T\right). \tag{18}$$

The matrix $\boldsymbol{S}$ obtained by the above objective function may contain negative elements, and each column may contain more than one nonzero element, which makes $\boldsymbol{S}$ deviate from the ideal label matrix. Therefore we use the non-negative and orthogonal constraints to constrain $\boldsymbol{S}$, i.e., $\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{I}_m, \boldsymbol{S} \geq 0$, where $\boldsymbol{I}_m \in \Re^{m \times m}$ is the identity matrix. This allows us to obtain a new objective function as follows:

$$\begin{aligned} \arg \min_{\boldsymbol{S}} \ & \frac{1}{2} \sum_{i,j=1}^{n} \left\|\boldsymbol{s}_i - \boldsymbol{s}_j\right\|_2^2 \boldsymbol{W}^S = \mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{L}^S\boldsymbol{S}^T\right) \\ \text{s.t.} \ & \boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{I}_m, \boldsymbol{S} \geq 0. \end{aligned} \tag{19}$$

The method for generating $\boldsymbol{P}$ is similar to that for $\boldsymbol{S}$. In practical problems, the data are usually non-negative. Therefore, we add a non-negative constraint to $\boldsymbol{P}$ to guarantee the features of $\boldsymbol{S}$ can be described as a positive linear combination of the original features. Therefore, we need to solve the following problem:

$$\begin{aligned} \arg \min_{\boldsymbol{P}} \ & \frac{1}{2} \sum_{i,j=1}^{d} \left\|\boldsymbol{p}_i^T - \boldsymbol{p}_j^T\right\|_2^2 \boldsymbol{W}^P = \mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}\right) \\ \text{s.t.} \ & \boldsymbol{P} \geq 0. \end{aligned} \tag{20}$$

B. Dual-Graph Sparse Regression

Next, we introduce dual-graph sparse regression. We need to regress each original sample $\boldsymbol{x}_i$ to its low-dimensional embedding $\boldsymbol{s}_i$ by a transformation matrix $\boldsymbol{P}$, i.e., $\boldsymbol{P}^T\boldsymbol{x}_i \to \boldsymbol{s}_i$. In this paper, we propose a dual-graph sparse regression method. In contrast to conventional sparse regression [11], our method simultaneously uses the low-dimensional embedding matrices $\boldsymbol{S}$ and $\boldsymbol{P}$. We can see that the transformation matrix is also denoted by $\boldsymbol{P}$. This is because the embedding matrix $\boldsymbol{P}$ can guide the learning of the transformation matrix. Therefore, we obtain the objective function of the regression as follows:

$$\arg \min_{\boldsymbol{P}} \sum_{i=1}^{n} \left\|\boldsymbol{P}^T\boldsymbol{x}_i - \boldsymbol{s}_i\right\|_2^2 = \left\|\boldsymbol{P}^T\boldsymbol{X} - \boldsymbol{S}\right\|_2^2. \tag{21}$$

Optimizing this objective function is equivalent to finding a suitable transformation matrix $\boldsymbol{P}$ which reduces the regression error below a certain threshold. By an appropriate operation, matrix $\boldsymbol{P}$ can be used to measure the importance of each feature. The use of sparse constraint with matrix $\boldsymbol{P}$ can more accurately reflect the importance of each feature, while the use of some important features can make the original data effectively regress to the low-dimensional space. Therefore, we apply the $L_{2,1}$-norm to matrix $\boldsymbol{P}$, which helps to avoid trivial solutions and ensure the sparsity of the feature array. Therefore, we obtain the new regression objective function as follows:

$$\begin{aligned} \arg \min_{\boldsymbol{P}} \ & \sum_{i=1}^{n} \left\|\boldsymbol{P}^T\boldsymbol{x}_i - \boldsymbol{s}_i\right\|_2^2 = \left\|\boldsymbol{P}^T\boldsymbol{X} - \boldsymbol{S}\right\|_2^2 \\ \text{s.t.} \ & \|\boldsymbol{P}\|_{2,1} \leq \varepsilon. \end{aligned} \tag{22}$$

Our NSSRD algorithm adopts a single step method, that is, the matrices $\boldsymbol{S}$ and $\boldsymbol{P}$ must be optimized within the same objective function. Therefore, we achieve joint dual-graph

non-negative spectral learning and dual-graph sparse regression. We obtain the final objective function as follows:

$$\begin{aligned}\arg\ \min_{\boldsymbol{S},\boldsymbol{P}}\ & \left\|\boldsymbol{P}^T\boldsymbol{X}-\boldsymbol{S}\right\|_2^2+\beta_1\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{L}^S\boldsymbol{S}^T\right)+\beta_2\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}\right)\\ &+\alpha\|\boldsymbol{P}\|_{2,1}+\frac{\lambda}{2}\left\|\boldsymbol{S}\boldsymbol{S}^T-\boldsymbol{I}_m\right\|_2^2\\ \text{s.t.}\ & \boldsymbol{S}\geq 0,\boldsymbol{P}\geq 0\end{aligned}\tag{23}$$

where the parameters $\beta_1>0,\beta_2>0,\alpha>0$, and $\lambda>0$. For ease of adjustment, we let $\beta_1=\beta_2=\beta$, and the objective function can be rewritten as follows:

$$\begin{aligned}\arg\ \min_{\boldsymbol{S},\boldsymbol{P}}\ & \left\|\boldsymbol{P}^T\boldsymbol{X}-\boldsymbol{S}\right\|_2^2+\beta\left(\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{L}^S\boldsymbol{S}^T\right)+\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}\right)\right)\\ &+\alpha\|\boldsymbol{P}\|_{2,1}+\frac{\lambda}{2}\left\|\boldsymbol{S}\boldsymbol{S}^T-\boldsymbol{I}_m\right\|_2^2\\ \text{s.t.}\ & \boldsymbol{S}\geq 0,\boldsymbol{P}\geq 0.\end{aligned}\tag{24}$$

C. Feature Selection

By optimizing the objective function of NSSRD, we can obtain the matrices $\boldsymbol{S}$ and $\boldsymbol{P}$, where $\boldsymbol{P}=[\boldsymbol{p}_1;\boldsymbol{p}_2;\ldots;\boldsymbol{p}_d]$, and $\boldsymbol{p}_i$ is the ith row of the matrix $\boldsymbol{P}$. Usually, $\|\boldsymbol{p}_i\|_2$ represents the contribution of the ith feature, and the greater the value of $\|\boldsymbol{p}_i\|_2$, the greater the contribution of the ith feature. So $\|\boldsymbol{p}_i\|_2$ can be used as the feature weights to rank features. We obtain all the weights of the d features by computing $\|\boldsymbol{p}_i\|_2$. By arranging weights in descending order and selecting features corresponding to the l ($l\leq d$) largest weights, we can obtain the new dataset, and complete the feature selection process.

D. Optimization

We now explain how the objective function of (24) is optimized. The problem is a nonconvex function of $\boldsymbol{S}$ and $\boldsymbol{P}$, making it nontrivial to find a globally optimal solution. Fortunately, the problem is convex individually for $\boldsymbol{S}$ and $\boldsymbol{P}$. Therefore, we propose an iterative and alternative optimization scheme to solve (24).

We introduce ψ_{ij} and ϕ_{ij} as the corresponding Lagrange multipliers for constraints $\boldsymbol{P}_{ij}\geq 0$ and $\boldsymbol{S}_{ij}\geq 0$, respectively. So (24) can be rewritten into a Lagrange function as follows:

$$\begin{aligned}L(\boldsymbol{S},\boldsymbol{P})=&\left\|\boldsymbol{P}^T\boldsymbol{X}-\boldsymbol{S}\right\|_2^2+\beta\left(\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{L}^S\boldsymbol{S}^T\right)+\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}\right)\right)\\ &+\alpha\|\boldsymbol{P}\|_{2,1}+\frac{\lambda}{2}\left\|\boldsymbol{S}\boldsymbol{S}^T-\boldsymbol{I}_m\right\|_2^2+\mathrm{Tr}\left(\psi\boldsymbol{P}^T\right)+\mathrm{Tr}\left(\phi\boldsymbol{S}^T\right).\end{aligned}\tag{25}$$

Before solving this problem, we introduce a diagonal matrix $\boldsymbol{U}\in\Re^{d\times d}$, whose ith diagonal element is defined as follows:

$$\boldsymbol{U}_{ii}=\frac{1}{2\left\|\boldsymbol{p}_i\right\|_2}.\tag{26}$$

To avoid overflow, we usually introduce a small constant ε in the definition of the matrix $\boldsymbol{U}$ as follows:

$$\boldsymbol{U}_{ii}=\frac{1}{2\max\left(\left\|\boldsymbol{p}_i\right\|_2,\varepsilon\right)}.\tag{27}$$

We rewrite $\|\boldsymbol{P}\|_{2,1}$ into $\mathrm{Tr}(\boldsymbol{P}^T\boldsymbol{U}\boldsymbol{P})$, and Lagrange formula (25) can be rewritten as follows:

$$\begin{aligned}L(\boldsymbol{S},\boldsymbol{P})=&\mathrm{Tr}\left(\left(\boldsymbol{P}^T\boldsymbol{X}-\boldsymbol{S}\right)\left(\boldsymbol{P}^T\boldsymbol{X}-\boldsymbol{S}\right)^T\right)+\beta\left(\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{L}^S\boldsymbol{S}^T\right)\right.\\ &\left.+\ \mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}\right)\right)+\frac{\lambda}{2}\mathrm{Tr}\left(\left(\boldsymbol{S}\boldsymbol{S}^T-\boldsymbol{I}_m\right)\left(\boldsymbol{S}\boldsymbol{S}^T-\boldsymbol{I}_m\right)^T\right)\\ &+\alpha\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{U}\boldsymbol{P}\right)+\mathrm{Tr}\left(\psi\boldsymbol{P}^T\right)+\mathrm{Tr}\left(\varphi\boldsymbol{S}^T\right).\end{aligned}\tag{28}$$

To update $\boldsymbol{P}$, we take the partial derivative of the Lagrange formula (28) with respect to $\boldsymbol{P}$, and arrive at

$$\frac{\partial L}{\partial\boldsymbol{P}}=2\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P}-2\boldsymbol{X}\boldsymbol{S}^T+2\beta\boldsymbol{L}^P\boldsymbol{P}+2\alpha\boldsymbol{U}\boldsymbol{P}+\psi.\tag{29}$$

Using the Karush–Kuhn–Tucker (KKT) conditions [27], $\psi_{ij}\boldsymbol{P}_{ij}=0$, we obtain

$$\left[\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P}-\boldsymbol{X}\boldsymbol{S}^T+\beta\boldsymbol{L}^P\boldsymbol{P}+\alpha\boldsymbol{U}\boldsymbol{P}\right]_{ij}\boldsymbol{P}_{ij}=0.\tag{30}$$

We then obtain the updating formula for $\boldsymbol{P}$ as follows:

$$\boldsymbol{P}_{ij}\leftarrow\boldsymbol{P}_{ij}\frac{\left[\boldsymbol{X}\boldsymbol{S}^T+\beta\boldsymbol{W}^P\boldsymbol{P}\right]_{ij}}{\left[\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P}+\beta\boldsymbol{D}^P\boldsymbol{P}+\alpha\boldsymbol{U}\boldsymbol{P}\right]_{ij}}.\tag{31}$$

To update $\boldsymbol{S}$, we take the partial derivative of the Lagrange formula (28) with respect to $\boldsymbol{S}$, giving

$$\frac{\partial L}{\partial\boldsymbol{S}}=-2\boldsymbol{P}^T\boldsymbol{X}+2\boldsymbol{S}+2\beta\boldsymbol{S}\boldsymbol{L}^S+2\lambda\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{S}-2\lambda\boldsymbol{S}+\varphi.\tag{32}$$

We also use the KKT conditions [27], $\phi_{jk}\boldsymbol{S}_{jk}=0$ and giving

$$\left[-\boldsymbol{P}^T\boldsymbol{X}+\boldsymbol{S}+\beta\boldsymbol{S}\boldsymbol{L}^S+\lambda\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{S}-\lambda\boldsymbol{S}\right]_{jk}\boldsymbol{S}_{jk}=0.\tag{33}$$

Therefore, we get the updating formula for $\boldsymbol{S}$ as follows:

$$\boldsymbol{S}_{jk}\leftarrow\boldsymbol{S}_{jk}\frac{\left[\boldsymbol{P}^T\boldsymbol{X}+\beta\boldsymbol{S}\boldsymbol{W}^S+\lambda\boldsymbol{S}\right]_{jk}}{\left[\boldsymbol{S}+\beta\boldsymbol{S}\boldsymbol{D}^S+\lambda\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{S}\right]_{jk}}.\tag{34}$$

To improve the learning efficiency and the convergence speed of the algorithm, we introduce a special method to initialize the matrices $\boldsymbol{S}$ and $\boldsymbol{P}$. For the matrix $\boldsymbol{S}$, we use the k-means algorithm to cluster the original data into c classes, and then get a good class label matrix. We then use this as the initialization of matrix $\boldsymbol{S}$. For the matrix $\boldsymbol{P}$, we calculate the eigenvalues and eigenvectors of graph Laplacian matrix $\boldsymbol{L}^P$, and select the eigenvectors corresponding to m maximum eigenvalues to form the eigenvector matrix. The d-by-m eigenvector matrix is used as the initialization of matrix $\boldsymbol{P}$. Additionally, we initialize $\boldsymbol{U}$ as an identity matrix.

Table I shows the procedure of NSSRD.

E. Convergence Analysis

In this section, we analyze the convergence properties of NSSRD. We prove that the objective function (24) is monotonically decreasing under the updating rules (31) and (34).

First, we prove that the objective function is monotonically decreasing under (31).

TABLE I
PROCEDURE OF NSSRD

Input: Data matrix $\boldsymbol{X}$; Neighborhood size k; Balance parameter α, β, λ; The maximum iteration number *NIter*; Selected feature number l.
Output: Index of selected features *index*; New data matrix $\boldsymbol{X}_{new}$.

1. Construct the nearest neighborhood graphs in data space and feature space.
2. Compute the similarity matrices $\boldsymbol{W}^S$, $\boldsymbol{W}^P$, graph Laplacian matrices $\boldsymbol{L}^S$, $\boldsymbol{L}^P$.
3. Initialize $\boldsymbol{S}$, $\boldsymbol{P}$, $\boldsymbol{U}$.
4. Update $\boldsymbol{S}$, $\boldsymbol{P}$ and $\boldsymbol{U}$ according to the iterative updating rules (27), (31) and (34), until the convergence conditions are satisfied.
5. Compute the scores for all the features in descending order according to $\|\boldsymbol{p}_i\|_2$, select the features corresponding to the largest l values and get a new data matrix $\boldsymbol{X}_{new}$.

Definition 1: If there is a function $G(h, h')$ which makes $F(h)$ satisfy the following conditions:

$$G(h, h') \geq F(h), G(h, h) = F(h) \tag{35}$$

then F is nonincreasing under the following updating formula:

$$h^{(t+1)} = \arg \min_h G\left(h, h^{(t)}\right) \tag{36}$$

where $G(h, h')$ is an auxiliary function for $F(h)$.

Proof: $F(h^{(t+1)}) \leq G(h^{(t+1)}, h^{(t)}) \leq G(h^{(t)}, h^{(t)}) = F(h^{(t)})$. ■

Considering that we only need to prove that the objective function is monotonic under the updating rules for $\boldsymbol{P}$, we only retain the objective function to contain the $\boldsymbol{P}$ term, and obtain the following functions:

$$F(\boldsymbol{P}) = \mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P} - 2\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{S}^T\right) + \beta\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}\right) + \alpha\mathrm{Tr}\left(\boldsymbol{P}^T\boldsymbol{U}\boldsymbol{P}\right). \tag{37}$$

The first-order and second-order partial derivatives for $\boldsymbol{F}(\boldsymbol{P})$ with respect to $\boldsymbol{P}$ are

$$F'_{ij} = \left[\frac{\partial F}{\partial \boldsymbol{P}}\right]_{ij} = \left[2\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P} - 2\boldsymbol{X}\boldsymbol{S}^T + 2\beta\boldsymbol{L}^P P + 2\alpha\boldsymbol{U}\boldsymbol{P}\right]_{ij} \tag{38}$$

$$F''_{ij} = 2\left[\boldsymbol{X}\boldsymbol{X}^T + \alpha\boldsymbol{U}\right]_{ii} + 2\beta\left[\boldsymbol{L}^P\right]_{jj}. \tag{39}$$

Lemma 1: The following function:

$$G\left(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^{(t)}\right) = F_{ij}\left(\boldsymbol{P}_{ij}^{(t)}\right) + F'_{ij}\left(\boldsymbol{P}_{ij}^{(t)}\right)\left(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^{(t)}\right) + \frac{\left[\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P} + \beta\boldsymbol{D}^P\boldsymbol{P} + \alpha\boldsymbol{U}\boldsymbol{P}\right]_{ij}}{\boldsymbol{P}_{ij}^{(t)}}\left(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^{(t)}\right)^2 \tag{40}$$

is the auxiliary function of $\boldsymbol{F}_{ij}$.

Proof: The Taylor expansion of $F_{ij}(\boldsymbol{P}_{ij})$ is

$$F_{ij}(\boldsymbol{P}_{ij}) = F_{ij}\left(\boldsymbol{P}_{ij}^{(t)}\right) + F'_{ij}\left(\boldsymbol{P}_{ij}^{(t)}\right)\left(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^{(t)}\right) + \left\{\left[\boldsymbol{X}\boldsymbol{X}^T + \alpha\boldsymbol{U}\right]_{ii} + \beta\left[\boldsymbol{L}^P\right]_{jj}\right\}\left(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^{(t)}\right)^2. \tag{41}$$

According to (40), $G(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^{(t)}) \geq F_{ij}(\boldsymbol{P}_{ij})$ is equivalent to

$$\frac{\left[\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P} + \beta\boldsymbol{D}^P\boldsymbol{P} + \alpha\boldsymbol{U}\boldsymbol{P}\right]_{ij}}{\boldsymbol{P}_{ij}^{(t)}} \geq \left[\boldsymbol{X}\boldsymbol{X}^T + \alpha\boldsymbol{U}\right]_{ii} + \beta\left[\boldsymbol{L}^P\right]_{jj}. \tag{42}$$

TABLE II
CHARACTERISTICS OF EIGHT DATASETS

dataset	Size	Dim	Class	Type
COIL20	1440	1024	20	Image
Isolet	1559	617	26	Voice
Umist	575	644	20	Image
ORL	400	1024	40	Image
PIE10P	210	2420	10	Image
Optdigit	3823	64	10	Image
Ionosphere	351	34	2	Text
AT&T	400	10304	40	Image

It is obvious that

$$\left[\left(\boldsymbol{X}\boldsymbol{X}^T + \alpha\boldsymbol{U}\right)\boldsymbol{P}\right]_{ij} = \sum_{l=1}^{d}\left[\boldsymbol{X}\boldsymbol{X}^T + \alpha\boldsymbol{U}\right]_{il}\boldsymbol{P}_{lj}^{(t)} \geq \left[\boldsymbol{X}\boldsymbol{X}^T + \alpha\boldsymbol{U}\right]_{ii}\boldsymbol{P}_{ij}^{(t)}$$

and

$$\beta\left[\boldsymbol{D}^P\boldsymbol{P}\right]_{ij} = \beta\sum_{l=1}^{d}\left[\boldsymbol{D}^P\right]_{il}\boldsymbol{P}_{lj}^{(t)} \geq \beta\boldsymbol{D}_{ii}^P\boldsymbol{P}_{ij}^{(t)} \geq \beta\left[\boldsymbol{D}^P - \boldsymbol{W}^P\right]_{ii}\boldsymbol{P}_{ij}^{(t)} = \beta\left[\boldsymbol{L}^P\right]_{ii}\boldsymbol{P}_{ij}^{(t)}.$$

Therefore, (42) holds and $G(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^{(t)}) \geq F_{ij}(\boldsymbol{P}_{ij})$, and we also have $G(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^{(t)}) \geq F_{ij}(\boldsymbol{P}_{ij})$.

Next, we prove that the variable $\boldsymbol{P}$ conforms to the updating rules (36) that make F_{ij} monotonically decreasing. ■

Proof: Substituting $G(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^{(t)})$ in (40) into (36), gives

$$\boldsymbol{P}_{ij}^{(t+1)} = \boldsymbol{P}_{ij}^{(t)} - \boldsymbol{P}_{ij}^{(t)}\frac{F'_{ij}\left(\boldsymbol{P}_{ij}^{(t)}\right)}{2\left[\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P} + \beta\boldsymbol{D}^P\boldsymbol{P} + \alpha\boldsymbol{U}\boldsymbol{P}\right]_{ij}} = \boldsymbol{P}_{ij}^{(t)}\frac{\left[\boldsymbol{X}\boldsymbol{S}^T + \beta\boldsymbol{W}^P\boldsymbol{P}\right]_{ij}}{\left[\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{P} + \beta\boldsymbol{D}^P\boldsymbol{P} + \alpha\boldsymbol{U}\boldsymbol{P}\right]_{ij}}.$$

From the updating rules for $\boldsymbol{P}$, we see that F_{ij} is monotonically decreasing under updating (31).

The proof of the convergence of the objective function to the updating rules of $\boldsymbol{S}$ is similar to that of $\boldsymbol{P}$. And we can also find that F_{ij} is monotonically decreasing under updating (34). Therefore we can conclude that the objective function is monotonically decreasing under (31) and (34). ■

IV. EXPERIMENTS AND ANALYSIS

In this section, we present the results of experiments to verify the performance of NSSRD. Specifically, we compare the performance of five state-of-the-art algorithms against that of NSSRD using public benchmark datasets. We choose *k-means* clustering algorithm [53] to verify the dimensionality reduction effect of all algorithms. We also provide an analysis of the experimental results, the computational complexity and the sensitivity of the algorithm to parameter values.

A. Dataset

In this experiment, we use eight datasets, which are similar to those in [10], [11], [40], [54], and [55], shown in Table II.

Table II describes the important information for the eight datasets, including the number of data samples, the dimension of each sample, the types and categories of each dataset. The information will be used in the following experiments.

Fig. 1. Two test samples from AT&T face database with different number of selected features.

B. Compared Algorithms

In order to validate the effectiveness of NSSRD, we choose four unsupervised feature selection algorithms, and a new NMF algorithm as the comparison algorithms. The five comparison algorithms are LapScor [13], MCFS [19], SPEC [10], JELSR [11], and AGNMF_{FS} [37].

C. Evaluation Metrics

We use clustering accuracy (ACC) [56] and normalized mutual information (NMI) [57]–[59] as metrics to indirectly evaluate the results of all of the algorithms. The higher the value of ACC or NMI is, the better the performance of the algorithm, and vice versa.

ACC is defined as

$$\text{ACC} = \frac{1}{n}\sum_{i=1}^{n}\delta(c_i, \text{map}(g_i)) \tag{43}$$

where c_i is the clustering label and g_i is the ground truth label of $\boldsymbol{x}_i$. map$(\cdot)$ is the optimal mapping function using *Hungarian* algorithm [60] to permute clustering labels and the ground truth labels. $\delta(c_i, g_i)$ is an indicator function that equals 1 if $c_i = g_i$ and equals 0 if $c_i \neq g_i$.

NMI is defined as

$$\text{NMI} = \frac{\text{MI}(C, C')}{\max(H(C), H(C'))} \tag{44}$$

where C and C' are clustering labels and the ground truth labels, respectively. MI(C, C') is the information entropy between C and C', and

$$\text{MI}(C, C') = \sum_{c_i \in C, c'_j \in C'} p\left(c_i, c'_j\right).\log_2 \frac{p\left(c_i, c'_j\right)}{p(c_i).p\left(c'_j\right)} \tag{45}$$

where $p(c_i)$ and $p(c'_j)$ denote the probabilities that a sample belongs to the clusters c_i and c'_j, respectively. $p(c_i, c'_j)$ is the joint probability that a sample belongs to the clusters c_i and c'_j simultaneously.

D. Comparisons With Four Feature Selection Algorithms

1) Experimental Settings: In this experiment, we need to obtain a low-dimensional spectrum embedding matrix $\boldsymbol{S}$ which is close to the ideal label matrix, so we set m equal to the true number of clusters. α is searched from {110, 120, 150, 180, 190, 500, 800}. We tune β in the range of $\{10^{-4}, 10^{-3}, 10^{-1}, 10^{2}, 10^{3}, 10^{7}\}$. λ is searched from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{3}\}$. We tune the parameter σ in the range of $\{10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}, 10^{6}, 10^{7}, 10^{8}\}$. All the parameters under different datasets are obtained by grid search. We tune the feature selection parameter l in the range of {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}. The neighborhood size k is set as 5. All the results are obtained with 20 iterations. We repeat the clustering for 100 runs independently to get the average value, since the performance of the k-means algorithm largely depends on initialization. We fix l, and tune the other parameters so that the algorithms have the best ACC and NMI.

2) Simple Illustrative Example Problem: We randomly chose two images from the AT&T face database as test samples. From reshaping the two images, we get two single vectors to represent the images, and each single vector is of the size 10 304. By using the proposed algorithm on the test samples, we select {1280, 2560, 3840, 5120, 6400, 7680, 8960, 10 240} features, respectively. We set the unselected features to white and maintain their selected features with original values. We illustrate the results in Fig. 1.

Fig. 1 shows that NSSRD can effectively select the important features of each face when we fixed each number of selected features, such as nose, mouth, eyes, and chin. These discriminative features can be used to effectively describe the individual's appearance.

3) Evaluating the Effectiveness of NSSRD: In this section, we first verify the effectiveness of our feature selection algorithm through a test experiment. We use the "ionosphere" dataset [54] as an example to test whether NSSRD can successfully find the most representative features. Table II shows that the ionosphere dataset has 351 samples and 34 features. We artificially generate 66 features, where each new feature is the linear combination of the 34 original features with a series of randomly generated combination coefficients, where the combination coefficients are normalized. This yields a new dataset, which has 351 samples and 100 features. The first 34 features are the original features and the rest are synthetic features. We apply NSSRD to this new dataset, and generate the sparse transformation matrix $\boldsymbol{P}$. By computing $\|\boldsymbol{p}_i\|_2$, we obtain scores for all 100 features. We use these scores to generate a diagonal matrix shown in Fig. 2.

TABLE III
CLUSTERING ACC OF SIX ALGORITHMS ON COIL20 DATASET WITH DIFFERENT NUMBERS OF SELECTED FEATURES (MEAN ± STD %)

l	LapScor	SPEC	MCFS	JELSR	NSSRD_PF	NSSRD
l=5	34.07±1.31	38.16±8.40	47.85±5.61	50.21±1.04	53.95±2.61	**57.91**±1.93
l=15	43.91±1.71	46.79±6.99	55.50±2.09	58.87±1.88	61.35±2.07	**65.43**±2.24
l=25	46.77±1.58	50.37±7.24	57.33±2.13	59.98±2.33	63.39±2.11	**65.71**±2.82
l=35	53.53±1.71	52.27±6.24	59.22±3.64	62.03±2.75	63.22±2.60	**65.25**±3.25
l=45	55.15±2.91	53.44±6.17	61.46±2.91	63.08±2.87	65.91±3.21	**65.73**±2.99

TABLE IV
CLUSTERING ACC OF SIX ALGORITHMS ON ISOLET DATASET WITH DIFFERENT NUMBERS OF SELECTED FEATURES (MEAN ± STD %)

l	LapScor	SPEC	MCFS	JELSR	NSSRD_PF	NSSRD
l=5	31.02±0.81	19.79±0.59	31.20±0.88	35.59±1.21	32.07±1.15	**41.61**±0.92
l=15	44.43±1.65	29.93±1.22	42.13±1.29	46.58±1.57	42.45±1.12	**52.51**±2.07
l=25	45.41±1.54	34.75±1.16	50.61±2.19	55.03±2.51	54.44±2.27	**58.74**±2.08
l=35	44.88±1.61	38.18±1.22	56.19±1.93	57.63±2.71	50.40±2.20	**60.62**±2.21
l=45	47.19±1.44	42.55±1.83	54.86±1.88	56.52±2.88	53.19±1.82	**62.94**±2.11

TABLE V
CLUSTERING ACC OF SIX ALGORITHMS ON UMIST DATASET WITH DIFFERENT NUMBERS OF SELECTED FEATURES (MEAN ± STD %)

l	LapScor	SPEC	MCFS	JELSR	NSSRD_PF	NSSRD
l=5	35.23±1.34	32.46±5.03	45.20±1.80	48.36±2.37	49.74±1.97	**52.11**±2.05
l=15	35.34±2.14	36.39±5.07	41.69±2.09	48.81±2.51	49.37±1.96	**51.61**±2.65
l=25	35.23±1.81	38.15±4.98	51.23±2.86	52.03±1.72	50.00±2.43	**55.09**±2.02
l=35	34.54±1.77	39.61±5.31	48.61±2.38	50.31±2.73	50.22±2.40	**55.23**±2.57
l=45	35.49±1.84	40.38±4.77	48.85±3.31	50.83±2.61	50.94±2.64	**53.68**±2.91

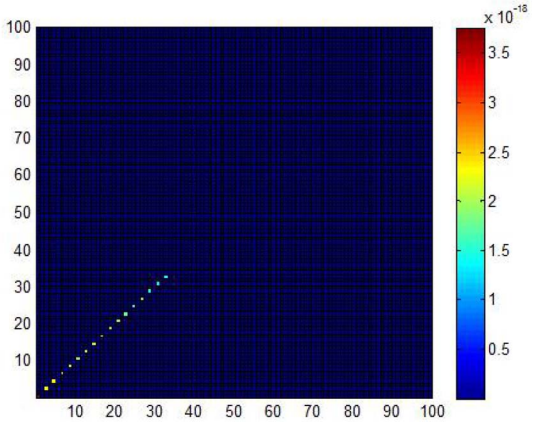


Fig. 2. Score diagonal matrix of 100 features.

From Fig. 2, we can clearly see that the original features have significantly larger scores than the synthetic features. This suggests that NSSRD can effectively select the most representative features.

4) Clustering Results and Analysis: Clustering results are presented in Tables III–VIII. We give the results of NSSRD using Gaussian function and parameter free method [52] as metric methods, respectively, which are denoted as NSSRD and NSSRD_PF. The bold numbers denote the highest statistics. The results are visualized in Figs. 3 and 4.

Tables III–V show the values of ACC of six algorithms, respectively, on COIL20, Isolet, and Umist datasets [11] with different numbers of selected features. From Table II, we can know that these three datasets are relatively easy to handle because their features are less than or similar to the number of samples. Therefore, all of the compared algorithms give good performance. However, we can clearly see that the performance of NSSRD on all of the datasets is better than that of the other algorithms. NSSRD_PF also performs well, and it reduces a parameter σ that needs to be adjusted. It is evident that NSSRD has good performance, which demonstrates its effectiveness. NSSRD, JELSR, and MCFS have better feature selection quality than the other algorithms, which suggests that a good learning mechanism is very important for feature selection. We know that MCFS is a two-stage feature selection algorithms, while NSSRD and JELSR unify embedded learning and sparse regression simultaneously to solve two objective functions. Overall, NSSRD and JELSR have better feature selection performance than MCFS, which suggests that the use of a single-step strategy to optimize the embedding matrix and transformation matrix produces a better learning effect. Compared with JELSR, the main improvement is that NSSRD utilizes the information in the feature space. The feature selection performance of NSSRD is better than that of JELSR, suggesting that the information in the feature space is of great importance for feature selection.

Tables VI and VII show the clustering ACC of six algorithms, respectively, on the ORL and PIE10P datasets, with different numbers of selected features. The two datasets have a common characteristic, that the number of samples is much less than the number of features. It is relatively difficult to do feature selection for this kind of dataset, because there are a lot of features that are redundant, and some features may even represent noise. From Tables VI and VII, we can see that, in most cases, NSSRD and NSSRD_PF perform better than the other four algorithms.

Table VIII shows the clustering ACC of six algorithms, respectively, on the Optdigit dataset with different numbers of selected features. The number of features in this dataset is

TABLE VI
CLUSTERING ACC OF SIX ALGORITHMS ON ORL DATASET WITH DIFFERENT NUMBERS OF SELECTED FEATURES (MEAN ± STD %)

l	LapScor	SPEC	MCFS	JELSR	NSSRD_PF	NSSRD
l=5	37.97±1.53	30.83±2.58	33.09±1.33	42.81±1.92	**43.54**±1.57	42.18±1.52
l=15	39.91±1.49	36.19±2.72	42.84±2.11	48.47±2.44	48.76±1.87	**49.51**±2.18
l=25	41.18±1.81	38.86±3.20	47.05±2.11	47.65±2.34	49.19±2.29	**52.92**±2.55
l=35	41.01±1.72	40.63±3.27	49.19±2.12	49.67±2.71	49.74±2.06	**53.02**±2.16
l=45	44.28±2.03	42.62±3.19	50.77±2.17	50.52±2.66	49.06±2.00	**52.09**±2.85

TABLE VII
CLUSTERING ACC OF SIX ALGORITHMS ON PIE10P DATASET WITH DIFFERENT NUMBERS OF SELECTED FEATURES (MEAN ± STD %)

l	LapScor	SPEC	MCFS	JELSR	NSSRD_PF	NSSRD
l=5	36.93±1.82	32.04±2.09	32.18±1.97	40.72±4.11	40.93±1.36	**47.31**±1.78
l=15	38.91±1.66	**50.28**±2.45	49.86±2.48	46.27±5.35	37.34±2.48	49.57±3.02
l=25	36.46±1.96	45.27±2.23	45.59±2.34	46.40±4.44	41.98±2.24	**49.41**±1.89
l=35	33.25±2.89	42.55±2.50	42.28±2.84	46.66±5.17	47.24±2.24	**49.71**±2.13
l=45	34.28±2.78	44.67±1.79	45.06±1.72	44.75±3.47	47.20±2,61	**49.59**±2.66

TABLE VIII
CLUSTERING ACC OF SIX ALGORITHMS ON OPTDIGIT DATASET WITH DIFFERENT NUMBERS OF SELECTED FEATURES (MEAN ± STD %)

l	LapScor	SPEC	MCFS	JELSR	NSSRD_PF	NSSRD
l=5	48.69±0.12	26.48±2.41	51.95±0.78	**53.52**±0.99	52.92±0.69	53.16±0.91
l=15	74.04±0.37	58.06±2.57	69.87±0.29	69.92±0.21	73.78±0.18	**77.95**±0.73
l=25	80.91±1.33	73.39±3.81	79.63±0.48	72.22±1.57	79.44±0.16	**85.38**±2.09
l=35	80.57±0.16	79.87±1.54	78.73±0.79	79.56±1.49	80.78±0.55	**87.18**±2.36
l=45	80.19±0.71	80.19±0.87	80.66±0.76	80.15±0.89	80.19±0.70	**84.12**±2.27

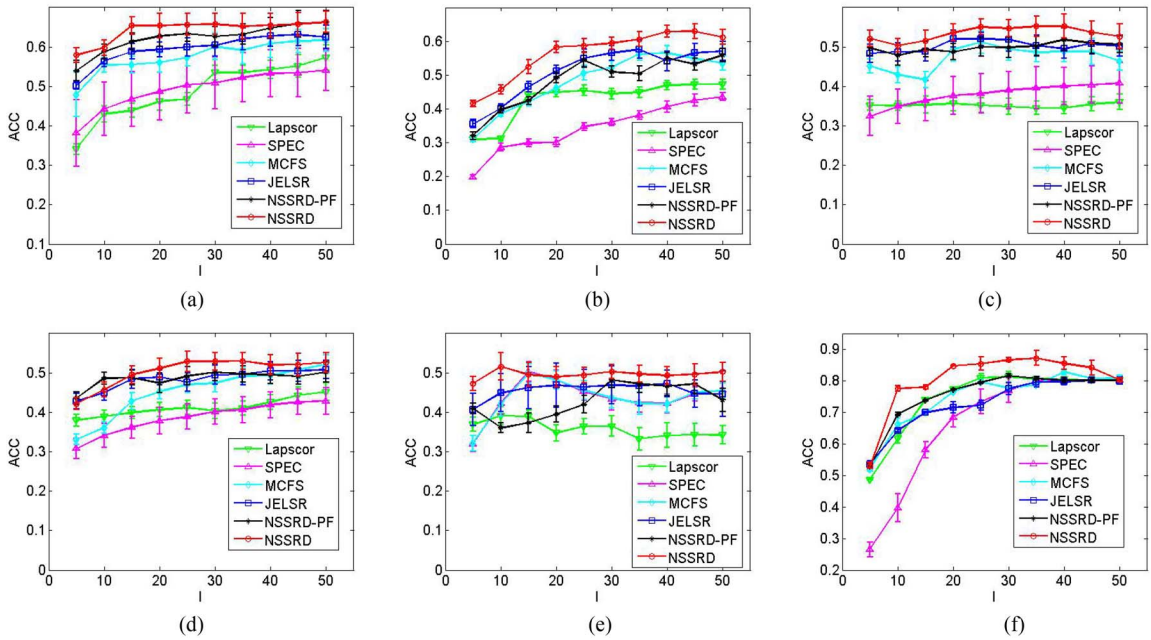


Fig. 3. Clustering ACC of six algorithms on six datasets with different numbers of selected features. x-axis is the number of selected features l and y-axis is the ACC. (a) COIL20. (b) Isolet. (c) Umist. (d) ORL. (e) PIE10P. (f) Optdigit.

far less than the number of samples, which is advantageous for feature selection. Therefore, all of the six algorithms have achieved good results; however, NSSRD is also clearly the best. The selection of a small number of features is able to represent the entire dataset, which makes the clustering effect greatly improved. This also helps illustrate the significance of feature selection.

Fig. 3 visually shows the clustering ACC of six algorithms, respectively, on six datasets with different numbers of selected features. We use six curves with different colors and shapes to express the corresponding six algorithms. Feature selection parameter l is in the range of {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}. In Fig. 3, we can see that the red curve of NSSRD is almost always above the other curves. The blue curve represents JELSR, which is somewhat lower than the red curve. Overall, the ACC of NSSRD is higher than the other algorithms, and demonstrates the effectiveness of NSSRD.

Fig. 4 shows the clustering NMI of six algorithms, respectively, on six datasets with different numbers of selected features. In Fig. 4, we also use a red curve to represent

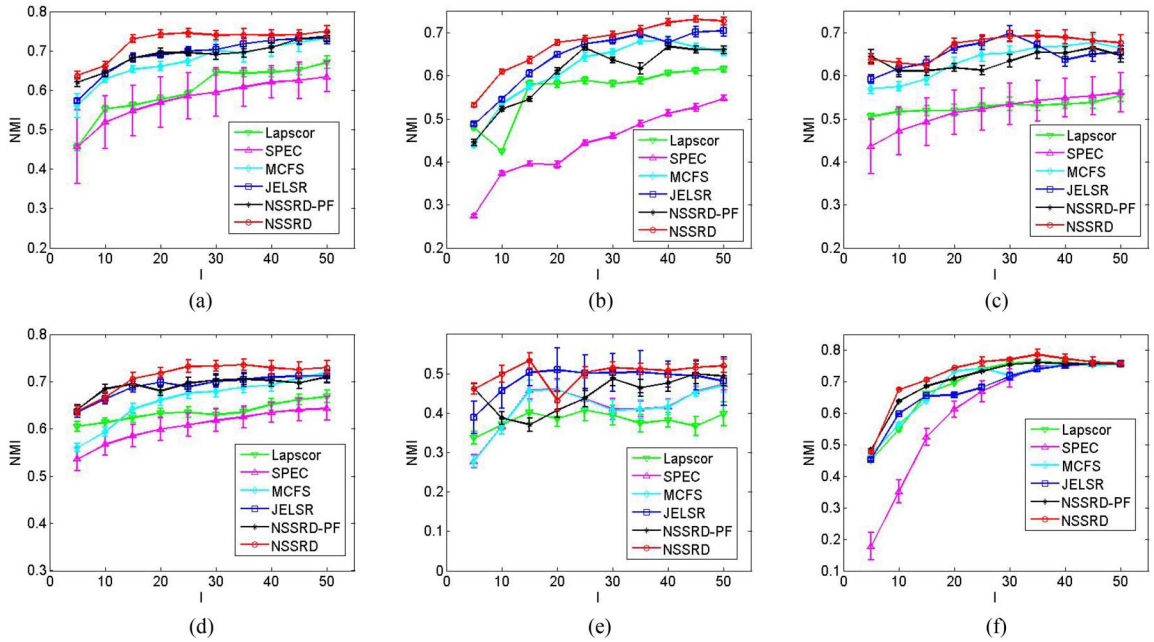


Fig. 4. Clustering NMI of six algorithms on six datasets with different numbers of selected features. x-axis is the number of selected features l and y-axis is the NMI. (a) COIL20. (b) Isolet. (c) Umist. (d) ORL. (e) PIE10P. (f) Optdigit.

TABLE IX
COMPUTATIONAL COMPLEXITY ANALYSIS

Algorithms	Computational complexity (O)
LapScor	$O(dn^2)$
SPEC	$O(dn^2)$
MCFS	$O(dn^2+ml^3+mnl^2)$
JELSR	$O(dn^2+t(n^3+mdn))$
NSSRD	$O(d^2n+dn^2+tmdn)$

TABLE X
COMPUTATION TIME OF FIVE ALGORITHMS ON OPTDIGIT DATASET WITH DIFFERENT NUMBERS OF SELECTED FEATURES (S)

l	LapScor	SPEC	MCFS	JELSR	NSSRD
l=10	0.6596	0.6137	1.4527	217.44	0.7375
l=30	0.7226	0.8861	1.4867	245.15	0.8295
l=50	0.8771	1.0309	1.8016	248.52	0.9326

NSSRD, which is predominantly higher than the other curves on the vertical axis. The clustering NMI of NSSRD on the six datasets are higher than those of the other algorithms. The results show that NSSRD is highly competitive with the compared algorithms.

5) Computational Complexity Analysis: The computational complexity of the five algorithms is shown in Table IX, and the specific experimental results of NSSRD are given to verify our analysis.

In Table IX, n is the number of samples, m represents the dimension of the embedding space, d is the total number of features, l represents the number of the selected features, and t is the number of iterations, $(n, d >> m, d \geq l)$. We will mainly analyze the computational complexity of NSSRD. First, we need $O(d^2n + dn^2)$ operations to build Laplacian matrices $\boldsymbol{L}^S$ and $\boldsymbol{L}^P$. Next, we need $O(mdn)$ operations to calculate each iteration of the alternating iteration step. Assuming NSSRD is iterated t times, the overall computational complexity of NSSRD is $O(d^2n + dn^2 + tmdn)$. Therefore, the computational complexity of NSSRD is lower than that of JELSR.

Table X shows the computation time of five algorithms on the Optdigit dataset with different numbers of selected features. We can see that the computation time needed by NSSRD is similar to that of LapScor and SPEC, and only half that of MCFS.

Table X shows that NSSRD adopts an iterative and alternative updating rule to optimize the objective function, which makes the algorithm converge faster and reduces the time complexity. The computational time of JELSR is over 296 times longer than that of NSSRD, which highlights the efficiency of our proposed algorithm.

6) Parameters Sensitivity Analysis: There are some parameters which need to be set in advance for NSSRD, such as neighborhood size k, Gaussian kernel bandwidth parameter σ, balance parameters α, β, and λ, and the number of selected feature parameter l. First, we discuss the sensitivity of α and β. We select the COIL20 and Umist datasets as test examples. The sensitivity of the parameters can be analyzed by the clustering ACC and NMI on each dataset under different values of α and β. We vary α in the range of {100, 300, 500, 700, 900}, and chose β from a wide range $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{+1}, 10^{+2}\}$.

We repeated 20 independent runs of each experiment, to get an average value, ploted in the 3-D figures in Fig. 5(a) and (b) to show the ACC and NMI of clustering, respectively, on COIL20 dataset under different values of α and β. Fig. 5(c) and (d) shows the ACC and NMI of clustering, respectively, on the Umist dataset under different values of α and β. Fig. 5 shows that on the COIL20 and Umist datasets, the ACC and NMI of clustering have little change under different values of α and β, which demonstrates that

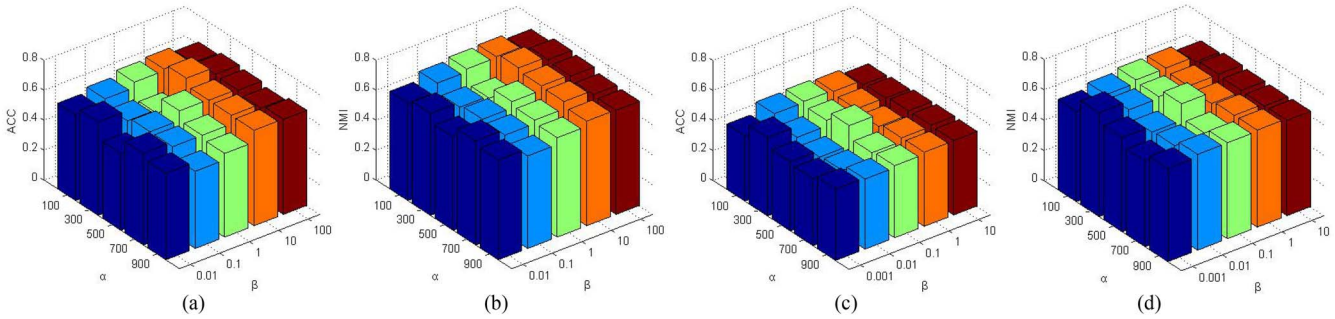


Fig. 5. ACC and NMI of clustering on COIL20 and Umist datasets under different values of α and β.

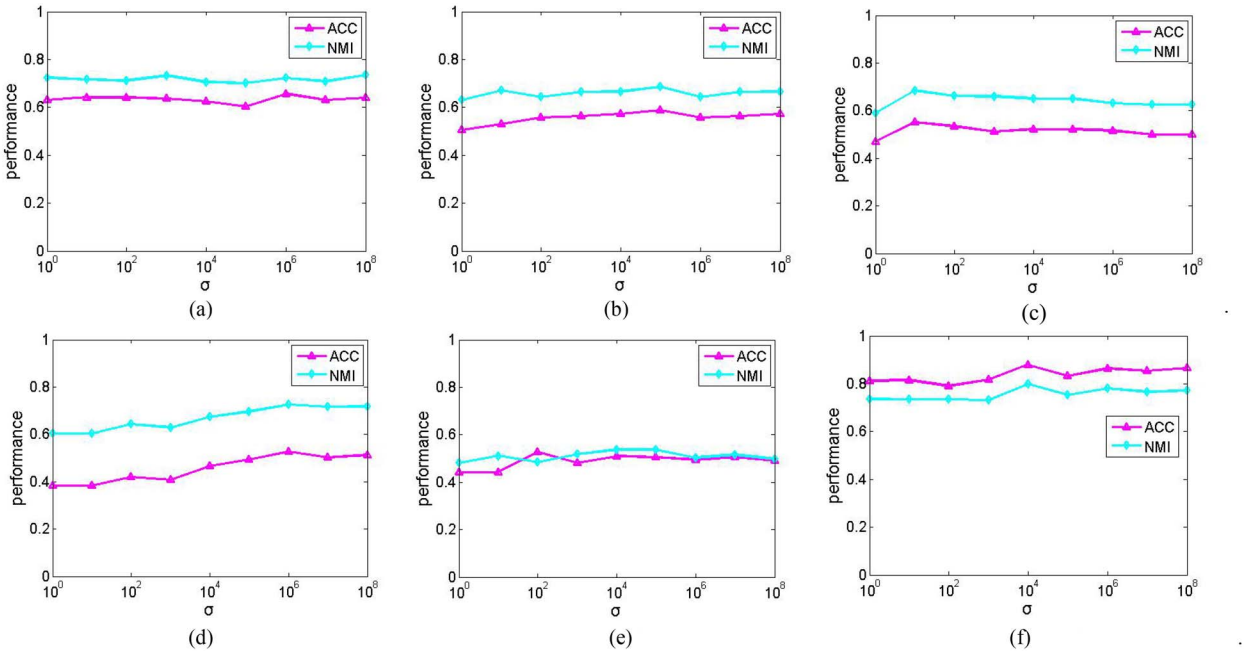


Fig. 6. ACC and NMI of clustering on six datasets under different values of σ. (a) COIL20. (b) Isolet. (c) Umist. (d) ORL. (e) PIE10P. (f) Optdigit.

TABLE XI
CLUSTERING ACC AND NMI OF TWO ALGORITHMS ON SIX DATASETS (MEAN ± STD %)

Results	Algorithms	COIL20	Isolet	Umist	ORL	PIE10P	Optdigit
ACC	NSSRD	**66.31**±2.63	**62.94**±2.11	**55.26**±2.96	**53.02**±2.16	**51.62**±3.43	**87.18**±2.36
	$AGNMF_{FS}$	54.51±2.91	48.26±1.80	52.75±3.69	52.57±2.44	49.45±5.22	80.27±0.50
NMI	NSSRD	**74.99**±1.43	**73.15**±0.76	69.35±1.18	**73.56**±1.23	53.35±1.92	**78.63**±1.56
	$AGNMF_{FS}$	68.30±1.96	65.53±0.58	**70.01**±2.17	72.22±1.34	**60.89**±3.96	75.77±0.31

NSSRD is relatively insensitive to the choice of parameters α and β.

Next, we discuss the sensitivity of the algorithm to parameter σ. We performed experiments on all six datasets. We tune σ in the range of $\{10^0, 10^1, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8\}$ and the other parameters remain fixed.

Fig. 6 shows the clustering ACC and NMI of NSSRD on six datasets with different values of σ. From Fig. 6, we can see that under different values of σ, ACC and NMI show little change on most of the datasets. This suggests that NSSRD is not sensitive to the parameter σ.

E. Comparisons With $AGNMF_{FS}$

NSSRD and $AGNMF_{FS}$ [37] are both dimension reduction methods. Therefore, we compare the performance of dimensionality reduction of the two algorithms. We record the best clustering results of NSSRD and $AGNMF_{FS}$ from the optimal parameters and show these in Table XI. The best ACC and NMI are highlighted in bold.

From Table XI, we can see that the results of NSSRD on almost all datasets are better than those of $AGNMF_{FS}$, except for the NMI values on datasets Umist and PIE10P. This suggests that NSSRD has a better effect of dimension reduction than $AGNMF_{FS}$ on many kinds of data.

V. CONCLUSION

In this paper, we have proposed a novel feature selection algorithm named NSSRD. Inspired by the idea of the dual-graph regularized algorithms, we introduce the feature graph based on an unsupervised feature selection framework: JELSR.

By making full use of underlying information of feature manifold and the advantages of this framework, we obtain a more efficient unsupervised feature selection algorithm. We construct the nearest neighborhood graphs in both data space and feature space, respectively, and compute the Laplacian matrices $\boldsymbol{L}^S$ and $\boldsymbol{L}^P$. By embedding the data space and feature space, respectively, into low-dimension spaces, we get the embedding matrices $\boldsymbol{S}$ and $\boldsymbol{P}$. We use non-negative and orthogonal constraints to constrain the embedding matrix $\boldsymbol{S}$, which helps $\boldsymbol{S}$ become much closer to the ideal label matrix, providing accurate discrimination information for feature selection. In addition, we use the embedding matrix $\boldsymbol{P}$ to represent the transformation matrix in the regression step. Thus, the manifold information of the feature space can guide the learning of the transformation matrix. The use of non-negative constraints and $L_{2,1}$-norm constraints ensures non-negative values of $\boldsymbol{P}$ and the sparsity of the feature array. The manifold information of data space and feature space are both fully exploited within the learning process. The use of the alternating iterative updating rule makes the algorithm converge faster in the optimization process and reduces the computational complexity. The experimental results show that the proposed algorithm outperforms several other unsupervised feature selection algorithms on a variety of datasets.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments, which have greatly helped them in improving the quality of this paper.

REFERENCES

[1] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997.

[2] H. Yan and J. Yang, "Sparse discriminative feature selection," *Pattern Recognit.*, vol. 48, no. 5, pp. 1827–1835, 2015.

[3] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l_2,1-norms minimization," in *Proc. Adv. NIPS*, Vancouver, BC, Canada, 2010, pp. 1813–1821.

[4] X. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 4, pp. 1170–1175, Aug. 2010.

[5] F. Nie, D. Xu, I. W.-H. Tsang, and C. Zhang, "Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1921–1932, Jul. 2010.

[6] F. Nie, S. Xiang, Y. Liu, C. Hou, and C. Zhang, "Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction," *Pattern Recognit. Lett.*, vol. 33, no. 5, pp. 485–491, 2012.

[7] B. Gu and V. S. Sheng, "A robust regularization path algorithm for ν-support vector classification," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2527796.

[8] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1403–1416, Jul. 2015.

[9] J. J.-Y. Wang, J. Yao, and Y. J. Sun, "Semi-supervised local-learning-based feature selection," in *Proc. IJCNN*, Beijing, China, 2014, pp. 1942–1948.

[10] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proc. Int. Conf. Mach. Learn.*, Corvallis, OR, USA, 2007, pp. 1151–1157.

[11] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[12] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, "Clustering-guided sparse structural learning for unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2138–2150, Sep. 2014.

[13] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12. Vancouver, BC, Canada, 2005, pp. 507–514.

[14] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *Proc. Assoc. Adv. Artif. Intell.*, vol. 7. Chicago, IL, USA, 2008, pp. 671–676.

[15] R. H. Shang, Z. Zhang, L. C. Jiao, C. Y. Liu, and Y. Y. Li, "Self-representation based dual-graph regularized feature selection clustering," *Neurocomputing*, vol. 171, pp. 1242–1253, Jan. 2016.

[16] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.

[17] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[18] C. Hou, F. Nie, D. Yi, and Y. Wu, "Feature selection via joint embedding learning and sparse regression," in *Proc. IJCAI*, Barcelona, Spain, 2011, pp. 1324–1329.

[19] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. KDD*, Washington, DC, USA, 2010, pp. 333–342.

[20] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proc. Assoc. Adv. Artif. Intell.*, Atlanta, GA, USA, 2010, pp. 673–678.

[21] S. Z. Yang, C. P. Hou, F. Nie, and Y. Wu, "Unsupervised maximum margin feature selection via $L_{2,1}$-norm minimization," *Neural Comput. Appl.*, vol. 21, no. 7, pp. 1791–1799, 2012.

[22] F. Nie, C. H. Q. Ding, D. J. Luo, and H. Huang, "Improved minmax cut graph clustering with nonnegative relaxation," in *Proc. ECML/PKDD*, Barcelona, Spain, 2010, pp. 451–466.

[23] D. Wang, F. Nie, and H. Huang, "Unsupervised feature selection via unified trace ratio formulation and k-means clustering (TRACK)," in *Proc. ECML/PKDD*, Nancy, France, 2014, pp. 306–321.

[24] Y. Q. Jia, F. Nie, and C. S. Zhang, "Trace ratio problem revisited," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 729–735, Apr. 2009.

[25] F. Nie, W. Zhu, and X. L. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 1302–1308.

[26] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 6, pp. 902–913, Jun. 2011.

[27] W. Xu and Y. Gong, "Document clustering by concept factorization," in *Proc. SIGIR*, Sheffield, U.K., 2004, pp. 202–209.

[28] J. Ye and Z. Jin, "Dual-graph regularized concept factorization for clustering," *Neurocomputing*, vol. 138, pp. 120–130, Aug. 2014.

[29] J. J.-Y. Wang, H. Bensmail, N. Yao, and X. Gao, "Discriminative sparse coding on multi-manifolds," *Knowl. Based Syst.*, vol. 54, pp. 199–206, Dec. 2013.

[30] J. J.-Y. Wang, X. L. Wang, and X. Gao, "Non-negative matrix factorization by maximizing correntropy for cancer clustering," *BMC Bioinformat.*, vol. 14, no. 107, p. 107, 2013.

[31] J. J.-Y. Wang and X. Gao, "Learning manifold to regularize nonnegative matrix factorization," *CoRR abs/1410.2191*, 2014.

[32] J. J.-Y. Wang, H. Bensmail, and X. Gao, "Feature selection and multi-kernel learning for sparse representation on a manifold," *Neural Netw.*, vol. 51, pp. 9–16, Mar. 2014.

[33] J. J.-Y. Wang and X. Gao, "Beyond cross-domain learning: Multiple-domain nonnegative matrix factorization," *Eng. Appl. Artif. Intell.*, vol. 28, pp. 181–189, Feb. 2014.

[34] J. J.-Y. Wang and X. Gao, "Max–min distance nonnegative matrix factorization," *Neural Netw.*, vol. 61, pp. 75–84, Jan. 2015.

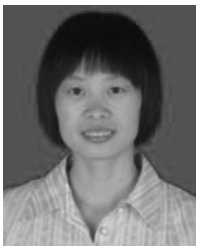
[35] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[36] J.-Y. Wang, I. Almasri, and X. Gao, "Adaptive graph regularized nonnegative matrix factorization via feature selection," in *Proc. ICPR*, Tsukuba, Japan, 2012, pp. 963–966.

[37] J. J.-Y. Wang, J. H. Z. Huang, Y. J. Sun, and X. Gao, "Feature selection and multi-kernel learning for adaptive graph regularized nonnegative matrix factorization," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1278–1286, 2015.

[38] J. J.-Y. Wang, H. Bensmail, and X. Gao, "Multiple graph regularized nonnegative matrix factorization," *Pattern Recognit.*, vol. 46, no. 10, pp. 2840–2847, 2013.

[39] J. Huang, F. Nie, H. Huang, and C. H.-Q. Ding, "Robust manifold nonnegative matrix factorization," *ACM Trans. Knowl. Disc. Data*, vol. 8, no. 3, 2013, Art. no. 11.
[40] F. Shang, L. C. Jiao, and F. Wang, "Graph dual regularization nonnegative matrix factorization for co-clustering," *Pattern Recognit.*, vol. 45, no. 6, pp. 2237–2250, 2012.
[41] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14. Vancouver, BC, Canada, 2001, pp. 585–591.
[42] P. Niyogi and X. He, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16. Whistler, BC, Canada, 2003, pp. 153–160.
[43] B. Gu, X. M. Sun, and V. S. Sheng, "Structural minimax probability machine," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2544779.
[44] B. Gu *et al.*, "Incremental learning for ν-support vector regression," *Neural Netw.*, vol. 67, pp. 140–150, Jul. 2015.
[45] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.
[46] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 342–352, Apr. 2008.
[47] F. Shang, Y. Liu, and F. Wang, "Learning spectral embedding for semi-supervised clustering," in *Proc. IEEE 11th Int. Conf. Data Min.* (*ICDM*), Vancouver, BC, Canada, 2011, pp. 597–606.
[48] L. F. Bo and C. Sminchisescu, "Supervised spectral latent variable models," in *Proc. AISTATS*, Apr. 2009, pp. 33–40.
[49] X. F. He, D. Cai, Y. L. Shao, H. J. Bao, and J. W. Han, "Laplacian regularized Gaussian mixture model for data clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1406–1418, Sep. 2011.
[50] P. Li, J. Bu, C. Chen, Z. He, and D. Cai, "Relational multimanifold coclustering," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1871–1881, Dec. 2013.
[51] C. Ding, D. Zhou, X. He, and H. Zha, "R1-PCA: Rotational invariant l1-norm principal component analysis for robust subspace factorization," in *Proc. Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, 2006, pp. 281–288.
[52] F. Nie, X. Q. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 1969–1976.
[53] A. Rakhlin and A. Caponnetto, "Stability of K-means clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12. 2007, pp. 216–222.
[54] L. Du, Z. Shen, X. Li, P. Zhou, and Y. D. Shen, "Local and global discriminative learning for unsupervised feature selection," in *Proc. IEEE Int. Conf. Data Min.*, Dallas, TX, USA, 2013, pp. 131–140.
[55] H. Lu, Z. Fu, and X. Shu, "Non-negative and sparse spectral clustering," *Pattern Rcognit.*, vol. 47, no. 1, pp. 418–426, 2014.
[56] M. Wu and B. Schölkopf, "A local learning approach for clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19. Vancouver, BC, Canada, 2006, pp. 1529–1536.
[57] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.
[58] R. H. Shang, Z. Zhang, L. C. Jiao, W. B. Wang, and S. Y. Yang, "Global discriminative-based nonnegative spectral clustering," *Pattern Recognit.*, vol. 55, pp. 172–182, Jul. 2016.
[59] M. Liang and X. Hu, "Feature selection in supervised saliency prediction," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 914–926, May 2015.
[60] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. New York, NY, USA: Dover, 1998.



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, Xi'an, China, 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.

Wenbing Wang received the B.S. degree from the School of Electronic Engineering, Xidian University, Xi'an, China.

His current research interests include pattern recognition, machine learning, and data mining.

Rustam Stolkin (M'12) received the bachelor's and master's degrees in engineering science from the University of Oxford, Oxford, U.K., in 1998, and the Ph.D. degree in robotic vision from University College London, London, U.K., in 2004.

He is a Senior Birmingham Fellow with the School of Mechanical Engineering, University of Birmingham, Birmingham, U.K., researching on robotics and machine intelligence. From 2004 to 2008, he was a Research Assistant Professor with the Stevens Institute of Technology, Hoboken, NJ, USA, where he researched on sensor systems for maritime security. His current research interests include science and engineering outside of robotics, as well as manipulation with robotic arms and hands, novel robotic vehicles, computer vision, and other kinds of autonomous sensing.

Licheng Jiao (SM'89) received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

He was a Post-Doctoral Fellow with the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, from 1990 to 1991, where he has been a Professor with the School of Electronic Engineering, since 1992, and currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education of China. He has charged of about 40 important scientific research projects, and published over 20 monographs and a hundred papers in international journals and conferences. His current research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is the Chairman of Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an expert of the Academic Degrees Committee of the State Council. He is a member of the IEEE Xi'an Section Execution Committee.