Neurocomputing 171 (2016) 1242–1253

Contents lists available at ScienceDirect

Neurocomputing

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

Self-representation based dual-graph regularized feature selection clustering

Ronghua Shang*, Zhu Zhang, Licheng Jiao, Chiyang Liu, Yangyang Li

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

ARTICLE INFO

Article history:
Received 21 March 2015
Received in revised form
21 July 2015
Accepted 22 July 2015
Communicated by: Jiayu Zhou
Available online 5 August 2015

Keywords:
Dual-graph
Self-representation
Feature selection
Clustering

ABSTRACT

Feature selection algorithms eliminate irrelevant and redundant features, even the noise, while preserving the most representative features. They can reduce the dimension of the dataset, extract essential features in high dimensional data and improve learning quality. Existing feature selection algorithms are all carried out in data space. However, the information of feature space cannot be fully exploited. To compensate for this drawback, this paper proposes a novel feature selection algorithm for clustering, named self-representation based dual-graph regularized feature selection clustering (DFSC). It adopts the self-representation property that data can be represented by itself. Meanwhile, the local geometrical information of both data space and feature space are preserved simultaneously. By imposing the $l_{2,1}$-norm constraint on the self-representation coefficients matrix in data space, DFSC can effectively select the most representative features for clustering. We give the objective function, develop iterative updating rules and provide the convergence proof. Two kinds of extensive experiments on some datasets demonstrate the effectiveness of DFSC. Extensive comparisons over several state-of-the-art feature selection algorithms illustrate that additionally considering the information of feature space based on self-representation property improves clustering quality. Meanwhile, because the additional feature selection process can select the most important features to preserve the intrinsic structure of dataset, the proposed algorithm achieves better clustering results compared with some co-clustering algorithms.

1. Introduction

In machine learning and data mining communities, high-dimensional data processing has emerged as a challenging problem. Examples of high-dimensional datasets include document data, user ratings data, gene expression data etc. [1,2]. Oftentimes, not all the features are important and discriminative, since correlation and redundancy exist between most of the features and sometimes some features are even noisy. Therefore, it is necessary and indispensable to use feature selection algorithms [1] to select an optional feature subset while retaining the salient characteristics of the original dataset as far as possible for compact data representation [2–4]. Feature selection algorithm has wide application, such as speech recognition [5], gene expression analysis [6], and disease diagnosis [7].

According to the way of utilizing label information [8], feature selection algorithms can be categorized as supervised algorithms [9], semi-supervised algorithms [10] and unsupervised algorithms [11]. Supervised approaches evaluate correlation between features using the label information, and discriminative information can be obtained from label information. Semi-supervised approaches use the labeled data as additional information to improve learning performance. However, the acquisition of label information needs an excessive cost in human label. Unsupervised feature selection determines the importance of features based on underlying properties of original dataset in the absence of label information [12]. In many practical applications, there is no label information available directly, which makes unsupervised feature selection quite demanding and challenging [13].

Traditional unsupervised feature selection approaches are prominent in many cases. However, there still exists some improvement as stated in the following [14].

1) Recent researches have shown that the observed data are found to lie on a low dimensional manifold embedded in a high dimensional space [15], while the manifold structure has not been fully taken into consideration.
2) Traditional unsupervised feature selection approaches focus only on data statistical character to rank the features, as in feature learning they often lack in learning mechanism, which is proved to be powerful and widely used in many fields [16,17].

* Corresponding author: Tel.: +86 02988202279.
E-mail address: rhshang@mail.xidian.edu.cn (R. Shang).

http://dx.doi.org/10.1016/j.neucom.2015.07.068
0925-2312/

3) Traditional unsupervised feature selection approaches only performed in data space, and the duality between data points and features is ignored.

As regard to learning mechanism, many clustering-based unsupervised feature selection algorithms [12,18,19] have been proposed. All these algorithms exploit either the manifold structure or discriminative structure of the dataset in data space to select the most representative features. However, the manifold structure of the feature space is ignored.

Some investigations have dedicated to leverage both the manifold structure and learning mechanism. Typical methods include: Laplacian score (LapScore) [20], spectral feature selection (SPEC) [21], multi-cluster feature selection (MCFS) [22], minimum redundancy spectral feature selection (MRSF) [23], joint embedding learning and sparse regression feature selection (JELSR)[14], and locality and similarity preserving embedding feature selection (LSPE) [24]. These methods construct graphs to characterize the manifold structure at first. LapScore and SPEC then calculate metrics based on which to rank all features. MCFS and MRSF add sparse constraints in multi-output regression, but both of them solve embedding learning and sparse regression in sequence. The difference is that MCFS uses l_1-norm as sparse regularization while MRSF uses $l_{2,1}-$norm instead. JELSR combines embedding learning and sparse regression, and applies the two steps jointly. LSPE unifies embedding learning and feature selection. These methods can be further improved in consideration of the aforementioned three factors.

Many unsupervised feature selection algorithms are used for clustering [24–29]. Clustering is the problem of dividing the data into several categories so that data points belonging to the same class have high similarity, while data points belonging to different classes have low degree of similarity [30–32]. For feature selection clustering methods, since the representative features obtained after selection are used for clustering, the clustering quality is enhanced.

On the other hand, in cluster analysis, matrix factorization based approaches have attracted considerable attention. Two typical matrix factorization methods widely applied in cluster analysis are nonnegative matrix factorization (NMF) [33] and concept factorization (CF) [34]. Based on NMF, Cai et al. proposed graph regularized nonnegative matrix factorization (GNMF) [35], GNMF can find a compact representation which uncovers the hidden semantics and simultaneously respects the intrinsic geometric structure. Based on CF, Cai et al. [36] proposed locally consistent concept factorization (LCCF) to extract the underlying concepts with respect to the intrinsic local geometric manifold structure. However, all the matrix factorization based approaches mentioned above performed in a single direction, i.e., in the row or column of the data matrix. The intrinsic information of the dataset cannot be fully discovered. Recent studies have found that not only the observed data are found to lie on a nonlinear low dimensional manifold, i.e., data manifold, but the features lie on a manifold, i.e., feature manifold [37]. Due to the consideration of the duality between data manifold and feature manifold, co-clustering approaches have shown to be superior to traditional one-sided clustering [15,37–40]. In [37], on the basis of CF, Ye et al. proposed dual-graph regularized concept factorization clustering (GCF). GCF considers the geometrical structures of both the data manifold and feature manifold for clustering to improve clustering accuracy. In [38], Dhillon et al. modeled a document collection as a bipartite graph using which a spectral algorithm is proposed for words and documents co-clustering. In [39], Dhillon et al. proposed a co-clustering algorithm which intertwines both the row and column clustering at all stages to increase the preserved mutual information monotonically. Shang et al. [15] improved GNMF by considering the geometrical information of both the data manifold and feature manifold simultaneously, and proposed graph dual regularization non-negative matrix factorization for co-clustering algorithm (DNMF). Ding et al. [40] proposed an orthogonal nonnegative matrix tri-factorization for clustering, which is used for words-documents co-clustering. All these co-clustering algorithms have achieved encouraging performance, which demonstrate that it is promising to consider the duality between data points and features.

Redundant features have properties of self-representation, i.e., each feature can be approximated by a linear combination of relevant features [41]. In real practice, the self-similarity is widespread. Any natural images involve high degree of self-similarity and redundancy. Similarity exists between different blocks of the same image, and the time series of climate monitoring may be very similar. Different sections of one coastline are also very alike. The self-similarity is used in a wide range of signal and image processing applications. In [42], the proposed joint image denoising algorithm uses self-similarity to construct similar patch groups. Self-similarity is also utilized to detect structural changes in time series [43]. Self-similarity property generally holds for most high-dimensional data and has been extensively used in machine learning and computer vision fields [41]. Just as sparsity leads to sparse representation, self-similarity results in self-representation [41].

Taking into account of manifold learning and feature selection, and inspired by the self-representation property and the idea of dual-regularization learning [44,45], we propose a novel feature selection algorithm for clustering, named self-representation based dual-graph regularized feature selection clustering (DFSC). This algorithm represents the data matrix and feature matrix simultaneously using self-representation property. In DFSC, two neighborhood graphs in data space and feature space are constructed respectively to encode the local geometrical information of both data space and feature space. We seek compact reconstruction of data matrix and feature matrix in data space and feature space respectively using self-representation property, and a sparse constraint is exerted on self-representation coefficients matrix in the data space, based on which to determine the importance of the features. We unify self-reconstruction, local manifold learning and sparse regression into a joint objective function and minimize this objective function with iterative and alternative updating optimization schemes.

DFSC differs from previous feature selection algorithms [18,19,24–28,31] in that it can preserve the local geometrical structure of the feature space. On the other hand, DFSC is related to co-clustering algorithms. Both of them preserve the information of data space and feature space. However, DFSC belongs to feature selection algorithms, which have a “selection” process, i.e., to select the most representative and effective features, and to eliminate redundant and relevant features. Our key contributions are highlighted as follows:

1. We adopt the property of self-representation in the proposed algorithm. The coefficients matrices in data space and feature space are used for local geometrical information preservation. And the underlying structure of the dataset can be detected effectively.
2. Compared with some co-clustering algorithms, DFSC can select a representative feature subset. It is more powerful for clustering.

The remaining of this paper is organized as follows. We introduce some related works in Section 2. In Section 3, we propose our framework and provide the convergence analysis of our optimization scheme. Extensive experiments are conducted in Section 4. In Section 5, we make some discussion about the efficiency of DFSC. Finally, we conclude our work with some possible improvement.

2. Related work

Before we go into the details of our algorithm, we briefly review some works that are closely related to this paper. We will introduce some feature selection methods that include LapScore, SPEC, MCFS, JELSR, MRSF and LSPE. And we also introduce some matrix factorization based clustering approaches including NMF, CF, DRCC, LCCF, GCF and DFSC.

We first introduce some notations. For matrix $\boldsymbol{B} \in \mathbb{R}^{s\times t}$, the $l_{r,p}-$ *norm* is defined as follows:

$$\|\boldsymbol{B}\|_{r,p} = \left(\sum_{i=1}^{s}\left(\sum_{j=1}^{t}|\boldsymbol{B}_{ij}|^{r}\right)^{r/p}\right)^{1/p} \tag{1}$$

When $r=p=1$, it is l_1-norm and we briefly denote it as $\|\cdot\|_1$. When $r=p=2$, it is l_2- *norm* and we briefly denote it as $\|\cdot\|_2$. When $r=2, p=1$, it is $l_{2,1}-$norm.

Given a dataset $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n]\in\mathbb{R}^{\mathbf{m}\times\mathbf{n}}$, where $\boldsymbol{x}_i=[\boldsymbol{x}_{i1},\ldots,\boldsymbol{x}_{im}]^T\in\mathbb{R}^{\boldsymbol{m}}$. $\boldsymbol{x}_i$ is the m-dimensional feature vector of the i-th data. n and m are the number of instances and features respectively. Feature selection aims at selecting a feature subset that optimizes certain criteria [46].

2.1. Feature selection methods

2.1.1. LapScore and SPEC

LapScore constructs the nearest neighborhood graph to model the local geometric structure of the dataset and chooses some features which have the largest Laplacian score. LapScore selects those features which are the smoothest on the graph.

SPEC can be regarded as an extension of LapScore. Both LapScore and SPEC select those features which can best reflect the underlying manifold structure.

While in LapScore and SPEC, the graph Laplacian is only used to characterize the data structure. They are lack of learning mechanism.

2.1.2. MCFS and MRSF

MCFS computes the low dimensional embedding $\boldsymbol{Y}$ at first, and then regresses each sample with l_1-norm regularization. MCFS can be regarded as solving the following problems in a two stage way:

$$\begin{aligned}&\arg\min_{\boldsymbol{Y}\boldsymbol{Y}^T=\boldsymbol{I}_d} Tr\left(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T\right)\\ &\operatorname{argmin}_{\boldsymbol{W}}\|\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{Y}\|_2^2+\alpha\|\boldsymbol{W}\|_1\end{aligned} \tag{2}$$

Similarly, MRSF can be regarded as solving the following two problems in a two stage way:

$$\begin{aligned}&\arg\min_{\boldsymbol{Y}\boldsymbol{Y}^T=\boldsymbol{I}_d} Tr\left(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T\right)\\ &\operatorname{argmin}_{\boldsymbol{W}}\|\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{Y}\|_2^2+\alpha\|\boldsymbol{W}\|_{2,1}\end{aligned} \tag{3}$$

where $\boldsymbol{Y}\in\mathbb{R}^{d\times n}$ is the low dimensional embedding, $\boldsymbol{L}$ is the graph Laplacian, $\boldsymbol{W}\in\mathbb{R}^{m\times d}$ is the transformation matrix, d is the dimensionality of embedding, and $\alpha\geq 0$ is the regularization parameter.

From the framework of MCFS and MRSF, though they apply different sparse constraints, both of them compute the low dimensional embedding and then rank the features based on regression coefficients. Since they separate embedding learning and sparse regression, the performance is degraded. Therefore, we expect to solve embedding learning and sparse regression jointly.

2.1.3. JELSR and LSPE

The framework of JELSR is

$$\arg\min_{\boldsymbol{W},\boldsymbol{Y}\boldsymbol{Y}^T=\boldsymbol{I}_d} Tr\left(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T\right)+\beta\left(\|\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{Y}\|_2^2+\alpha\|\boldsymbol{W}\|_{2,1}\right) \tag{4}$$

where $\beta,\alpha\geq 0$ are two regularization parameters.

LSPE solves the following problem:

$$\min_{A,Q}\|\boldsymbol{A}^T(\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Q})\|^2+\beta\boldsymbol{Tr}(\boldsymbol{Q}\boldsymbol{L}\boldsymbol{Q}^T)+\alpha\|\boldsymbol{A}\|_{2,1} \tag{5}$$

where $\boldsymbol{A}$ is a projection matrix whose row vectors act as measurement for the importance of features, and $\alpha\geq 0$ is the regularization parameter.

Comparing the formulations in (4) and (5), we know that JELSR selects the features which can best preserve the locality, and that LSPE preserves the locality and similarity of data space simultaneously to find the optimal feature subset. However, none of them considers the structure of feature space.

2.2. Co-clustering methods

2.2.1. NMF

NMF seeks to factorize $\boldsymbol{X}$ into the product of two low rank nonnegative matrices which are basis matrix $\boldsymbol{U}\in\mathbb{R}^{m\times k}$ and coefficient matrix $\boldsymbol{V}\in\mathbb{R}^{n\times k}$, where $k<<\min(m,n)$. The objective function of NMF can be concluded as follows:

$$\begin{aligned}&\min_{\boldsymbol{U},\boldsymbol{V}}\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\|_{\boldsymbol{F}}^2\\ &s.t.\qquad \boldsymbol{U},\boldsymbol{V}\geq 0\end{aligned} \tag{6}$$

where $\|\cdot\|_F$ denotes Frobenius norm (F-norm).

2.2.2. DRCC

DRCC is based on semi-nonnegative matrix tri-factorization, which factorizes the data matrix into three matrices. It also preserves the geometrical manifold structures of both data graph and feature graph. It solves the problem as follows:

$$\min_{\boldsymbol{U},\boldsymbol{F},\boldsymbol{V}}\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{F}\boldsymbol{V}^T\|_F^2+\lambda\mathrm{Tr}(\boldsymbol{V}^T\boldsymbol{L}_{\boldsymbol{V}}\boldsymbol{V})+\mu\mathrm{Tr}(\boldsymbol{U}^T\boldsymbol{L}_{\boldsymbol{U}}\boldsymbol{U})\text{s.t.}\qquad \boldsymbol{U},\boldsymbol{V}\geq 0 \tag{7}$$

where $\lambda,\mu\geq 0$ are two regularization parameters, and $\boldsymbol{F}$ is a matrix whose entries can take any sign.$\boldsymbol{L}_{\boldsymbol{V}}=\boldsymbol{D}^{\boldsymbol{V}}-\boldsymbol{W}^{\boldsymbol{V}}$ and $\boldsymbol{L}_{\boldsymbol{U}}=\boldsymbol{D}^{\boldsymbol{U}}-\boldsymbol{W}^{\boldsymbol{U}}$ are the graph Laplacian of data graph and feature graph respectively. $\boldsymbol{L}_{\boldsymbol{V}}$ and $\boldsymbol{L}_{\boldsymbol{U}}$ reflect the label smoothness of data points and features respectively.

2.2.3. CF

CF differs from NMF in that it can be applied to data containing negative values and it can adopt the idea of the kernel method [34]. CF solves the following problem:

$$\begin{aligned}&\min_{\boldsymbol{U},\boldsymbol{V}}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{V}^T\|_F^2\\ &s.t.\qquad \boldsymbol{W},\boldsymbol{V}\geq 0\end{aligned} \tag{8}$$

where $\boldsymbol{W}\in\mathbb{R}^{n\times k}$ is the association matrix, and $\boldsymbol{V}\in\mathbb{R}^{n\times k}$ is the projection matrix. Cluster labels can be derived from $\boldsymbol{V}$.

2.2.4. LCCF

Compared with CF, LCCF aims to preserve the intrinsic local manifold geometry structure of the dataset, and the objective function of LCCF is as follows:

$$\begin{aligned}&\min_{\boldsymbol{U},\boldsymbol{V}}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{V}^T\|_{\boldsymbol{F}}^2+\lambda\mathrm{Tr}\left(\boldsymbol{V}^T\boldsymbol{L}\boldsymbol{V}\right)\\ &s.t.\qquad \boldsymbol{W},\boldsymbol{V}\geq 0\end{aligned} \tag{9}$$

where $\boldsymbol{W}\in\mathbb{R}^{n\times k}$ is the association matrix, $\boldsymbol{V}\in\mathbb{R}^{n\times k}$ is the

projection matrix, $\lambda \geq 0$ is the regularization parameter, and $\boldsymbol{L}$ is the graph Laplacian matrix.

2.2.5. GCF

GCF simultaneously considers the geometrical information of data manifold and feature manifold. The objective of GCF is

$$\min_{\boldsymbol{W},\boldsymbol{V}} \|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{V}^T\|^2+\lambda \mathrm{Tr}(\boldsymbol{V}^T\boldsymbol{L}_{\boldsymbol{V}}\boldsymbol{V})+\mu \mathrm{Tr}(\boldsymbol{W}^T\boldsymbol{L}_{\boldsymbol{W}}\boldsymbol{W})$$
$$s.t. \quad \boldsymbol{W},\boldsymbol{V} \geq 0 \tag{10}$$

where $\lambda, \mu \geq 0$ are two regularization parameters, $\boldsymbol{L}_{\boldsymbol{V}}$ is the Laplacian matrix of data graph, $\boldsymbol{L}_{\boldsymbol{W}}=\boldsymbol{X}^T\boldsymbol{L}_{\boldsymbol{U}}\boldsymbol{X}$, and $\boldsymbol{L}_{\boldsymbol{U}}$ is the Laplacian matrix of feature graph.

3. The proposed algorithm

3.1. Objective function

Data points and features are represented by themselves by exploiting self-representation. For each vector $\boldsymbol{x}_i$, by self-representation property we know that $\boldsymbol{x}_i$ can be represented by all the features in $\boldsymbol{X}$, i.e., $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$. We have

$$\boldsymbol{x}_i = \sum_k^n \boldsymbol{x}_k \boldsymbol{S}_{ki} + \boldsymbol{f}_i \tag{11}$$

where $\boldsymbol{S}=[\boldsymbol{S}_{ki}] \in \mathbb{R}^{n\times n}$ is the self-representation coefficients matrix in feature space, and $\boldsymbol{f}_i$ is the residual error term. Formula (11) can be rewritten in matrix form as follows:

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{S}+\boldsymbol{F} \tag{12}$$

$\boldsymbol{F}$ is the corresponding error matrix. Similarly, we have the following formula in data space:

$$\boldsymbol{X}^{\boldsymbol{T}} = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{P}+\boldsymbol{H} \tag{13}$$

where $\boldsymbol{P}\in\mathbb{R}^{m\times m}$ is the self-representation coefficients matrix in data space, and $\boldsymbol{H}$ is the corresponding error matrix. Since $\boldsymbol{P}$ and $\boldsymbol{S}$ reflect the contribution of features and data points in the process of self-representation, we restrict $\boldsymbol{P}$ and $\boldsymbol{S}$ to be non-negative, i.e., $\boldsymbol{P},\boldsymbol{S} \geq 0$.

We minimize the self-representation reconstruction error, and solve the following problem:

$$\min \|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{S}\|_F^2+\beta\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{P}\|_F^2 \tag{14}$$

where the parameter $\beta > 0$ balances these two self-representation error terms.

To detect the underlying geometrical structure, many manifold learning algorithms have been proposed [15], such as locally linear embedding (LLE) [47], ISOMAP [48] and Laplacian Eigenmap [49]. These methods adopt the locally invariant idea [50], namely the nearby points are likely to have similar data representation. It has been proven that preserving the geometrical structure of the data can improve learning quality significantly.

Now, two neighborhood graphs in data space and feature space are constructed to preserve the local geometrical information, i.e., data graph and feature graph.

In data space we construct the nearest neighborhood graph G. Each node of the graph corresponds to a data point. An edge is set up if two data points are in the k nearest neighborhood. The similarities between the two data points act as the edge weight. We choose Gaussian kernel function or 0–1 weighting scheme as weight function. The Gaussian kernel function is defined as follows:

$$\left[\boldsymbol{W}^P\right]_{ij} = \begin{cases} exp\left(-\|\boldsymbol{x}_{:,i}-\boldsymbol{x}_{:,j}\|^2/2\sigma\right), & if\ \boldsymbol{x}_{:,i}\in N(\boldsymbol{x}_{:,j})\ \text{ or } \boldsymbol{x}_{:,j}\in N(\boldsymbol{x}_{:,i}), \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where $\boldsymbol{x}_{:,j}$ denotes the j-th column of the matrix $\boldsymbol{X}$, $N(\boldsymbol{x}_{:,j})$ denotes the k nearest neighborhood set for $\boldsymbol{x}_{:,j}$, and σ is the bandwidth parameter.

The 0–1 weighting scheme is defined as follows:

$$\left[\boldsymbol{W}^P\right]_{ij} = \begin{cases} 1, & if\ \boldsymbol{x}_{:,i}\in N(\boldsymbol{x}_{:,j})\ \text{ or } \boldsymbol{x}_{:,j}\in N(\boldsymbol{x}_{:,i}), \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

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

Similarly, we construct feature graph in feature space, and the nodes correspond to the feature set $\{\boldsymbol{X}_{1,:}^T, \ldots, \boldsymbol{X}_{m,:}^T\}$. Gaussian kernel function has the following definition:

$$\left[\boldsymbol{W}^S\right]_{ij} = \begin{cases} \exp\left(-\|\boldsymbol{x}_{i,:}-\boldsymbol{x}_{j,:}\|^2/2\sigma\right), & \textit{if } \boldsymbol{x}_{i,:}\in N(\boldsymbol{x}_{j,:})\ \text{ or } \boldsymbol{x}_{j,:}\in N(\boldsymbol{x}_{i,:}), \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where $\boldsymbol{x}_{i,:}$ denotes the i-th row of the matrix $\boldsymbol{X}$, and $N(\boldsymbol{x}_{j,:})$ denotes the k-nearest neighborhood set for feature $\boldsymbol{x}_{j,:}$.

The 0–1 weighting is defined as follows:

$$\left[\boldsymbol{W}^S\right]_{ij} = \begin{cases} 1, & if\ \boldsymbol{x}_{i,:}\in N(\boldsymbol{x}_{j,:})\ \text{ or } \boldsymbol{x}_{j,:}\in N(\boldsymbol{x}_{i,:}), \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

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

We know that $\boldsymbol{W}_{ij}^P$ means the similarity between data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and a large value of $\boldsymbol{W}_{ij}^P$ means that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have high degree of similarity. From Eq. (8), we have

$$\begin{aligned} \boldsymbol{x}_i &= \boldsymbol{x}_1\boldsymbol{s}_{1i}+\boldsymbol{x}_2\boldsymbol{s}_{2i}+\ldots+\boldsymbol{x}_n s_{ni} \\ \boldsymbol{x}_j &= \boldsymbol{x}_1\boldsymbol{s}_{1j}+\boldsymbol{x}_2\boldsymbol{s}_{2j}+\ldots+\boldsymbol{x}_n\boldsymbol{s}_{nj} \end{aligned} \tag{19}$$

We denote the i-th and j-th column of $\boldsymbol{S}$ as

$$\begin{aligned} \boldsymbol{S}_i &= [\boldsymbol{s}_{1i}, \boldsymbol{s}_{2i}, \ldots, \boldsymbol{s}_{ni}]^T \\ \boldsymbol{S}_j &= [\boldsymbol{s}_{1j}, \boldsymbol{s}_{2j}, \ldots, \boldsymbol{s}_{nj}]^T \end{aligned} \tag{20}$$

If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have high degree of similarity, driven by the idea that nearby points are likely to have similar data representation, we draw a conclusion that a large value of W_{ij}^P means that $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ are close. Thus we have the representation smoothness as follows:

$$\begin{aligned} &\tfrac{1}{2}\sum_{\boldsymbol{i}=1}^{n}\sum_{\boldsymbol{j}=1}^{n}\|\boldsymbol{s}_i-\boldsymbol{s}_j\|^2\boldsymbol{W}_{ij}^P \\ &= \sum_{\boldsymbol{i}=1}^{n}\boldsymbol{s}_i{}^T\boldsymbol{s}_i\boldsymbol{D}_{ii}^P-\sum_{\boldsymbol{i}=1}^{n}\sum_{\boldsymbol{j}=1}^{n}\boldsymbol{s}_i{}^T\boldsymbol{s}_i\boldsymbol{W}_{ij}^P \\ &= Tr(\boldsymbol{S}\boldsymbol{D}^P\boldsymbol{S}^T)-Tr(\boldsymbol{S}\boldsymbol{W}^P\boldsymbol{S}^T) \\ &= Tr(\boldsymbol{S}\boldsymbol{L}^P\boldsymbol{S}^T) \end{aligned} \tag{21}$$

Similarly, considering the self-representation matrix $\boldsymbol{P}$ in data space and the similarity matrix $\boldsymbol{W}^S$, we have the following data representation smoothness:

$$\begin{aligned} &\tfrac{1}{2}\sum_{\boldsymbol{i}=1}^{m}\sum_{\boldsymbol{j}=1}^{m}\|\boldsymbol{p}_i-\boldsymbol{p}_j\|^2\boldsymbol{W}_{ij}^S \\ &= \sum_{\boldsymbol{i}=1}^{m}\boldsymbol{p}_i^T\boldsymbol{p}_i\boldsymbol{D}_{ii}^S-\sum_{\boldsymbol{i}=1}^{m}\sum_{\boldsymbol{j}=1}^{m}\boldsymbol{p}_i^T\boldsymbol{p}_i\boldsymbol{W}_{ij}^S \\ &= Tr(\boldsymbol{P}\boldsymbol{D}^S\boldsymbol{P}^T)-Tr(\boldsymbol{P}\boldsymbol{W}^S\boldsymbol{P}^T) \\ &= Tr(\boldsymbol{P}\boldsymbol{L}^S\boldsymbol{P}^T) \end{aligned} \tag{22}$$

Based on the above data graph and feature graph, exploiting the self-representation property, and considering the manifold information of data space and feature space simultaneously, we seek to get a compact representation in both data space and

feature space. DFSC solves the following minimization problem:

$$\min \|\boldsymbol{X}-\boldsymbol{XS}\|_F^2+\beta\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{P}\|_F^2+\alpha_1 Tr(\boldsymbol{SL}^P\boldsymbol{S}^T)+\alpha_2 Tr(\boldsymbol{PL}^S\boldsymbol{P}^T),$$
$$s.t.\quad \boldsymbol{S}\geq 0,\boldsymbol{P}\geq 0 \tag{23}$$

where the parameters $\beta>0$, $\alpha_1>0,\alpha_2>0$. For simplicity and easy adjustment, we let $\alpha_1=\alpha_2=\alpha$, and the objective function can be rewritten as follows:

$$\min \|\boldsymbol{X}-\boldsymbol{XS}\|_F^2+\beta\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{P}\|_F^2+\alpha\left(Tr(\boldsymbol{SL}^P\boldsymbol{S}^T)+Tr(\boldsymbol{PL}^S\boldsymbol{P}^T)\right),$$
$$s.t.\quad \boldsymbol{S}\geq 0,\boldsymbol{P}\geq 0 \tag{24}$$

Let $\boldsymbol{P}=[\boldsymbol{P}_{1.};\ldots;\boldsymbol{P}_{i.};\ldots;\boldsymbol{P}_{m.}]$, $\boldsymbol{P}_{i.}$ is the i-th row of the matrix $\boldsymbol{P}$. $\|\boldsymbol{P}_{i.}\|_2$ stands for the contribution of the i-th feature in the process of self-representation. Therefore $\|\boldsymbol{P}_{i.}\|_2$ can be used as feature weights to rank features. To avoid the trivial solution $\boldsymbol{P}=\boldsymbol{I}_m$ ($\boldsymbol{I}_m$ is an m-dimensional identity matrix) and to ensure sparsity, we exert $l_{2,1}-norm$ on matrix $\boldsymbol{P}$. The $l_{2,1}-norm$ constraint ensures that matrix $\boldsymbol{P}$ is row-sparse, so $\|\boldsymbol{P}_{i.}\|_2$ reflects the importance of the i-th feature in the whole feature. According to $\|\boldsymbol{P}_{i.}\|_2$, we select the most important features. Thus, our problem is

$$\min_{\boldsymbol{P},\boldsymbol{S}}\|\boldsymbol{X}-\boldsymbol{XS}\|_F^2+\beta\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{P}\|_F^2+\alpha(Tr(\boldsymbol{SL}^P\boldsymbol{S}^T)+Tr(\boldsymbol{PL}^S\boldsymbol{P}^T))+\lambda\|\boldsymbol{P}\|_{2,1},$$
$$s.t.\quad \boldsymbol{S}\geq 0,\boldsymbol{P}\geq 0 \tag{25}$$

The parameter $\lambda>0$ balances the last sparse item with other items.

3.2. Iterative updating schemes for solving DFSC

For the problem in (25), it is difficult to obtain a closed-form solution. Therefore, we propose an iterative and alternative optimization scheme. Formula (25) can be rewritten as follows:

$$\begin{aligned}L(\boldsymbol{P},\boldsymbol{S})&=\|\boldsymbol{X}-\boldsymbol{XS}\|_F^2+\beta\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{P}\|_F^2+\alpha(Tr(\boldsymbol{SL}^P\boldsymbol{S}^T)+Tr(\boldsymbol{PL}^S\boldsymbol{P}^T))+\lambda\|\boldsymbol{P}\|_{2,1}\\&=Tr(\boldsymbol{XX}^T-2\boldsymbol{XSX}^T+\boldsymbol{XSS}^T\boldsymbol{X}^T)+\beta Tr(\boldsymbol{X}^T\boldsymbol{X}-2\boldsymbol{X}^T\boldsymbol{PX}+\boldsymbol{X}^T\boldsymbol{PP}^T\boldsymbol{X})\\&\quad+\alpha[Tr(\boldsymbol{SL}^P\boldsymbol{S}^T)+Tr(\boldsymbol{PL}^S\boldsymbol{P}^T)]+\lambda\|\boldsymbol{P}\|_{2,1}\end{aligned} \tag{26}$$

Let $\boldsymbol{\psi}_{ij}$ and $\boldsymbol{\phi}_{kj}$ be the corresponding Lagrange multiplier for constraint $\boldsymbol{P}_{ij}\geq 0$ and $\boldsymbol{S}_{kj}\geq 0$, respectively. Then we have the following Lagrange function:

$$\begin{aligned}L_1&=Tr(\boldsymbol{XX}^T-\boldsymbol{2XSX}^T+\boldsymbol{XSS}^T\boldsymbol{X}^T)+\beta\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{X}-2\boldsymbol{X}^T\boldsymbol{PX}+\boldsymbol{X}^T\boldsymbol{PP}^T\boldsymbol{X})\\&\quad+a[Tr(\boldsymbol{SL}^P\boldsymbol{S}^T)+Tr(\boldsymbol{PL}^S\boldsymbol{P}^T)]+\lambda\|\boldsymbol{P}\|_{2,1}+Tr(\boldsymbol{\psi P}^T)+Tr(\boldsymbol{\phi S}^T)\end{aligned} \tag{27}$$

The partial derivative of L_1 with respect to $\boldsymbol{S}$ is

$$\frac{\partial L_1}{\partial \boldsymbol{S}}=-2\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{2X}^T\boldsymbol{XS}+2\alpha\boldsymbol{SL}^P+\boldsymbol{\phi} \tag{28}$$

Using the KKT conditions, $\boldsymbol{\phi}_{kj}\boldsymbol{S}_{kj}=0$, we have $(-\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{X}^T\boldsymbol{XS}+\alpha\boldsymbol{SL}^P)\boldsymbol{S}=0$. Since $\boldsymbol{L}^P=\boldsymbol{D}^P-\boldsymbol{W}^P$, then $[-\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{X}^T\boldsymbol{XS}+\alpha\boldsymbol{S}(\boldsymbol{D}^P-\boldsymbol{W}^P)]\boldsymbol{S}=0$, we get the following updating formula:

$$\boldsymbol{S}=\boldsymbol{S}\frac{\boldsymbol{X}^T\boldsymbol{X}+\alpha\boldsymbol{SW}^P}{\boldsymbol{X}^T\boldsymbol{XS}+\alpha\boldsymbol{SD}^P} \tag{29}$$

Similarly, for updating rule for $\boldsymbol{P}$, we first introduce an auxiliary function, then (27) can be rewritten as follows:

$$\begin{aligned}\boldsymbol{L}_1&=Tr(\boldsymbol{XX}^T-2\boldsymbol{XSX}^T+\mathrm{XSS}^T\boldsymbol{X}^T)+\beta\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{X}-2\boldsymbol{X}^T\boldsymbol{PX}+\boldsymbol{X}^T\boldsymbol{PP}^T\boldsymbol{X})\\&\quad+\alpha[Tr(\boldsymbol{SL}^P\boldsymbol{S}^T)+Tr(\boldsymbol{PL}^S\boldsymbol{P}^T)]+\lambda\mathrm{Tr}(\boldsymbol{P}^T\boldsymbol{UP})+Tr(\boldsymbol{\psi P}^T)+Tr(\boldsymbol{\phi S}^T)\end{aligned} \tag{30}$$

where $\boldsymbol{U}\in\mathbb{R}^{m\times m}$ is a diagonal matrix and the i-th diagonal element of which is given as follows:

$$\boldsymbol{U}_{ii}=\frac{1}{2\|\boldsymbol{P}_{i.}\|_2} \tag{31}$$

Taking the partial derivative of L_1 with respect to $\boldsymbol{P}$, we arrive at

$$\frac{\partial \boldsymbol{L}_1}{\partial \boldsymbol{P}}=2\alpha\boldsymbol{PL}^S-2\beta\boldsymbol{XX}^T+2\beta\boldsymbol{XX}^T\boldsymbol{P}+2\lambda\boldsymbol{UP}+\boldsymbol{\psi} \tag{32}$$

Using the KKT conditions $\boldsymbol{\psi}_{ij}\boldsymbol{P}_{ij}=0$, we have $(2\alpha\boldsymbol{PL}^S-2\beta\boldsymbol{XX}^T+2\beta\boldsymbol{XX}^T\boldsymbol{P}+2\lambda\boldsymbol{UP})\boldsymbol{P}=0$, since $\boldsymbol{L}^S=\boldsymbol{D}^S-\boldsymbol{W}^S$, $[\alpha\boldsymbol{P}(\boldsymbol{D}^S-\boldsymbol{W}^S)-\beta\boldsymbol{XX}^S+\beta\boldsymbol{XX}^S\boldsymbol{P}+\lambda\boldsymbol{UP}]\boldsymbol{P}=0$, we get the following updating formula:

$$\boldsymbol{P}=\boldsymbol{P}\frac{\beta\boldsymbol{XX}^T+\alpha\boldsymbol{PW}^S}{\alpha\boldsymbol{PD}^S+\beta\boldsymbol{XX}^T\boldsymbol{P}+\lambda\boldsymbol{UP}} \tag{33}$$

To avoid overflow, we introduce a sufficiently small constant ε in the definition of the matrix $\boldsymbol{U}$.

$$\boldsymbol{U}_{ii}=\frac{1}{2\max(\|\boldsymbol{P}_{i.}\|_2,\varepsilon)} \tag{34}$$

Table 1 shows the process of DFSC.

3.3. Convergence analysis

In this section, we will investigate the convergence of the proposed algorithm. We prove that the objective function (25) is monotonically decreasing under the updating rules (29) and (33).

We start from the convergence analysis of Eq. (29).

Definition 1. If the following conditions

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

and

$$G(u,u)=F(u) \tag{36}$$

are satisfied, $G(u,u')$ is an auxiliary function for $F(u)$.Then F is non-increasing under the following updating formula:

$$u^{(t+1)}=\arg\min_u G(u,u^{(t)}) \tag{37}$$

Proof. $F(u^{(t+1)})\leq G(u^{(t+1)},u^{(t)})\leq G(u^{(t)},u^{(t)})=F(u^{(t)})$. Let

$$F(\boldsymbol{S})=Tr(-2\boldsymbol{XSX}^T+\boldsymbol{XSS}^T\boldsymbol{X}^T)+\alpha\mathrm{Tr}(\boldsymbol{SL}^P\boldsymbol{S}^T) \tag{38}$$

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

$$F'_{ij}=\left[\frac{\partial F}{\partial \boldsymbol{S}}\right]_{ij}=\left[-2\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{2X}^T\boldsymbol{XS}+2\alpha\boldsymbol{SL}^P\right]_{\boldsymbol{ij}}\text{ and }F''_{ij}=2\alpha\left[\boldsymbol{L}^P\right]_{jj}+2\left[\boldsymbol{X}^T\boldsymbol{X}\right]_{ii} \tag{39}$$

Lemma 1. *The following function:*

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

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

Proof. The Taylor expansion of $\boldsymbol{F}_{ij}(\boldsymbol{S}_{ij})$ is

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

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

$$\frac{\left[\boldsymbol{X}^T\boldsymbol{XS}+\alpha\boldsymbol{SD}^P\right]_{ij}}{\boldsymbol{S}_{ij}^{(t)}}\geq\alpha\left[\boldsymbol{L}^S\right]_{jj}+\left[\boldsymbol{X}^T\boldsymbol{X}\right]_{ii} \tag{42}$$

Since $\left[\boldsymbol{X}^T\boldsymbol{XS}\right]_{ij}=\sum_{l=1}^{n}\left[\boldsymbol{X}^T\boldsymbol{X}\right]_{il}\boldsymbol{S}_{ij}^{(t)}\geq\left[\boldsymbol{X}^T\boldsymbol{X}\right]_{ii}\boldsymbol{S}_{ij}^{(t)}$ and $\alpha\left[\boldsymbol{SD}^P\right]_{ij}=\alpha\sum_{l=1}^{n}\boldsymbol{S}_{il}^{(t)}\left[\boldsymbol{D}^P\right]_{lj}\geq\alpha\boldsymbol{S}_{ij}^{(t)}\left[\boldsymbol{D}^P\right]_{jj}\geq\alpha\boldsymbol{S}_{ij}^{(t)}\left[\boldsymbol{D}^P-\boldsymbol{W}^P\right]_{jj}=\alpha\boldsymbol{S}_{ij}^{(t)}\left[\boldsymbol{L}^P\right]_{jj}$.

Table 1
DFSC algorithm.

Input: data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, $\boldsymbol{W}^P, \boldsymbol{W}^S$, α, β, λ, the maximum iteration number *maxiter*, the number of selected features q, the number of clusters c.
Output: Self-representation matrices $\boldsymbol{P}$ and $\boldsymbol{S}$, clustering *label*.
1. Initialize matrix $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{P}$ as identity matrices, $\boldsymbol{U}=\boldsymbol{I_m}$, $\boldsymbol{S}=\boldsymbol{I_n}$, $\boldsymbol{P}=\boldsymbol{I_m}$.
2. Updating the $\boldsymbol{S}, \boldsymbol{P}, \boldsymbol{U}$ according to the iterative updating rules (29), (33) and (34), until the convergence conditions are satisfied.
3. Ranking all the features in descending order according to $\|\boldsymbol{P}_{i.}\|_2$, select the top p features.
4. Clustering the selected features using *K-means* algorithm.

therefore, (42) holds and $G(\boldsymbol{S}_{ij}, \boldsymbol{S}_{ij}^{(t)}) \geq F_{ij}(\boldsymbol{S}_{ij})$, we have $G(\boldsymbol{S}_{ij}, \boldsymbol{S}_{ij}) = F_{ij}(\boldsymbol{S}_{ij})$.

Next, we will make use of the auxiliary function to show that F_{ij} decreases monotonically under the updating rules in Eq. (33).

Proof. Substituting $G(\boldsymbol{S}_{ij}, \boldsymbol{S}_{ij}^{(t)})$ in (37) into (40), we can get

$$\boldsymbol{S}_{ij}^{(t+1)} = \boldsymbol{S}_{ij}^{(t)} - \boldsymbol{S}_{ij}^{(t)} \frac{F'_{ij}(\boldsymbol{S}_{ij}^{(t)})}{2\left[\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{S} + \alpha\boldsymbol{S}\boldsymbol{D}^P\right]_{ij}} = \boldsymbol{S}_{ij}^{(t)} \frac{\left[\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{S}\boldsymbol{W}^P\right]_{ij}}{\left[\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{S} + \alpha\boldsymbol{S}\boldsymbol{D}^P\right]_{ij}} \tag{43}$$

Since (41) is an auxiliary function for F_{ij}, F_{ij} is non-increasing under the updating rule stated in Eq. (29).

For the convergence proof of updating rules in Eq. (33) for $\boldsymbol{P}$, we adopt the similar process as in [27].

Lemma 2. *For any nonzero vector* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$,

$$\|\boldsymbol{x}\|_2 - \frac{\|\boldsymbol{x}\|_2^2}{2\|\boldsymbol{y}\|_2} \leq \|\boldsymbol{y}\|_2 - \frac{\|\boldsymbol{y}\|_2^2}{2\|\boldsymbol{x}\|_2} \tag{44}$$

See detailed proof of Lemma 2 in reference [24].

We now give the proof of the convergence.

Proof. In the i-th iteration, we fix $\boldsymbol{U}$ as $\boldsymbol{U}^t$, compute $\boldsymbol{S}^{t+1}$ and $\boldsymbol{P}^{t+1}$, and we have the following inequality:

$$\begin{aligned}
&Tr(-2\boldsymbol{X}\boldsymbol{S}^{t+1}\boldsymbol{X}^T + \boldsymbol{X}\boldsymbol{S}^{t+1}(\boldsymbol{S}^{t+1})^T\boldsymbol{X}^T) + \alpha\left[Tr(\boldsymbol{S}^{t+1}\boldsymbol{L}^P(\boldsymbol{S}^{t+1})^T) + Tr(\boldsymbol{P}^{t+1}\boldsymbol{L}^S(\boldsymbol{P}^{t+1})^T)\right] \\
&\quad + \beta Tr(-2\boldsymbol{X}^T\boldsymbol{P}^{t+1}X + \boldsymbol{X}^TP^{t+1}(\boldsymbol{P}^{t+1})^T\boldsymbol{X}) + \lambda Tr((\boldsymbol{P}^{t+1})^T\boldsymbol{U}^t\boldsymbol{P}^{t+1}) \\
&\leq Tr(-2\boldsymbol{X}\boldsymbol{S}^t\boldsymbol{X}^T + \boldsymbol{X}\boldsymbol{S}^t(\boldsymbol{S}^t)^T\boldsymbol{X}^T) + \alpha\left[Tr(\boldsymbol{S}^t\boldsymbol{L}^P(\boldsymbol{S}^t)^T) + Tr(\boldsymbol{P}^t\boldsymbol{L}^S(\boldsymbol{P}^t)^T)\right] \\
&\quad + \beta Tr(-2\boldsymbol{X}^T\boldsymbol{P}^t\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{P}^t(\boldsymbol{P}^t)^T\boldsymbol{X}) + \lambda Tr((\boldsymbol{P}^t)^T\boldsymbol{U}^t\boldsymbol{P}^t)
\end{aligned} \tag{45}$$

Since $\|\boldsymbol{P}\|_{2,1} = \sum_{i=1}^m \|\boldsymbol{P}_{i.}\|_2$, the above inequality indicates

$$\begin{aligned}
&Tr(-2\boldsymbol{X}\boldsymbol{S}^{t+1}\boldsymbol{X}^T + \boldsymbol{X}\boldsymbol{S}^{t+1}(\boldsymbol{S}^{t+1})^T\boldsymbol{X}^T) + \alpha\left[Tr(\boldsymbol{S}^{t+1}\boldsymbol{L}^P(\boldsymbol{S}^{t+1})^T) + Tr(\boldsymbol{P}^{t+1}\boldsymbol{L}^S(\boldsymbol{P}^{t+1})^T)\right] \\
&+ \beta Tr(-2\boldsymbol{X}^T\boldsymbol{P}^{t+1}\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{P}^{t+1}(\boldsymbol{P}^{t+1})^T\boldsymbol{X}) + \lambda\|\boldsymbol{P}^{t+1}\|_{2,1} + \lambda\sum_{i=1}^m\left(\frac{\|\boldsymbol{P}_{i.}^{t+1}\|_2^2}{2\|\boldsymbol{P}_{i.}^t\|_2} - \|\boldsymbol{P}_{i.}^{t+1}{}_2\|_2\right) \\
&\leq Tr(-2\boldsymbol{X}\boldsymbol{S}^t\boldsymbol{X}^T + \boldsymbol{X}\boldsymbol{S}^t(\boldsymbol{S}^t)^T\boldsymbol{X}^T) + \alpha\left[Tr(\boldsymbol{S}^t\boldsymbol{L}^P(\boldsymbol{S}^t)^T) + Tr(\boldsymbol{P}^t\boldsymbol{L}^S(\boldsymbol{P}^t)^T)\right] \\
&\quad + \beta Tr(-2\boldsymbol{X}^T\boldsymbol{P}^t\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{P}^t(\boldsymbol{P}^t)^T\boldsymbol{X}) + \lambda\|\boldsymbol{P}^t\|_{2,1} + \lambda\sum_{i=1}^m\left(\frac{\|\boldsymbol{P}_{i.}^t\|_2^2}{2\|\boldsymbol{P}_{i.}^t\|_2} - \|\boldsymbol{P}_{i.}^t{}_2\|_2\right)
\end{aligned} \tag{46}$$

According to Lemma 2, we have

$$\frac{\|\boldsymbol{P}_{i.}^{t+1}\|_2^2}{2\|\boldsymbol{P}_{i.}^t\|_2} - \|\boldsymbol{P}_{i.}^{t+1}{}_2\|_2 \geq \frac{\|\boldsymbol{P}_{i.}^t\|_2^2}{2\|\boldsymbol{P}_{i.}^t\|_2} - \|\boldsymbol{P}_{i.}^t{}_2\|_2 \tag{47}$$

From (47) and (48), we have

$$\begin{aligned}
&Tr(-2\boldsymbol{X}\boldsymbol{S}^{t+1}\boldsymbol{X}^T + \boldsymbol{X}\boldsymbol{S}^{t+1}(\boldsymbol{S}^{t+1})^T\boldsymbol{X}^T) + \alpha\left[Tr(\boldsymbol{S}^{t+1}\boldsymbol{L}^P(\boldsymbol{S}^{t+1})^T) + Tr(\boldsymbol{P}^{t+1}\boldsymbol{L}^S(\boldsymbol{P}^{t+1})^T)\right] \\
&\quad + \beta Tr(-2\boldsymbol{X}^T\boldsymbol{P}^{t+1}\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{P}^{t+1}(\boldsymbol{P}^{t+1})^T\boldsymbol{X}) + \lambda\|\boldsymbol{P}^{t+1}\|_{2,1} \\
&\leq Tr(-2\boldsymbol{X}\boldsymbol{S}^t\boldsymbol{X}^T + \boldsymbol{X}\boldsymbol{S}^t(\boldsymbol{S}^t)^T\boldsymbol{X}^T) + \alpha\left[Tr(\boldsymbol{S}^t\boldsymbol{L}^P(\boldsymbol{S}^t)^T) + Tr(\boldsymbol{P}^t\boldsymbol{L}^S(\boldsymbol{P}^t)^T)\right] \\
&\quad + \beta Tr(-2\boldsymbol{X}^T\boldsymbol{P}^t\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{P}^t(\boldsymbol{P}^t)^T\boldsymbol{X}) + \lambda\|\boldsymbol{P}^t\|_{2,1}
\end{aligned} \tag{48}$$

In summary, the objective function in (25) decreases monotonically in the alternative updating rules in (30) and (34).

4. Experiments and analysis

In this section, we present the experimental clustering results on some datasets. Our experiments have two parts. We firstly show the comparison results of the proposed DFSC and other feature selection algorithms on 7 datasets. Then we compare the proposed algorithm with some co-clustering algorithms. We also give an analysis of the results.

4.1. Comparison with other feature selection algorithms

4.1.1. The Compared algorithms

DFSC is an innovative feature selection algorithm. It preserves the geometrical information of both data space and feature space simultaneously, which is the key difference from previous feature selection algorithms that performed only in data space. DFSC is related to some other feature selection algorithms.

DFSC does not use the projection matrix, but it makes use of self-representation property of data and features, and it uses self-representation coefficients matrix in the data space as importance measurement for the self-representation construction of feature among all features. The compared algorithms include LapScore [20], SPEC [21], MCFS [22], JELSR [14], MRSF [23] and LSPE [24]. LapScore only preserves the locality of data manifold. SPEC can be seen as an extension of LapScore, but it is mainly used for supervised feature selection [27]. MCFS preserves the multi-cluster structure of the dataset, considering the relation between different clusters. JELSR unifies embedded learning and sparse regression in an unsupervised feature selection framework, and the learned sparse projection matrix is used to select features. MRSF is based on sparse multi-output model to minimize redundancy features. LSPE integrates embedded learning and feature selection in a joint framework. All the above feature selection algorithms are performed in data manifold space. Some preserve the local information, and some preserve similarity to improve the learning performance. DFSC preserves not only the manifold information of data, but also the manifold information of feature space, where the learned self-representation coefficients matrix in the data space is used to rank features.

4.1.2. Datasets

We first compare clustering ability between the proposed algorithm and some other feature selection algorithms on several datasets. The datasets is similar to those in [27], shown in Table 2.

4.1.3. Evaluation metrics

We evaluate the performance of clustering by two widely used evaluation matrices, i.e., clustering Accuracy (ACC) [51,52] and Normalized Mutual Information (NMI). The larger value of ACC and NMI indicate better performance.

Given a data point $\boldsymbol{x}_i$, c_i and g_i are clustering label and the ground truth label of $\boldsymbol{x}_i$ respectively. ACC is defined by

$$ACC = \frac{\sum_{i=1}^{n} \delta(g_i, map(c_i))}{n} \tag{49}$$

where n is the total number of data, $\delta(.,.)$ is the *delta* function defined by $\delta(x,y) = \begin{cases} 1, & x = y \\ 0, & otherwise \end{cases}$, and $map(.)$ is the optimal mapping function using *Hungarian* algorithm [53] to permute clustering labels and the ground truth labels.

NMI is defined as

$$NMI = \frac{MI(C, C')}{\max(H(C), H(C'))} \tag{50}$$

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

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j).log_2 \frac{p(c_i, c'_j)}{p(c_i).p(c'_j)} \tag{51}$$

where $p(c_i)$ and $p(c'_j)$ denote the probabilities 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.

ACC is based on one-to-one match between clustering labels and the ground truth labels. NMI is an external criterion, which evaluates the degree of similarity between clustering labels and the ground truth labels. ACC and NMI are two clustering evaluation criteria, they may not be best on one dataset simultaneously.

Table 2
Datasets used in this paper.

Dataset	Dimensionality	Size	Class
Umist	644	575	20
Isolet	617	1560	26
ORL	1024	400	40
Sonar	60	208	2
BC	30	569	2
Ionosphere	34	351	2
Dbworld_bodies	4702	64	2

4.1.4. Experimental settings

We also use all features as the baseline. For graph-based algorithms, such as DFSC, LSPE, LapScore, JELSR, SPEC and MCFS, the neighborhood size of graph is chosen from {3, 5, 7, 10, 15}. The bandwidth σ for Gaussian function is chosen from $\{10^0, 10^3, 10^5\}$. For LSPE, α is chosen from {300, 500, 800, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000}. We tune β from {0.01, 0.1, 0.5, 1.0, 3.0, 5.0, 7.0, 9.0, 11.0, 13.00, 15.00, 17.00}. For DFSC, we set α as {0.01, 0.1, 0.5, 1.0, 3.0, 5.0, 7.0, 9.0, 11.0, 13.00, 15.00, 17.00}. λ is searched from{300, 500, 800, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000}. We tune β in the range of {10, 100, 1000}. The number of clusters is set equal to the true number of clusters.

We tune these parameters so that the algorithms have the best ACC and NMI. Different parameters may be used for different datasets. We cluster samples based on the selected features using *K-means* algorithm. We repeat the clustering 100 times with random for each step since the performance of *K-means* algorithm depends on initialization.

4.1.5. Experimental results and analysis

We record the best clustering results from the optimal parameters. The average ACC with standard deviation (std) on 7 datasets is reported in Table 3. We highlight the best results in bold.

Table 4 shows clustering results of these feature selection algorithms in terms of NMI on these datasets, the best results are marked in bold.

From Tables 3 and 4, we have the following observations. DFSC is superior to all other algorithms and acquires the best result in terms of clustering accuracy on almost all the datasets. It is evident that the proposed algorithm has satisfactory performance, which demonstrates the effectiveness of the proposed algorithm. Compared with other feature selection algorithms, the main improvement is that DFSC utilizes the information in feature space by self-representation property. We can draw a conclusion that the information in feature space is of great importance for clustering. We know that SPEC, MCFS and MRSF are two-stage feature selection algorithms, while JELSR,

Table 3
ACC of some feature selection algorithms on seven datasets (MEAN ± STD%).

Algorithms	Umist	Isolet	ORL	Ionosphere	Sonar	BC	Dbworld_bodies
All features	44.23 ± 1.02	50.58 ± 0.85	50.00 ± 0.43	63.81 ± 0.50	54.32 ± 1.20	72.27 ± 0.20	73.81 ± 0.00
LapScore	37.30 ± 0.93	48.79 ± 0.56	44.50 ± 0.73	66.94 ± 2.20	58.80 ± 1.14	70.17 ± 0.36	73.47 ± 1.16
SPEC	42.56 ± 1.20	49.50 ± 0.63	49.88 ± 0.23	67.70 ± 2.33	61.00 ± 1.26	74.00 ± 0.23	77.94 ± 1.85
MCFS	46.55 ± 1.00	54.48 ± 0.84	49.40 ± 0.93	57.26 ± 3.00	54.20 ± 0.84	71.00 ± 0.58	91.13 ± 1.04
JELSR	48.90 ± 1.03	55.08 ± 0.45	50.02 ± 0.56	67.90 ± 2.81	64.20 ± 0.94	74.20 ± 0.30	90.63 ± 0.00
MRSF	48.38 ± 1.05	50.80 ± 0.69	49.78 ± 0.69	63.00 ± 2.30	60.33 ± 1.40	72.79 ± 0.22	85.02 ± 1.59
LSPE	49.26 ± 1.12	56.11 ± 0.63	50.25 ± 0.80	70.00 ± 2.66	**66.25 ± 1.67**	75.86 ± 0.24	**93.75 ± 0.00**
DFSC	**50.12 ± 2.79**	**60.14 ± 3.51**	**51.71 ± 2.61**	**82.90 ± 0.29**	**58.57 ± 2.31**	**85.41 ± 0.00**	91.75 ± 1.09

Table 4
Clustering NMI of feature selection algorithms on seven datasets (MEAN ± STD%).

Algorithms	Umist	Isolet	ORL	Ionosphere	Sonar	BC	Dbworld_bodies
All features	60.30 ± 1.45	73.02 ± 0.92	70.36 ± 1.17	13.12 ± 0.00	0.88 ± 0.00	17.61 ± 0.00	24.00 ± 0.00
LapScore	56.32 ± 1.52	66.80 ± 1.20	67.80 ± 1.76	8.16 ± 0.00	1.68 ± 0.00	16.79 ± 0.00	23.82 ± 1.01
SPEC	57.04 ± 1.24	66.90 ± 1.49	70.26 ± 1.65	8.33 ± 0.00	5.97 ± 0.42	18.83 ± 0.00	25.20 ± 1.62
MCFS	69.20 ± 1.31	70.43 ± 1.93	70.98 ± 1.78	1.01 ± 0.77	1.87 ± 2.85	17.32 ± 0.00	67.88 ± 1.62
JELSR	70.18 ± 1.64	70.50 ± 1.34	70.20 ± 1.72	7.84 ± 1.21	6.24 ± 0.00	18.86 ± 0.00	54.89 ± 0.00
MRSF	66.67 ± 1.43	68.35 ± 1.67	70.50 ± 1.81	3.82 ± 0.00	2.96 ± 1.04	17.32 ± 0.00	56.79 ± 2.39
LSPE	**70.91 ± 1.50**	71.01 ± 1.85	71.04 ± 1.11	13.10 ± 0.49	**7.24 ± 0.38**	18.83 ± 0.00	**68.09 ± 0.00**
DFSC	65.85 ± 1.76	**73.98 ± 1.33**	**73.27 ± 1.25**	**30.52 ± 0.79**	2.22 ± 1.03	**42.23 ± 0.00**	58.93 ± 3.67

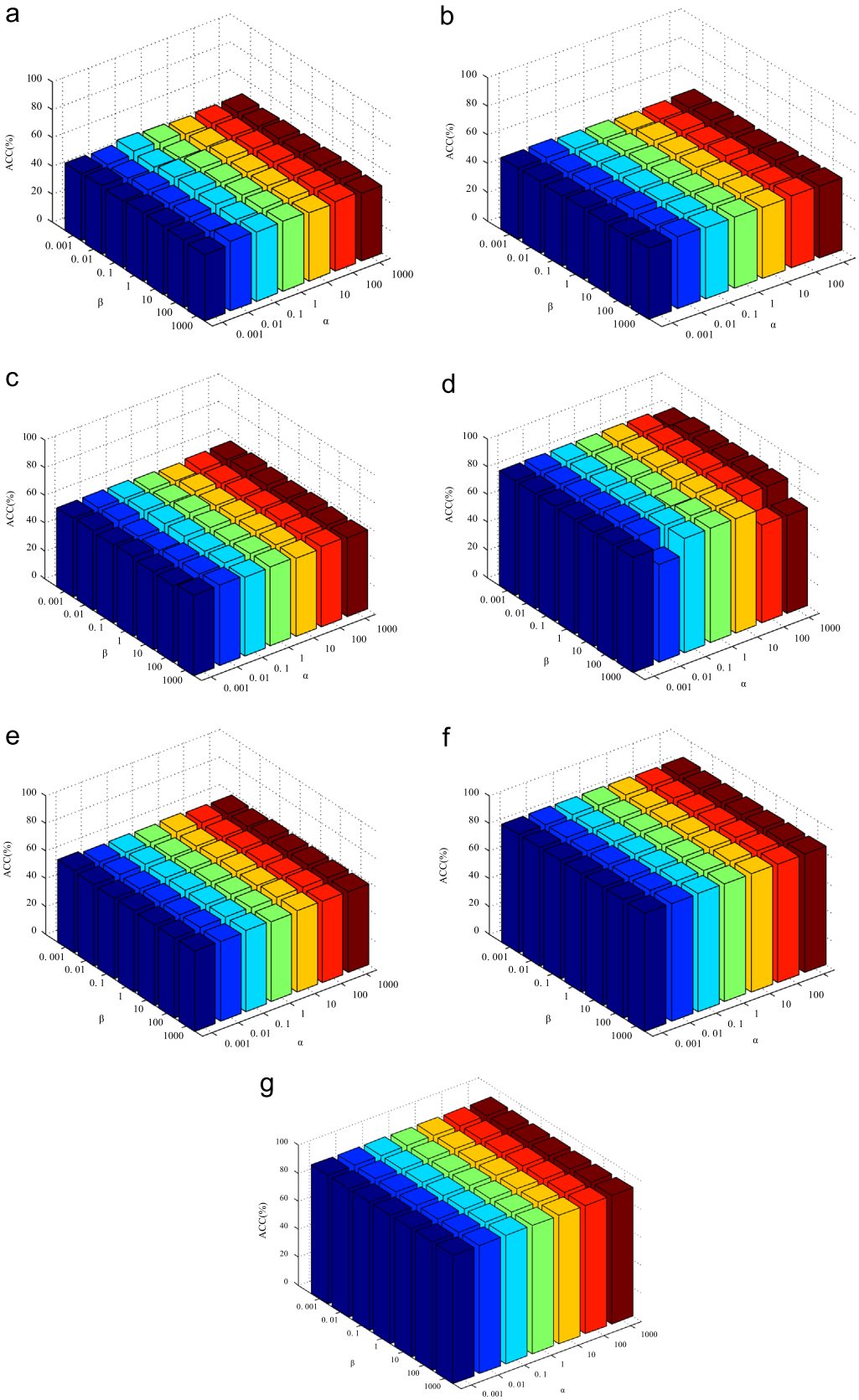

Fig. 1. Clustering accuracy with regard to different values of α and β. (a) Umist (b) ORL, (c) Isolet (d) Ionosphere, (e) Sonar (f) BC, and (g) Dbworld_bodies.

LSPE and DFSC simultaneously solving two objective functions. JELSR unifies embedded learning and sparse regression, LSPE integrates embedded learning and feature selection, and DFSC combines self-representation, manifold embedding and feature selection. Overall, JELSR, LSPE and DFSC have better clustering quality than other algorithms, which indicates that simultaneously solving several

problems is superior to solving problems in sequence. LSPE is the second best algorithm in our experiments, which validates that it is a better way to solve embedding learning and feature selection jointly for feature selection.

4.1.6. Parameters sensitivity

There are some parameters needed to be set in advance for DFSC, such as graph neighborhood number k, Gaussian kernel bandwidth parameter σ, λ, regularization parameters α and β, and the number of selected parameters q. Here we only focus on the sensitivities of α, β, λ and σ. As for analyzing the clustering results with regard to different values of α and β, we fix $\lambda=8000$ the parameters k, σ and q as constants. We chose α and β from a wide range $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. We record the average results of 20 runs, and plot seven 3-D figures in Fig. 1.

From Fig. 1, we have some observations. On "Ionosphere" dataset, when $\beta=1000$, the clustering results fluctuate with changing α. DFSC has really consistent results in terms of clustering ACC with regard to different values of α and β in general, which demonstrates that DFSC is insensitive to regularization parameters α and β.

Now, we change the value of σ with other parameters fixed. We select the bandwidth parameter σ from a wide range $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5\}$. The performance of DFSC is quite steady on these datasets with the changingσ. We take "Ionosphere" dataset as an example to present how the performance of DFSC changes with the changing σ in Fig. 2.

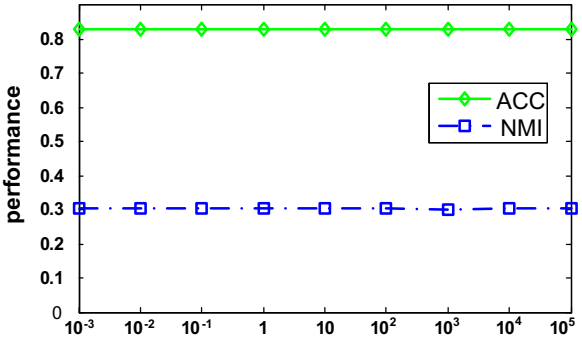

Fig. 2. Clustering performance on "Ionosphere" dataset with regard to different values of σ.

From Fig. 2, it is clear that the performance of DFSC is really stable with changing σ.

Similarly, just like parameter σ, when parameter λ changes from a wide range $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5\}$, the clustering results hardly change.

We also present how the results of LSPE and DFSC change in a wider range. We perform experiments when σ is less than 1, α for LSPE less than 300, and β for LSPE greater than 20. Similarly, we set α for DFSC greater than 20, and β for DFSC less than 10. We record the average results of 20 runs, shown in Tables 5 and 6. All the results remain approximately constant. When σ changes from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5\}$, the clustering results of LSPE hardly change. When β changes from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5\}$, the clustering results of DFSC hardly change. We take "Sonar" dataset as an example to show how the performances change when $\sigma=10^{-3}$ for LSPE and when $\beta=10^{-3}$ for DFSC in Tables 5 and 6 respectively.

From the data in Table 5, we have conclusion that the clustering results of LSPE are insensitive to α and β.

From the data in Table 6, we have conclusion that the clustering results of DFSC are insensitive to α and λ.

4.2. Comparisons with co-clustering algorithms

DFSC has some connection with co-clustering algorithms, what they have in common is that both of them have considered the information of feature space and data space. But DFSC belongs to feature selection algorithm, since it has a "selection" process, where it chooses the most important features from all the features, and removes related features to avoid redundancy.

Next, we conduct experiments on COIL20 dataset to compare clustering qualities using DFSC, matrix factorization based algorithms (NMF, CF, LCCF) and co-clustering algorithms (DRCC, GCF). COIL20 dataset consists of 1440 images of 20 objects, each image is scaled to 32×32 pixel, and each image is represented by a 1024-dimensional vector.

In the test, we explore the clustering performance of these algorithms with different clusters. We take *K-means* and NMF as baselines. For LCCF, DRCC, GCF and DFSC, we use 0–1 weighting scheme to construct neighborhood graph and set the size of neighborhood graph $p=5$. For LCCF algorithm, we set $\lambda=100$. For DRCC and GCF, we set $\lambda=\mu=100$. For fair comparison and adjustment

Table 5
Clustering ACC (first row) and NMI (second row) of LSPE with regard to different values of α and β on "Sonar" dataset.

α	β 0.01	0.1	1	10	20	50	100	200	500	1000
0.001	0.5260	0.5264	0.5235	0.5250	0.5240	0.5276	0.5269	0.5276	0.5288	0.5255
	0.0013	0.0012	0.0007	0.0021	0.0008	0.0017	0.0014	0.0017	0.0017	0.0014
0.01	0.6058	0.6019	0.6310	0.6082	0.6038	0.6029	0.6062	0.6086	0.6091	0.5990
	0.0330	0.0315	0.0366	0.0349	0.0324	0.0322	0.0335	0.0350	0.0350	0.0306
0.1	0.5365	0.5336	0.5308	0.5336	0.5336	0.5336	0.5394	0.5336	0.5336	0.5322
	0.0060	0.0053	0.0047	0.0053	0.0053	0.0053	0.0066	0.0053	0.0054	0.0050
1	0.5560	0.5567	0.5575	0.5558	0.5575	0.5582	0.5539	0.5582	0.5560	0.5567
	0.0071	0.0093	0.0095	0.0090	0.0095	0.0097	0.0086	0.0097	0.0091	0.0093
10	0.5570	0.5529	0.5488	0.5488	0.5567	0.5553	0.5519	0.5510	0.5613	0.5534
	0.0085	0.0077	0.0069	0.0069	0.0084	0.0082	0.0075	0.0073	0.0093	0.0078
100	0.5500	0.5507	0.5498	0.5498	0.5512	0.5534	0.5503	0.5495	0.5507	0.5505
	0.0065	0.0064	0.0065	0.0065	0.0063	0.0078	0.0064	0.0066	0.0064	0.0064
200	0.5507	0.5505	0.5500	0.5505	0.5503	0.5500	0.5491	0.5503	0.5512	0.5488
	0.0064	0.0064	0.0065	0.0065	0.0064	0.0065	0.0066	0.0064	0.0063	0.0067
500	0.5501	0.5507	0.5493	0.5493	0.5500	0.5510	0.5500	0.5507	0.5507	0.5495
	0.0065	0.0064	0.0066	00066	0.0065	0.0063	0.0065	0.0064	0.0064	0.0067
1000	0.5500	0.5500	0.5500	0.5500	0.5503	0.5495	0.5498	0.5498	0.5503	0.5505
	0.0065	0.0065	0.0065	0.0065	0.0064	0.0066	0.0065	0.0065	0.0064	0.0064

Table 6
Clustering ACC (first row) and NMI (second row) of DFSC with regard to different values of α and λ on "Sonar" dataset.

	α									
λ	0.01	0.1	1	10	20	50	100	200	500	1000
0.001	0.5846	0.5791	0.5820	0.5789	0.5767	0.5794	0.5741	0.5813	0.5810	0.5818
	0.0226	0.0200	0.0215	0.0199	0.0190	0.0202	0.0178	0.0209	0.0207	0.0212
0.01	0.5760	0.5873	0.5851	0.5808	0.5866	0.5765	0.5810	0.5863	0.5813	0.5863
	0.0185	0.0238	0.0230	0.0205	0.0233	0.0188	0.0206	0.0231	0.0209	0.0231
0.1	0.5868	0.5976	0.5914	0.5762	0.5791	0.5794	0.5863	0.5741	0.5760	0.5813
	0.0235	0.0284	0.0253	0.0186	0.0201	0.0202	0.0231	0.0178	0.0183	0.0208
1	0.5741	0.5844	0.5837	0.5887	0.5810	0.5839	0.5688	0.5871	0.5885	0.5842
	0.0178	0.0225	0.0219	0.0241	0.0207	0.0221	0.0154	0.0237	0.0239	0.0222
10	0.5837	0.5736	0.5818	0.5736	0.5861	0.5839	0.5709	0.5813	0.5844	0.5813
	0.0219	0.0174	0.0213	0.0175	0.0228	0.0221	0.0163	0.0209	0.0224	0.0209
100	0.5791	0.5861	0.5839	0.5837	0.5738	0.5844	0.5815	0.5736	0.5866	0.5813
	0.0201	0.0229	0.0221	0.0218	0.0176	0.0225	0.0220	0.0174	0.0233	0.0209
200	0.5868	0.5897	0.5714	0.5789	0.5818	0.5767	0.5839	0.5743	0.5808	0.5794
	0.0234	0.0248	0.0167	0.0199	0.0213	0.0190	0.0221	0.0180	0.0205	0.0202
500	0.5738	0.5914	0.5815	0.5794	0.5770	0.5849	0.5866	0.5784	0.5851	0.5916
	0.0176	0.0293	0.0210	0.0203	0.0192	0.0228	0.0233	0.0195	0.0230	0.0255
1000	0.5842	0.5791	0.5765	0.5717	0.5731	0.5791	0.5868	0.5844	0.5794	0.5902
	0.0223	0.0200	0.0188	0.0168	0.0171	0.0200	0.0235	0.0225	0.0230	0.0252

Table 7
Clustering ACC on COIL20.

K	2	3	4	5	6	7	8	9	10	Avg.
KM	92.71	79.35	73.19	71.67	67.78	68.34	66.13	66.23	64.60	72.22
NMF	89.84	77.80	73.01	70.36	65.20	64.64	65.16	64.87	65.37	70.69
CF	89.72	79.34	73.04	71.33	75.21	63.85	64.64	62.86	62.15	71.34
DRCC	91.04	83.42	80.36	75.15	77.74	70.13	71.67	67.42	68.97	76.21
LCCF	90.74	84.22	78.14	74.46	79.59	70.08	71.64	67.87	65.71	75.82
GCF	92.48	85.36	82.69	79.23	82.90	73.62	75.51	70.02	68.44	78.91
DFSC	**100.00**	**92.01**	**90.10**	**80.27**	**84.84**	**81.94**	**80.44**	**79.19**	**72.32**	**84.56**

Table 8
Clustering NMI on COIL20.

K	2	3	4	5	6	7	8	9	10	Avg.
K-means	79.64	66.11	67.56	68.95	71.51	72.17	71.32	72.39	70.57	71.13
NMF	71.25	63.42	67.87	66.07	68.34	70.14	70.40	71.65	71.89	69.00
CF	71.13	63.21	66.38	67.67	65.33	66.67	67.28	66.40	66.27	66.70
DRCC	77.29	74.57	75.14	72.26	72.86	73.42	73.89	70.38	69.40	73.25
LCCF	74.51	68.69	70.63	72.22	68.81	70.57	70.67	69.86	68.69	70.52
GCF	80.40	76.35	77.43	78.56	74.89	75.31	76.45	72.71	70.63	75.86
DFSC	**100.00**	**90.97**	**93.97**	**82.77**	**86.09**	**83.28**	**84.91**	**74.17**	**76.43**	**85.84**

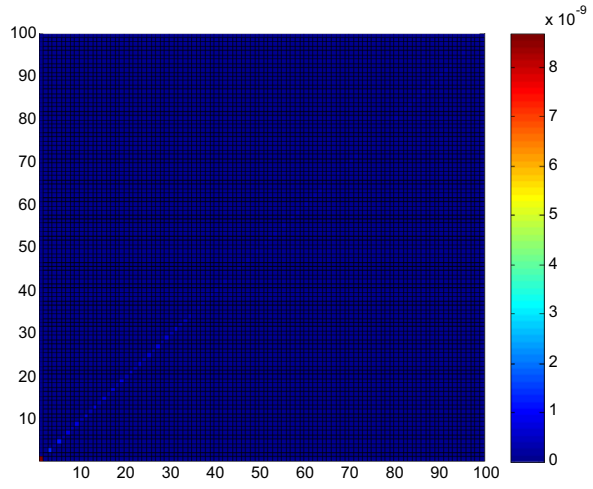

Fig. 3. The self-representation coefficients matrix ***P***.

simplicity, we set $\alpha = 100$ for DFSC, we fixed $\lambda = 10^8$. $\beta$is chosen from $\{10^{-1}, 1, 10, 10^2\}$ and the number of selected features q is chosen from set{20, 40,. , 200}. We evaluate the performances of these algorithms in terms of ACC and NMI. For each given cluster number, 20 runs are conducted on different randomly selected clusters, and we recode the average results in Tables 7 and 8.

From the results shown in Tables 7 and 8, we can observe the following. *K-means*, NMF and CF perform generally much inferior, because they do not consider the geometric information of the dataset. LCCF achieves better results than *K-means*, NMF and CF. LCCF seeks to capture the local geometry. DRCC and GCF obtain good results, because they consider the information of both data space and feature space. The overall results of GCF and DRCC are better than others, since the information of feature space is considered to improve accuracy. DFSC achieves the best clustering results. Both GCF and DFSC consider the information of the data manifold and feature manifold. But DFSC has another selection process that can select the most effective features and avoid redundancy, thus it improves learning quality effectively. Compared with KM, NMF, CF and LCCF, DFSC makes use of the

information in feature space and self-representation property, and it improves the clustering quality significantly. All these results demonstrate that based on self-representation property, DFSC preserves the geometrical structure of both data space and feature space, and utilizes the self-representation coefficients matrix in data space to select the most effective features, which is beneficial to improve clustering power.

5. The effectiveness of the proposed algorithm

In this section, we first illustrate the effectiveness of the proposed feature selection algorithm. We use "Ionosphere" dataset as an example to test whether DFSC can find the most representative features. The original Ionosphere dataset has 351 samples and 34 features, and we artificially generate 66 features as the liner combination of the original 34 features with randomly generated combination coefficients, the sum of combination coefficients being 1. Now, we get a synthetic data matrix with 351 samples and 100 features, the first 34 features are the original features.

By applying the proposed DFSC to the obtained synthetic dataset, we get the coefficients matrix $\boldsymbol{P}$, and we show the coefficients matrix $\boldsymbol{P}$ in Fig. 3.

We can clearly see from Fig. 3 that the coefficients of the original 34 features are much larger than those of the other features. Note that the last 66 features are generated from the original features. This experiment validates that DFSC can select the most representative features.

6. Conclusions

This paper presents a novel feature selection clustering algorithm named self-representation based dual-graph regularized feature selection clustering (DFSC). Since recent studies have shown that feature manifold also contains underlying information of dataset, we construct data graph and feature graph, and utilize self-representation property in data space and feature space, the learned data space and the feature space of self-representation coefficients matrix $\boldsymbol{P}$ and $\boldsymbol{S}$ are used for preservation of the local geometrical structures of data space and feature space respectively. We exert a sparse constraint on the self-representation matrix $\boldsymbol{P}$ in data space, and we rank all the features based on $\boldsymbol{P}$ to select the most representative features for clustering. DFSC considers the information of feature space, while conventional feature selection algorithms neglect. Information of feature space also reflects the underlying structure of the dataset, which contributes to improving the discriminative power. On the other hand, DFSC and co-clustering algorithms have some relations, both of which are conducted simultaneously on the rows and columns of data. The difference is that DFSC exploits self-representation property, and determines the importance of features according to $\boldsymbol{P}$, thus it has an additional selection process, where irrelevant or redundant features are removed.

In Section 4, the results on some datasets are not sound, one reason may be that the proposed DFSC optimizes the variables $\boldsymbol{P}$ and $\boldsymbol{S}$ independently. It is expected to develop an optimization mechanism that can update $\boldsymbol{P}$ and $\boldsymbol{S}$ simultaneously. On the other hand, self-representation property is based on the nature fact that redundancy exists in features. However, for a dataset which has strong independence between the data or features, or the correlation is weak, the self-representation property is not very suitable.

Acknowledgment

We would like to express our sincere appreciation to the anonymous reviewers for their valuable comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Basic Research Program (973 Program) of China (No. 2013CB329402), the National Natural Science Foundation of China (Nos. 61371201 and 61272279), and the EUFP7 project (No. 247619) on "Nature Inspired Computation and its Applications".

References

[1] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, IEEE Trans. Cybern. 19 (2) (1997) 153–158.
[2] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization, Adv. Neural Inf. Process. Syst. (2012) 1813–1821.
[3] Y. Sun, S. Todorovic, S. Goodison, Local-learning-based feature selection for high-dimensional data analysis, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1610–1626.
[4] N. Kwak, C. Choi, Input feature selection by mutual information based on parzen window, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1667–1671.
[5] W.H. Abdulla, N. Kasabov, Reduced feature-set based parallel CHMM speech recognition systems, Inf. Sci. 156 (1–2) (2003) 21–38.
[6] L. Goh, Q. Song, N. Kasabov, A novel feature selection method to improve classification of gene expression data, in: Proceedings of the second conference on Asia-Pacific bioinformatics, 2004, pp. 161–166.
[7] X. Jin, A. Xu, R. Bie, P. Guo, Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles, Data Min. Biomed. Appl. (2006) 106–115.
[8] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (4) (2005) 491–502.
[9] M. Sikonja, I. Kononenko, Theoretical and empirical analysis of relief and relieff, Mach. Learn. 53 (1-2) (2003) 23–69.
[10] Z. Xu, I. King, M.R. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Netw. 21 (7) (2010) 1033–1047.
[11] J.G. Dy, C.E. Brodley, Feature selection for unsupervised learning, J. Mach. Learn. Res. 5 (2004) 845–889.
[12] Z. Li, J. Liu, Y. Yang, X. Zhou, H. Lu, Clustering-guided sparse structural learning for unsupervised feature selection, IEEE Trans. Knowl. Data Eng. 26 (9) (2014) 2138–2150.
[13] P. Mitra, C.A. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 301–312.
[14] 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) (2014) 793–804.
[15] F. Shang, L.C. Jiao, F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, Pattern Recognit. 45 (2012) 2237–2250.
[16] F. Nie, D. Xu, I.W. Tsang, C. Zhang, Flexible manifold embedding: a framework for semi-supervised and unsupervised dimension reduction, IEEE Trans. Image Process 19 (7) (2010) 1921–1932.
[17] Y. Pang, Y. Yuan, X. Li, Effective feature extraction in high dimensional space, IEEE Trans. Syst. Man Cybern. B 38 (6) (2008) 1652–1656.
[18] S. Zhou, X. Liu, C. Zhu, Q. Liu, J. Yin, Spectral clustering-based local and global structure preservation for feature selection, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2014, pp. 550–557.
[19] H. Liu, Y. Mo, J. Wang, J. Zhao, A new feature selection method based on clustering, in: Proceedings of the Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2011, pp. 965–969.
[20] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, Adv. Neural Inf. Process. Syst. 18 (2005) 507–514.
[21] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of 24th International Conference on Machine Learning, 2007, pp. 1151–1158.
[22] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proceedings of the16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 333–342.
[23] Z. Zhao, L. Wang, H. Liu, Efficient spectral feature selection with minimum redundancy, in: Proceedings of 24th AAAI Conference on Artificial Intelligence, 2010, pp. 673–678.
[24] 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.
[25] S. Bandyopadhyay, T. Bhadra, P. Mitra, U. Maulik, Integration of dense subgraph finding with feature clustering for unsupervised feature selection, Pattern Recognit. Lett. 40 (2014) 104–112.
[26] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, X. Zhou, $L_{2,1}$-norm regularized discriminative feature selection for unsupervised learning, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2011, pp. 1589–1594.

[27] L. Du, Z. Shen, X. Li, P. Zhou, Y.D. Shen, local and global discriminative learning for unsupervised feature selection, IEEE 13th International Conference on Data Mining, 2013, pp. 131–140.
[28] J.J. Wang, J.Z. Huang, Y. Sun, X. Gao, Feature selection and multi-kernel learning for adaptive graph regularized nonnegative matrix factorization, Expert Syst. Appl. 42 (3) (2015) 1278–1286.
[29] N. Ahmed, A. Jalil, A. Khan, Feature selection based image clustering using local discriminant model and global integration, in: Proceedings of the IEEE 14th International Multitopic Conference (INMIC), 2011, pp. 13–18.
[30] D. Cai, X. He, X. Wu, J. Han, Non-negative matrix factorization on manifold, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 63–72.
[31] C. Lin, On the convergence of multiplicative update algorithms for non-negative matrix factorization, IEEE Trans. Neural Netw. 18 (6) (2007) 1589–1596.
[32] X. Yan, Y. Zhu, W. Zou, L. Wang, A new approach for data clustering using hybrid artificial bee colony algorithm, Neurocomputing 97 (15) (2012) 241–250.
[33] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[34] W. Xu, Y. Gong, Document clustering by concept factorization, in: Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval (SIGIR'04), 2004, pp. 202–209.
[35] D. Cai, X. He, J. Han, T. Huang, Graph regularized nonnegative matrix factorization for data representation, EEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.
[36] D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, IEEE Trans. Knowl. Data Eng. 23 (6) (2011) 902–913.
[37] J. Ye, Z. Jin, Dual-graph regularized concept factorization for clustering, Neurocomputing 138 (2014) 120–130.
[38] I.S. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2001, pp. 269–274.
[39] I.S. Dhillon, S. Mallela, D.S. Modha, Information-theoretic co-clustering, in: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2003, pp. 89–98.
[40] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix tri-factorization for clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2006, pp. 126–135.
[41] P. Zhu, W. Zuo, L. Zhang, Q. Hu, S.C.K. Shiu, Unsupervised feature selection by regularized self-representation, Pattern Recognit. 48 (2015) 438–446.
[42] Y. Zhang, J. Liu, M. Li, Z. Guo, Joint image denoising using adaptive principal component analysis and self-similarity, Inf. Sci. 259 (2014) 128–141.
[43] G. Boracchi, M. Roveri, Exploiting self-Similarity for change detection, in: Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), 2014, pp. 3339–3346.
[44] V. Sindhwani, J. Hu, A. Mojsilovic, Regularized co-clustering with dual supervision, Adv. Neural Inf. Process. Syst. (2009) 1505–1512.
[45] Q. Gu, J. Zhou, Co-clustering on manifolds, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 359–368.
[46] G. Yao, K. Lu, X. He, G-Optimal feature selection with Laplacian regularization, Neurocomputing 119 (2013) 175–181.
[47] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[48] J. Tenenbaum, V. de Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[49] M. Belkin, P. Niyogi, Laplacian Eigenmaps and spectral techniques for embedding and clustering, Adv. Neural Inf. Process. Syst. (2001) 585–591.
[50] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 1735–1742.
[51] D. Cai, X. He, J. Han, T. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.
[52] H. Liu, Z. Wu, X. Li, D. Cai, T.S. Huang, Constrained nonnegative matrix factorization for imagine representation, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1299–1311.
[53] C. Papadimitriou, K. Steiglitz, Combinatorial optimization: algorithms and complexity, Dover, New York, 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 an associate professor with Xidian University. Her current research interests include optimization problems, evolutionary computation, artificial immune systems, and data mining.

Zhu Zhang received the B.S. degree in Electronic Information Engineering from Yangtze University, Hubei, China, in July 2013. Since September 2013, she has been a post graduate with the School of Electronic Engineering, Xidian University, Xi'an, China. Her current research interests include pattern recognition and machine 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, Xi'an, China, 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. In 1992, Dr. Jiao was awarded the Youth Science and Technology Award. In 1996, he was granted by the Cross-century Specialists Fund from the Ministry of Education of China. And he was selected as a member of the First level of Millions of Talents Project of China from 1996. In 2006, he was awarded the First Prize of Young Teacher Award of High School by the Fok Ying Tung Education Foundation. From 2006, he was selected as an Expert with the Special Contribution of Shaanxi Province. In 2007, as a principal member, he and his colleagues founded an Innovative Research Team of the Ministry of Education of 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.