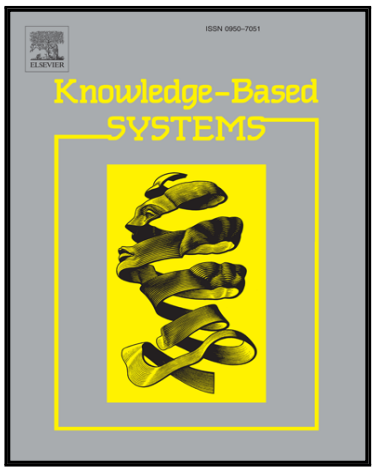
Subspace learning-based graph regularized feature selection

Shang, Ronghua; Wang, Wenbing; Stolkin, Rustam; Jiao, Licheng

DOI:
10.1016/j.knosys.2016.09.006

Document Version
Peer reviewed version

Citation for published version (Harvard):
Shang, R, Wang, W, Stolkin, R & Jiao, L 2016, 'Subspace learning-based graph regularized feature selection', *Knowledge-Based Systems*, vol. 112, pp. 152-165. https://doi.org/10.1016/j.knosys.2016.09.006



Subspace learning-based graph regularized feature selection

Ronghua Shang , Wenbing Wang , Rustam Stolkin , Licheng Jiao

PII: S0950-7051(16)30322-7
DOI: 10.1016/j.knosys.2016.09.006
Reference: KNOSYS 3664

To appear in: *Knowledge-Based Systems*

Received date: 27 May 2016
Revised date: 23 August 2016

Please cite this article as: Ronghua Shang , Wenbing Wang , Rustam Stolkin , Licheng Jiao , Subspace learning-based graph regularized feature selection, *Knowledge-Based Systems* (2016), doi: 10.1016/j.knosys.2016.09.006

Subspace learning-based graph regularized feature selection

Ronghua Shang[1], Wenbing Wang[1], Rustam Stolkin[2] and Licheng Jiao[1]

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

[2]Extreme Robotics Lab, University of Birmingham UK.)

Abstract: In recent years, a variety of feature selection algorithms based on subspace learning have been proposed. However, such methods typically do not exploit information about the underlying geometry of the data. To overcome this shortcoming, we propose a novel algorithm called subspace learning-based graph regularized feature selection (SGFS). SGFS builds on the feature selection framework of subspace learning, but extends it by incorporating the idea of graph regularization, in which a feature map is constructed on the feature space in order to preserve geometric structure information on the feature manifold. Additionally, the $L_{2,1}$-*norm* is used to constrain the feature selection matrix to ensure the sparsity of the feature array and avoid trivial solutions. The resulting method can provide more accurate discrimination information for feature selection. We evaluate SGFS by comparing it against five other state-of-the-art algorithms from the literature, on twelve publicly available benchmark data sets. Empirical results suggest that SGFS is more effective than the other five feature selection algorithms.

Key words: Graph regularized; subspace learning; feature manifold; sparse constraint; feature selection

1. Introduction

The increasingly rapid growth of information technology has seen a corresponding growth in the number of dimensions of gathered data. In many high dimensional data sets, only a small subset of the available features are useful, with most features being redundant, and some features even corresponding to information-less noise [1], [2], [3], [4], [5]. To facilitate the subsequent processing, it is often necessary to reduce the dimension of such high dimensional data. Processing of high dimensional data has become a challenge for researchers in many different fields [6], [7], [8], including data mining, machine learning, pattern recognition and others. Dimensionality reduction

methods can broadly be categorized into methods for feature selection and feature extraction [9], [10], [11], [12], [13]. Feature selection methods select representative features from the original set of features based on a variety of evaluation methods. In contrast, feature extraction methods map high dimensional data into a low dimensional space through a transformation matrix. Feature selection methods select a subset of the original “raw” featuire data, and so retain the physical or real-world meaning of the original data. This means that the performance of the resulting classifiers can often be readily explained in terms of intuitively meaningful trends in the underlying data. In contrast, it may be difficult to explain the behaviour of feature extraction methods in terms of the relationship between the new feature and the sample class [1]. In this paper, we propose a new feature selection algorithm.

Feature selection methods can broadly be divided into: supervised [2], [14], semi supervised [15], unsupervised [16], [17], [18]. In supervised feature selection problems, the data discrimination information and also the correlation between features and the class of each data sample is available during training. However, in order to obtain large amounts of such class information, need for training such methods, a large amount of human resources are typically required, e.g. for hand annotation of data [18]. Semi-supervised feature selection requires only a smaller portion of the training data to be annotated with class label information to improve the accuracy of feature selection [15]. Unsupervised feature selection, without any class label information, only relies on the inherent information of the input data to determine the importance of features [16]. In many practical applications, the true class label information is unknown, which makes unsupervised feature selection methods more widely applicable to real problems, but also engenders greater challenges for researchers. According to various possible search strategies, unsupervised feature selection can be divided into filter, wrapper and embedded [19], [20], [21], [22], [23], [24], [25], [26] methods.

In recent years, powerful new algorithms have been proposed which exploit the advantages of matrix decomposition techniques. Well known examples of such methods include nonnegative matrix factorization (NMF) [27], [28], principal component analysis (PCA) [29], [30] and singular value decomposition (SVD) [30], [31]. However, all of these are examples of feature extraction methods. A samller body of literature has explored how the idea of matrix decomposition can also be applied to feature selection. Wang et al. [32] proposed subspace learning for unsupervised feature

selection via matrix factorization (MFFS). In [33], Wang et al. proposed unsupervised feature selection via maximum projection and minimum redundancy (MPMR). These two algorithms find a suitable feature subspace through matrix decomposition, and the feature subspace is then used to represent the original feature space. By exploiting the advantages of the matrix factorization technique, MFFS and MPMR can both achieve good performance. However, MFFS and MPMR ignore the underlying geometry information of the data itself. In contrast, this paper shows how such geometry information can be used to further improve the quality of feature selection.

A variety of literature has shown that the distributions of high dimensional data are often sparse. Such data contain a lot of local information, which is important for mining the internal structure of such data and improving the performance of nonlinear learning [34], [35]. Some manifold learning algorithms have been proposed to discover the underlying manifold structure of data, such as Locality Preserving Projection [36], local linear embedding (LLE) [37] and Laplacian Eigenmap [38]. By analyzing the manifold structure of the data set, we can use the underlying geometric information to improve the learning efficiency of the algorithm.

Spectral graph theory [39], [40] can be used to characterize the underlying manifold structure of the data. The spectral clustering method [39] exploits spectral graph theory to obtain good clustering performance. Based on nonnegative matrix factorization (NMF) [28], Cai et al. [41] proposed graph regularized nonnegative matrix factorization (GNMF), which uses the geometry information of the data itself to greatly improve performance. Compared with concept factorization (CF) [42], locally consistent concept factorization (LCCF) [43] shows better performance, because it is able to exploit the local structure of data. In recent years, new work [44], [45], [46] has shown that the manifold information of the data is not only distributed in the data space, but also in the feature space. In [44], Shang et al. proposed a graph dual regularization non-negative matrix factorization for co-clustering algorithm (DNMF). Ye et al. [46] proposed dual-graph regularized concept factorization clustering (GCF).

Some classification algorithms also use the spectral graph theory. Belkin et al. [47] proposed manifold regularization, a geometric framework for learning from labeled and unlabeled examples, which uses graph theory to exploit the manifold structure of the data. Experimental results show that this method can use unlabeled data effectively. Based on standard SVM, Chova et al. proposed

semi-supervised image classification with Laplacian support vector machines (LapSVM) [48], which uses the geometry information of both labeled and unlabeled samples by using the graph Laplacian. Some experimental evidence suggests that LapSVM can outperform conventional SVM. In [49], Yang et al. proposed the Laplacian twin parametric-margin support vector machine for semi-supervised classification (LTPMSVM), which overcomes the shortcomings of conventional methods which are unable to effectively handle unlabeled data. LTPMSVM uses the geometric information of the unlabeled data to construct a better classifier, and experimental results have confirmed the strong performance of LTPMSVM.

Some feature selection algorithms which use local structure information have previously been proposed. Laplacian score (LapScor) [21], spectral feature selection (SPEC) [18], minimum redundancy spectral feature selection (MRSF) [50], unsupervised feature selection for multicluster data (MCFS) [51] are four well known algorithms. Extensions of MRSF and MCFS include clustering-guided sparse structural learning for unsupervised feature selection (CGSSL) [52] and joint embedding learning and sparse regression (JELSR) [53].

In this paper, we propose a new method called subspace learning-based graph regularized feature selection (SGFS). SGFS is based on the framework of subspace learning feature selection, which exploits the advantages of matrix factorization techniques. On this basis, we introduce the concept of graph regularization and preserve the local structure information of the feature space of the data. The local structure information of the feature space directly guides the learning of the coefficient matrix in the error reconstruction term, and indirectly guides the learning of the feature selection matrix. Additionally, we propose the use of the $L_{2,1}$-norm to constrain the feature selection matrix, which guarantees its sparsity, so as to provide more accurate discrimination information for feature selection. We use an alternating iterative optimization mechanism to optimize the objective function and adjust the parameters to minimize the reconstruction error. Finally, we obtain the feature selection matrix. Through this matrix, we can calculate the scores of all the features, and select the most representative features.

The main contributions of this paper are as follows:

1. By using graph theory, the geometric structure information of the feature manifold is preserved. Through the guidance of geometry information, the learning of the feature selection matrix and

coefficient matrix are more rapid and accurate.

2. By introducing $L_{2,1}$-norm to constrain the feature selection term, the sparsity of the feature selection matrix is guaranteed, enabling more accurate discrimination information for feature evaluation.

The structure of this paper is organized as follows. In Section 2, we introduce the framework, the iterative update rules and convergence proof of SGFS. In Section 3, we present the experimental results of comparing the performance of SGFS against five other state-of-the-art algorithms on twelve public benchmark data sets. Section 4 provides concluding remarks.

2. Subspace Learning-based Graph regularized Feature Selection

In this section, we present details of the SGFS method, which breaks down into three main parts: sparse subspace learning, local structure preserving and feature evaluation.

2.1 Related notations

First of all, we introduce some related notations. Denote $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n] \in \Re^{d \times n}$ as the unlabeled sample data set. Where n and d respectively represent the number and dimension of the samples. We use l to indicate the number of selected features, $l \leq d$.

Given an arbitrary matrix $\boldsymbol{A} \in \Re^{e \times f}$, its $L_{r,s}$ is defined as:

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

According to the definition, when $r=s=2$, it indicates Frobenius-norm or L_2-*norm*. In contrast, when $r=2$, $s=1$, it represents sparse constraint $L_{2,1}$-*norm*. We denote L_2-*norm* and $L_{2,1}$-*norm* respectively as $\|\cdot\|_2^2$ and $\|\cdot\|_{2,1}$ in the following.

2.2 Sparse subspace learning

2.2.1 Distance between subspaces

According to [32], we first define the distance between subspaces. Given a vector group $\boldsymbol{X}$ in an m-dimensional real number space. We define $span(\boldsymbol{X}) = \{\boldsymbol{a}^T \boldsymbol{X} \mid \boldsymbol{a} \in \Re^{|\boldsymbol{X}|}\}$ as the spanning subspace of $\boldsymbol{X}$, which is the set of all combinations of elements of $\boldsymbol{X}$. Where, $|\boldsymbol{X}|$ is the basis of $\boldsymbol{X}$. Given two

vector groups $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ in an m-dimensional real number space, the directional distance between *span* ($\boldsymbol{X}_1$) and *span* ($\boldsymbol{X}_2$) can be defined as:

$$\vec{d}(span(\boldsymbol{X}_1), span(\boldsymbol{X}_2)) = \sum_{\boldsymbol{x}\in\boldsymbol{X}_1} d(\boldsymbol{x}, span(\boldsymbol{X}_2)) = \min_{\boldsymbol{H}\in\Re^{|\boldsymbol{X}_2|\times|\boldsymbol{X}_1|}} \left\|\boldsymbol{X}_1 - \boldsymbol{X}_2\boldsymbol{H}\right\|_2^2. \tag{2}$$

where, $\boldsymbol{H}\in\Re^{l\times d}$ is the coefficient matrix, which is used to calculate the directional distance between *span* ($\boldsymbol{X}_1$) and *span* ($\boldsymbol{X}_2$).

2.2.2 Sparse subspace learning

The main purpose of subspace learning is to find a suitable feature subspace for representing the original feature space, referred to as the process of feature selection [32]. Denote $\boldsymbol{X}\in\Re^{d\times n}$, $\boldsymbol{X}_I\in\Re^{l\times n}$. Where $\boldsymbol{X}_I$ is the subspace of $\boldsymbol{X}$. According to the definition of equation (2), the problem of subspace learning or feature selection can be considered as solving the following problems:

$$\begin{aligned} \arg\ \min_{I}\ & \vec{d}(span(\boldsymbol{X}^T), span(\boldsymbol{X}_I{}^T)) \\ s.t.\ & |I| = l. \end{aligned} \tag{3}$$

where, I is the index set of selected features, $|I|$ represents the number of elements in set I. To minimize the distance of the two spaces, a suitable feature subset is obtained.

According to equation (2), formula (3) can be rewritten as:

$$\begin{aligned} \arg\ \min_{I}\ & \left\|\boldsymbol{X}^T - \boldsymbol{X}_I{}^T\boldsymbol{H}\right\|_2^2 \\ s.t.\ & |I| = l. \end{aligned} \tag{4}$$

where, $\boldsymbol{H}\in\Re^{l\times d}$ is the coefficient matrix, which is used for data reconstruction. By means of the coefficient matrix $\boldsymbol{H}$, a new element of data can be approximately reconstructed as a linear combination of the features of the feature subset. The feature subset and the coefficient matrix can be obtained simultaneously by minimizing the reconstruction error. According to the idea of matrix factorization [32], equation (4) can be rewritten as:

$$\begin{aligned} \arg\ \min_{\boldsymbol{W},\boldsymbol{H}}\ & \left\|\boldsymbol{X}^T - \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{H}\right\|_2^2 \\ s.t.\ & \boldsymbol{W},\boldsymbol{H}\geq 0, \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_l. \end{aligned} \tag{5}$$

where, $\boldsymbol{W}\in\Re^{d\times l}$ is the feature selection matrix, and $\boldsymbol{I}_l\in\Re^{l\times l}$ is an identity matrix. Each row or column of $\boldsymbol{W}$ contains no more than one non-zero element. It is therefore an indication of the feature

selection, i.e. $\boldsymbol{X}_I^T=\boldsymbol{X}^T\boldsymbol{W}$. The definition of $\boldsymbol{W}$ is as follows:

$$W_{i,j}=\begin{cases}1, & \text{the } j\text{th element of } I \text{ is } i,\\ 0, & \text{otherwise.}\end{cases} \tag{6}$$

In the objective function, the non-negative constraint and the orthogonal constraint make $\boldsymbol{W}$ more close to the matrix containing 0-1 elements in the learning process. However, in the actual learning process, this process is difficult. Therefore we introduce the sparse constraint to increase the sparsity of matrix $\boldsymbol{W}$.

We use $L_{2,1}$-*norm* to constrain $\boldsymbol{W}$. It can ensure the sparsity of $\boldsymbol{W}$ and help avoid trivial solutions, which makes the feature selection more discriminative. Thus, the objective function (5) can be rewritten as:

$$\begin{aligned}\arg \min_{\boldsymbol{W},\boldsymbol{H}} & \left\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{H}\right\|_2^2+\beta\left\|\boldsymbol{W}\right\|_{2,1}\\ s.t.\ & \boldsymbol{W},\boldsymbol{H}\geq 0,\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}_l.\end{aligned} \tag{7}$$

where $\left\|\boldsymbol{W}\right\|_{2,1}=\sum_{i=1}^{d}\left\|\boldsymbol{w}_i\right\|_2$, $\boldsymbol{w}_i$ is the i-th row of the matrix $\boldsymbol{W}$, β is balance parameter.

2.3 Local structure preserving

Some studies [44], [45], [46] show that the feature manifold contains underlying geometric structure information, which is very useful for improving the performance of the algorithm. We use spectral theory [46], [44] to preserve the local structure information on the feature manifold, which can improve the efficiency and accuracy of feature selection.

According to the method of constructing graphs in [44], we construct the nearest neighbor graph in feature space. We use a row vector $\boldsymbol{f}_i$ to represent the i-th feature of the sample matrix, so that the sample matrix $\boldsymbol{X}$ can be rewritten as $\boldsymbol{X}=[\boldsymbol{f}_1;\boldsymbol{f}_2;...;\boldsymbol{f}_d]\in\Re^{d\times n}$. We construct a k-nearest neighborhood graph $\boldsymbol{G}=(V,E)$, where V represents a set of feature points $\{\boldsymbol{f}_1,\boldsymbol{f}_2,...,\boldsymbol{f}_d\}$. E represents the weights of the edges between the vertices. It represents the similarity between the two features: the higher the weight, the more similar the features. In this paper, we adopt the Gaussian function as the weight measurement, and its definition is as follows:

$$[\boldsymbol{S}]_{ij}=\begin{cases}\exp(-\left\|\boldsymbol{f}_i-\boldsymbol{f}_j\right\|_2^2/\sigma^2), \text{if } \boldsymbol{f}_i\in N(\boldsymbol{f}_j) \\ \qquad\qquad\qquad\qquad \text{or } \boldsymbol{f}_j\in N(\boldsymbol{f}_i) \\ 0, \qquad\qquad\qquad\qquad \text{otherwise}\end{cases}. \tag{8}$$

where, i, j=1,2,...,d, $N(\boldsymbol{f}_i)$ represents the k-nearest neighbor set of feature $\boldsymbol{f}_i$, $[\boldsymbol{S}]_{ij}$ represents the similarity between $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$. σ is the Gaussian scale parameter. $\boldsymbol{L}$ is the graph Laplacian matrix of the feature space, $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{S}$, $\boldsymbol{D}$ is a diagonal matrix, and $\boldsymbol{D}_{ii}=\sum_j[\boldsymbol{S}]_{ij}$.

Denote $\boldsymbol{H}=[\boldsymbol{h}_1,\boldsymbol{h}_2,...,\boldsymbol{h}_d]\in\Re^{l\times d}$ as the coefficient matrix, the objection of local structure preserving is as follows:

$$\arg\ \min_{\boldsymbol{H}}\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\left\|\boldsymbol{h}_i-\boldsymbol{h}_j\right\|_2^2[\boldsymbol{S}]_{ij}=Tr(\boldsymbol{HLH}^T). \tag{9}$$

If $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ are very similar, $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ are also very close to each other. Therefore, we combine equations (7) and (9), and obtain a new objective function as follows:

$$\begin{aligned}\arg\ \min_{H,W} & Tr(\boldsymbol{HLH}^T)+\alpha\left\|\boldsymbol{X}^T-\boldsymbol{X}^T\boldsymbol{WH}\right\|_2^2+\beta\left\|\boldsymbol{W}\right\|_{2,1}+\frac{\lambda}{2}\left\|\boldsymbol{W}^T\boldsymbol{W}-\boldsymbol{I}_l\right\|_2^2 \\ & s.t.\ \boldsymbol{W}\geq 0,\ \boldsymbol{H}\geq 0.\end{aligned} \tag{10}$$

where, α, β, λ are balance parameters, α, β, $\lambda\geq 0$.

2.4 Feature evaluation

By optimizing the objective function of SGFS, we can obtain a feature selection matrix $\boldsymbol{W}$. Where $\boldsymbol{W}=[\boldsymbol{w}_1;\boldsymbol{w}_2;...;\boldsymbol{w}_d]$, Using $\left\|\boldsymbol{w}_i\right\|_2$ to calculate the value of each row of matrix $\boldsymbol{W}$, these values represent the importance of the corresponding features. The higher the value, the more important the feature.

We arrange all the evaluation values of features in descending order, obtaining an index I. According to the index I, we select the first l features to represent the original data set, generating a new data matrix $\boldsymbol{X}_{new}\in\Re^{l\times n}$. In this way, the feature selection process is completed.

2.5 Connection with MFFS

By observing the objective function (10), we can see that when α=1, β=0, and removing the local structure preserving regularization term, SGFS degenerates into MFFS. MFFS is mainly used to solve the following problem:

$$\begin{aligned} \arg \min_{H,W} & \left\| \boldsymbol{X}^T - \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{H} \right\|_2^2 + \frac{\lambda}{2} \left\| \boldsymbol{W}^T \boldsymbol{W} - \boldsymbol{I}_l \right\|_2^2 \\ & s.t. \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0. \end{aligned} \tag{11}$$

2.6 Update rules for SGFS

We use an alternating iterative method [44] to optimize the objective function (10). We first introduce two Lagrange multipliers ψ_{ij} and ϕ_{ij} to constraints $\boldsymbol{W}_{ij} \geq 0$ and $\boldsymbol{H}_{ij} \geq 0$ respectively. Therefore, formula (10) can be rewritten as Lagrange's function:

$$\begin{aligned} \mathrm{L}(\boldsymbol{W}, \boldsymbol{H}) = & Tr(\boldsymbol{H} \boldsymbol{L} \boldsymbol{H}^T) + \alpha \left\| \boldsymbol{X}^T - \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{H} \right\|_2^2 + \beta \left\| \boldsymbol{W} \right\|_{2,1} \\ & + \frac{\lambda}{2} \left\| \boldsymbol{W}^T \boldsymbol{W} - \boldsymbol{I}_l \right\|_2^2 + Tr(\psi \boldsymbol{W}^T) + Tr(\phi \boldsymbol{H}^T). \end{aligned} \tag{12}$$

First, we need to define a diagonal matrix $\boldsymbol{U} \in \Re^{d \times d}$. The i-th diagonal element of $\boldsymbol{U}$ is defined as follows:

$$\boldsymbol{U}_{ii} = \frac{1}{2 \left\| \boldsymbol{w}_i \right\|_2}. \tag{13}$$

On this basis, we introduce a small enough constant ε to avoid overflow, and get the following formula:

$$\boldsymbol{U}_{ii} = \frac{1}{2 \max(\left\| \boldsymbol{w}_i \right\|_2, \varepsilon)}. \tag{14}$$

Due to the definition of $\boldsymbol{U}$, we rewrite $\left\| \boldsymbol{W} \right\|_{2,1}$ as $Tr(\boldsymbol{W}^T \boldsymbol{U} \boldsymbol{W})$, so formula (12) can be rewritten as:

$$\begin{aligned} \mathrm{L}(\boldsymbol{W}, \boldsymbol{H}) = & Tr(\boldsymbol{H} \boldsymbol{L} \boldsymbol{H}^T) + \alpha Tr((\boldsymbol{X}^T - \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{H})(\boldsymbol{X}^T - \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{H})^T) + \beta Tr(\boldsymbol{W}^T \boldsymbol{U} \boldsymbol{W}) \\ & + \frac{\lambda}{2} Tr((\boldsymbol{W}^T \boldsymbol{W} - \boldsymbol{I}_l)(\boldsymbol{W}^T \boldsymbol{W} - \boldsymbol{I}_l)^T) + Tr(\psi \boldsymbol{W}^T) + Tr(\phi \boldsymbol{H}^T). \end{aligned} \tag{15}$$

Next, we fix $\boldsymbol{H}$ and $\boldsymbol{U}$, and update $\boldsymbol{W}$. By taking the partial derivative of formula (15) with respect to $\boldsymbol{W}$, we arrive at:

$$\frac{\partial \mathrm{L}}{\partial \boldsymbol{W}} = 2\alpha(\boldsymbol{X} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{H} \boldsymbol{H}^T - \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{H}^T) + 2\beta \boldsymbol{U} \boldsymbol{W} + 2\lambda \boldsymbol{W} \boldsymbol{W}^T \boldsymbol{W} - 2\lambda \boldsymbol{W} + \psi. \tag{16}$$

Using the KKT conditions [42] $\psi_{ij} \boldsymbol{W}_{ij} = 0$, we get:

$$[\alpha(\boldsymbol{X} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{H} \boldsymbol{H}^T - \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{H}^T) + \beta \boldsymbol{U} \boldsymbol{W} + \lambda \boldsymbol{W} \boldsymbol{W}^T \boldsymbol{W} - \lambda \boldsymbol{W}]_{ij} \boldsymbol{W}_{ij} = 0. \tag{17}$$

Hence, we get the iterative update rule for $\boldsymbol{W}$ as follows:

$$\boldsymbol{W}_{ij} \leftarrow \boldsymbol{W}_{ij} \frac{[\alpha \boldsymbol{X}\boldsymbol{X}^T \boldsymbol{H}^T + \lambda \boldsymbol{W}]_{ij}}{[\alpha \boldsymbol{X}\boldsymbol{X}^T \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T + \beta \boldsymbol{U}\boldsymbol{W} + \lambda \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{W}]_{ij}}. \tag{18}$$

Finally, we fix $\boldsymbol{W}$ and $\boldsymbol{U}$, and update $\boldsymbol{H}$. By taking the partial derivatives of formula (15) with respect to $\boldsymbol{H}$, we arrive at:

$$\frac{\partial \mathrm{L}}{\partial \boldsymbol{H}} = 2\alpha(\boldsymbol{W}^T \boldsymbol{X}\boldsymbol{X}^T \boldsymbol{W}\boldsymbol{H} - \boldsymbol{W}^T \boldsymbol{X}\boldsymbol{X}^T) + 2\boldsymbol{H}(\boldsymbol{D} - \boldsymbol{S}) + \phi. \tag{19}$$

Using the KKT conditions $\phi_{ij}\boldsymbol{H}_{ij}=0$, we get:

$$[\alpha(\boldsymbol{W}^T \boldsymbol{X}\boldsymbol{X}^T \boldsymbol{W}\boldsymbol{H} - \boldsymbol{W}^T \boldsymbol{X}\boldsymbol{X}^T) + \boldsymbol{H}(\boldsymbol{D} - \boldsymbol{S})]_{ij} \boldsymbol{H}_{ij} = 0. \tag{20}$$

We get the iterative update rule for $\boldsymbol{H}$ as follows:

$$\boldsymbol{H}_{ij} \leftarrow \boldsymbol{H}_{ij} \frac{[\alpha \boldsymbol{W}^T \boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{H}\boldsymbol{S}]_{ij}}{[\alpha \boldsymbol{W}^T \boldsymbol{X}\boldsymbol{X}^T \boldsymbol{W}\boldsymbol{H} + \boldsymbol{H}\boldsymbol{D}]_{ij}}. \tag{21}$$

Table 1 shows the procedure of SGFS.

Table 1 The procedure of SGFS

Input: Data matrix $\boldsymbol{X} \in \Re^{d \times n}$; neighborhood size k; balance parameter α, β, λ; Gaussian scale parameter σ; maximum number of iterations *NIter*; number of selected features l.

Output: Index of selected features I; new data matrix $\boldsymbol{X}_{new} \in \Re^{l \times n}$.

1. Construct the k-nearest neighborhood graph $\boldsymbol{G}=(V, E)$ in feature space.
2. Compute the similarity matrix $\boldsymbol{S}$, graph Laplacian matrix $\boldsymbol{L}$.
3. Initialize $\boldsymbol{W}$, $\boldsymbol{H}$, $\boldsymbol{U}$.
4. Update $\boldsymbol{W}$, $\boldsymbol{H}$, $\boldsymbol{U}$ according to the iterative update rules (14), (18) and (21), until the number of iterations is equal to *NIter*.
5. Compute the evaluation values for all the features according to $\|\boldsymbol{w}_i\|_2$, select the features corresponding to the largest l values and get a new data matrix $\boldsymbol{X}_{new} \in \Re^{l \times n}$.

2.7 Convergence analysis

Next, we analyze the convergence of SGFS. Based on the proof method in [54][32], we give the proof of the monotone property of formula (10) under the update rules (18) and (21).

First, we prove that formula (10) is nonincreasing under the update rule (21).

Definition 1. From Lee and Seung [54], if there is a function $G(k, k')$, which makes $F(k)$ satisfy the following conditions:

$$G(k,k') \geq F(k)\,,\; G(k,k) = F(k) \tag{22}$$

then $F(k)$ is monotonically decreasing under the following updating formula:

$$k^{(t+1)} = \arg\min_{h} G(k, k^{(t)}) \tag{23}$$

Here $G(k, k')$ is an auxiliary function for $F(k)$.

Proof. $F(k^{(t+1)}) \le G(k^{(t+1)}, k^{(t)}) \le G(k^{(t)}, k^{(t)}) = F(k^{(t)})$.

We only retain the items with $\boldsymbol{H}$ from the formula (10) and obtain the following function:

$$F(\boldsymbol{H}) = Tr(\boldsymbol{HLH}^T) + \alpha Tr(\boldsymbol{X}^T\boldsymbol{WHH}^T\boldsymbol{W}^T\boldsymbol{X} - 2\boldsymbol{X}^T\boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{X}). \tag{24}$$

Through taking the first-order and second-order partial derivatives of $F(\boldsymbol{H})$ with respect to $\boldsymbol{H}$, we arrive at:

$$F'_{ij} = \left[\frac{\partial F}{\partial \boldsymbol{H}}\right]_{ij} = [2\alpha(\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{WH} - \boldsymbol{W}^T\boldsymbol{XX}^T) + 2\boldsymbol{HL}]_{ij} \tag{25}$$

$$F''_{ij} = 2\alpha[\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{W}]_{ii} + 2[\boldsymbol{L}]_{jj} \tag{26}$$

Lemma 1. From Lee and Seung [54], the auxiliary function of $\boldsymbol{F}_{ij}$ is given as follows:

$$G(\boldsymbol{H}_{ij}, \boldsymbol{H}_{ij}^{(t)}) = F_{ij}(\boldsymbol{H}_{ij}^{(t)}) + F'_{ij}(\boldsymbol{H}_{ij}^{(t)})(\boldsymbol{H}_{ij} - \boldsymbol{H}_{ij}^{(t)}) + \frac{[\alpha\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{WH} + \boldsymbol{HD}]_{ij}}{\boldsymbol{H}_{ij}^{(t)}}(\boldsymbol{H}_{ij} - \boldsymbol{H}_{ij}^{(t)})^2 \tag{27}$$

Given the Taylor expansion of $F_{ij}(\boldsymbol{H}_{ij})$:

$$F_{ij}(\boldsymbol{H}_{ij}) = F_{ij}(\boldsymbol{H}_{ij}^{(t)}) + F'_{ij}(\boldsymbol{H}_{ij}^{(t)})(\boldsymbol{H}_{ij} - \boldsymbol{H}_{ij}^{(t)}) + \left\{\alpha[\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{W}]_{ii} + [\boldsymbol{L}]_{jj}\right\}(\boldsymbol{H}_{ij} - \boldsymbol{H}_{ij}^{(t)})^2 \tag{28}$$

According to formula (27), we know that $G(\boldsymbol{H}_{ij}, \boldsymbol{H}_{ij}^{(t)}) \ge F_{ij}(\boldsymbol{H}_{ij})$ is equivalent to

$$\frac{[\alpha\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{WH} + \boldsymbol{HD}]_{ij}}{\boldsymbol{H}_{ij}^{(t)}} \ge \alpha[\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{W}]_{ii} + [\boldsymbol{L}]_{jj} \tag{29}$$

Clearly, it can be seen that $[\alpha\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{WH}]_{ij} = \sum_{l=1}^{d}[\alpha\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{W}]_{il}\boldsymbol{H}_{lj}^{(t)} \ge \alpha[\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{W}]_{ii}\boldsymbol{H}_{ij}^{(t)}$ and $[\boldsymbol{HD}]_{ij} = \sum_{l=1}^{d}\boldsymbol{H}_{il}^{(t)}[\boldsymbol{D}]_{lj} \ge \boldsymbol{H}_{ij}^{(t)}\boldsymbol{D}_{jj} \ge \boldsymbol{H}_{ij}^{(t)}[\boldsymbol{D} - \boldsymbol{W}]_{jj} = \boldsymbol{H}_{ij}^{(t)}[\boldsymbol{L}]_{jj}$. Thus, inequality (29) holds, i.e. $G(\boldsymbol{H}_{ij}, \boldsymbol{H}_{ij}^{(t)}) \ge F_{ij}(\boldsymbol{H}_{ij})$ holds, $G(\boldsymbol{H}_{ij}, \boldsymbol{H}_{ij}) = F_{ij}(\boldsymbol{H}_{ij})$ also holds.

Proof. The variable $\boldsymbol{H}$ satisfies the updating formula (23) that makes the F_{ij} monotonically decreasing.

By substituting $G(\boldsymbol{H}_{ij}, \boldsymbol{H}_{ij}^{(t)})$ in (27) into (23), we get the following formula:

$$\boldsymbol{H}_{ij}^{(t+1)} = \boldsymbol{H}_{ij}^{(t)} - \boldsymbol{H}_{ij}^{(t)}\frac{F'_{ij}(\boldsymbol{H}_{ij}^{(t)})}{2[\alpha\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{WH} + \boldsymbol{HD}]_{ij}} = \boldsymbol{H}_{ij}^{(t)}\frac{[\alpha\boldsymbol{W}^T\boldsymbol{XX}^T + \boldsymbol{HS}]_{ij}}{[\alpha\boldsymbol{W}^T\boldsymbol{XX}^T\boldsymbol{WH} + \boldsymbol{HD}]_{ij}}$$

We can see that the above formula is the update rule (21). Therefore F_{ij} is non-increasing under the update rule (21).

The proof of the monotone property of formula (10) under the update rule (18) is similar to the above proof, and we can prove that F_{ij} is non-increasing under the update rule (18). Therefore, we conclude that formula (10) is also non-increasing under the update rules (18) and (21).

3. Experiments

In this section, we present the results and analysis of empirical experiments. We apply the proposed algorithm, and five comparison algorithms, to twelve public benchmark data sets, and the experimental results are analyzed. We use a clustering algorithm to cluster the feature selection results, and the clustering results are used as criteria for evaluating the performance of the feature selection algorithms. In this paper, we use *k-means* [55] as the clustering algorithm. In addition, we also analyze the parameter sensitivity of SGFS.

3.1 Data sets

In this experiment, we used twelve datasets, including digital image[1], text image[1], face image[1] and biological data[1] [32], [56]. The detailed information of the datasets is shown in table 2.

Table 2 The information of twelve datasets

Data set	Size	Dim	Classes	Type
Usps	9298	256	10	Digital image
Lung_dis	73	325	7	Biological
Isolet	1560	617	26	Letter image
Umist	575	644	20	Face image
COIL20	1440	1024	20	Digital image
AR10P	130	2400	10	Face image
Lung	203	3312	5	Biological
Yale64	165	4096	15	Face image
Orl64	400	4096	40	Face image
TOX_171	171	5748	4	Biological
Orlraws	100	10304	10	Face image
AT&T	400	10304	40	Face image

[1] http://featureselection.asu.edu/datasets.php

3.2 Comparison algorithms

To verify the effectiveness of SGFS, we compare its performance against five feature selection algorithms, as follows:

1. Baseline: all the features of the data set are selected.

2. LapScor: Laplacian Score [21] is a classic unsupervised feature selection algorithm, it is characterized by being simple and fast, but it lacks an effective learning mechanism.

3. MCFS: unsupervised feature selection for multicluster data [51] uses a spectral embedding learning and sparse regression feature selection framework. It uses a two step strategy, and mainly solves the following problems:

$$\begin{aligned} &\arg \min_{\boldsymbol{S}\boldsymbol{S}^T=\boldsymbol{I}_m} Tr(\boldsymbol{S}\boldsymbol{L}\boldsymbol{S}^T) \\ &\arg \min_{\boldsymbol{P}} \left\|\boldsymbol{P}^T\boldsymbol{X}-\boldsymbol{S}\right\|_2^2+\alpha\left\|\boldsymbol{P}\right\|_1. \end{aligned} \tag{30}$$

4. UDFS: unsupervised discriminant feature selection algorithm [57] uses the discriminant information and feature correlation, which aims to find the most discriminative features. Its objective function is as follows:

$$\arg \min_{\boldsymbol{P}^T\boldsymbol{P}=\boldsymbol{I}_m} Tr(\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{P})+\alpha\left\|\boldsymbol{P}\right\|_{2,1}. \tag{31}$$

5. MFFS: subspace learning for unsupervised feature selection via matrix factorization [32] uses a matrix decomposition technique to obtain a feature selection matrix. According to this matrix, a representative feature subspace can be found in the original feature space.

3.3 Evaluation metrics

By evaluating the clustering results of different feature selection algorithms, their performance can be compared. In this paper, we use Clustering Accuracy (ACC) [46][58][59] and Normalized Mutual Information (NMI) [46][60] to evaluate the clustering results. The higher the values of ACC and NMI, the better the performance of the algorithm.

NMI is defined as:

$$\mathrm{NMI}(R,T)=\frac{\mathrm{I}(R,T)}{\sqrt{\mathrm{H}(R)\mathrm{H}(T)}} \tag{32}$$

where R and T are two arbitrary variables, and $\mathrm{I}(R,T)$ is the mutual information between R and T.

H(R) and H(T) are the information entropies of R and T respectively. When NMI is used to evaluate the clustering results, R and T represent the clustering label and the ground truth label respectively.

ACC is defined as:

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

where n represents the total number of samples, a_i and b_i are the clustering label and the ground truth label for the sample $\mathbf{x}_i$ respectively. $\delta(x, y)=1$, if $x=y$; $\delta(x, y)=1$, otherwise. map(·) is an optimal mapping function which uses Hungarian [61] to match the clustering label and the ground truth label.

3.4 Experimental results and analysis

3.4.1 Experimental setting

Before the experiment, we set up the parameters of the algorithm. For all datasets, the number of selected features l are tuned from {20, 30, 40, 50, 60, 70, 80, 90, 100}. For LapScor, MCFS, UDFS and SGFS, the nearest neighborhood parameter k is set to 5, the Gaussian scale parameter σ is set to 10. For MCFS, UDFS, MFFS and SGFS, we set the maximum number of iterations to 30. For SGFS, the balance parameter α and β are searched in the range of $\{10^{-7}, 10^{-6}, ..., 10^{+7}\}$, the range of parameter λ is set to $\{10^{-6}, 10^{-5}, 10^{-3}, 10^{-2}, 10^{+4}, 10^{+7}, 10^{+8}\}$. On each dataset, the selection of parameters is relatively stable. By adjusting the balance parameters α, β, λ, we obtain the maximum value of ACC and NMI. We independently run 20 times, and take the average of these results as the final result. In the experiment, we used a computer with 4G memory and 2.3GHz frequency, and used Matlab as a simulation software.

3.4.2 Convergence test

First, we verify the convergence of the objective function of SGFS. We let SGFS iterate 30 times on twelve datasets, and recorded the value of the objective function after each iteration. The results are shown in **Fig. 1**.

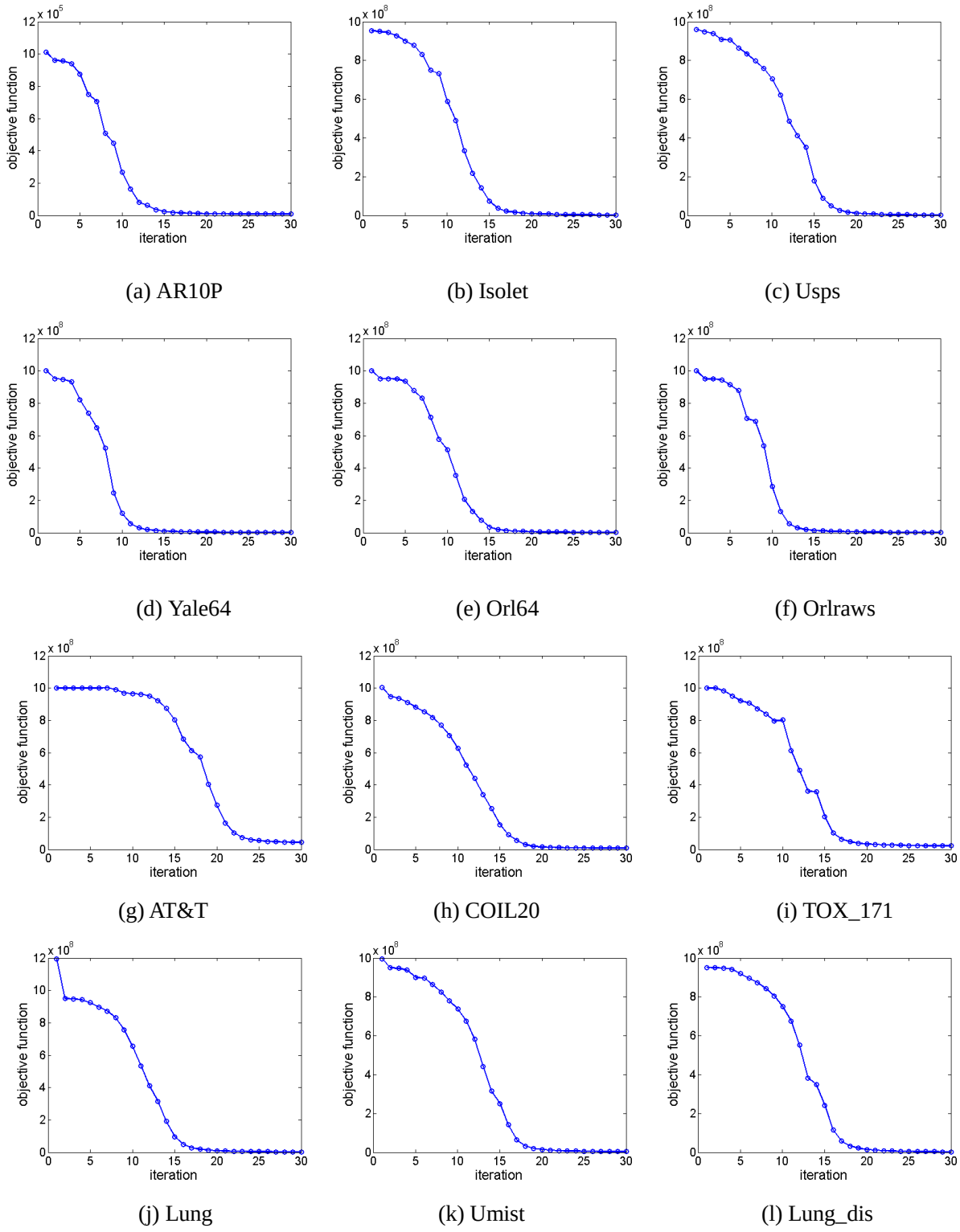


Fig. 1. Convergence of the objective function on twelve data sets.

In **Fig. 1**, the horizontal and vertical axes represent the number of iterations and the value of the objective function respectively. We can see that the value of the objective function decreases with increasing number of iterations. This is consistent with our previous proof of convergence. Also note that the value of the objective function tends to become stable within 20 iterations on most of the

datasets, which suggests that the objective function rapidly converges to a stable value.

In order to prove that the proposed algorithm is faster and more accurate than MFFS in the learning of the feature selection matrix and coefficient matrix, we analyze the number of iterations and the time required for the convergence of each objective function. Considering that the time spent in each iteration of SGFS and MFFS is almost the same, we only analyze the number of iterations required to achieve convergence of the objective function. For MFFS and SGFS, we set the maximum number of iterations to 30, the number of selected features l is fixed to 20, the range of parameter λ is set to $\{10^{-6}, 10^{-5}, 10^{-3}, 10^{-2}, 10^{+3}, 10^{+4}, 10^{+5}, 10^{+6}, 10^{+7}, 10^{+8}\}$. For SGFS, the balance parameters α and β are searched in the range of $\{10^{-7}, 10^{-6}, ..., 10^{+7}\}$. The comparison results are given in Table 3.

Table 3 The number of iterations required for the convergence of the objective function

Dataset	AR10P	Isolet	Usps	Yale64	Orl64	Orlraws
MFFS	25	25	24	28	25	29
SGFS	16	19	20	14	17	15
Dataset	AT&T	COIL20	TOX_171	Umist	Lung	Lung_dis
MFFS	27	27	22	24	25	23
SGFS	25	19	20	17	20	20

From Table 3, we can see that, for the objective function to converge, SGFS always requires significantly less iterations than MFFS. This suggests that SGFS can learn appropriate feature selection and coefficient matrices more quickly.

3.4.3 AT&T face dataset example

We randomly selected two different samples from the AT&T face dataset as the experimental samples, and apply the proposed algorithm to these two samples. In this experiment, the two samples were selected from the third class and the sixth class, they all contain 10304 features. The number of selected features l are tuned from {1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240}. With the increase in the number of selected features, the extracted face information is also increased. We use white to represent the feature which is not selected, and the selected feature retains its original gray. The experimental results are shown in **Fig. 2.**

Fig. 2. Results of two AT&T samples under different number of selected features.

From **Fig. 2**, we can see how SGFS selects different numbers of features from the two samples. From left to right, as the number of selected features gradually increases, the reconstructed image becomes increasingly clear. We can see that a convincing outline of the human face can be seen by selecting only a very small number of features. The reason is that the most representative face features are preferentially selected, such as the eyes, nose, mouth and chin. This preferential selection of the most salient features illustrates the strong performance of the proposed algorithm for the task of feature selection.

3.4.4 Experimental results and analysis

In Table 4 and Table 5, we show the ACC and NMI values of the six algorithms on all data sets. In these results, the best ACC and NMI values are marked in bold black, and the second best ACC and NMI values are marked in underlined.

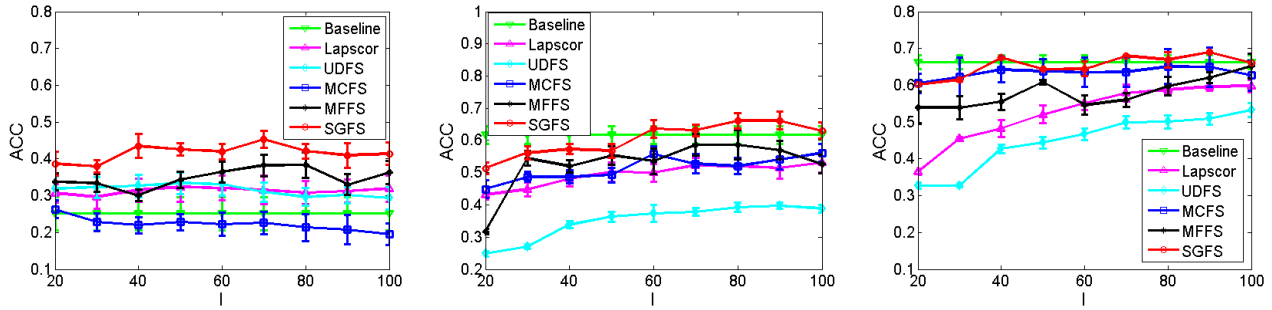
Table 4 Clustering accuracy of six algorithms on twelve datasets (ACC±STD%)

Dataset	Baseline	LapScor	UDFS	MCFS	MFFS	SGFS
AR10P	25.12±4.47	32.42±4.00	33.38±2.96	26.31±2.45	38.31±3.40	**45.38**±2.34
Isolet	61.73±2.77	53.17±3.11	39.66±0.94	56.11±2.70	58.68±4.60	**66.20**±2.67
Usps	66.26±1.93	59.79±1.62	53.30±1.91	65.06±4.75	65.25±3.30	**68.97**±0.17
Yale64	48.15±4.10	45.09±4.15	38.18±2.59	44.70±3.01	50.52±3.87	**51.70**±2.56
Orl64	53.14±3.33	42.04±1.84	47.71±2.38	50.59±2.96	38.36±1.85	**53.95**±3.08
Orlraws	73.65±7.06	64.45±5.97	64.45±5.97	75.15±3.95	72.80±8.46	**81.40**±5.15
AT&T	**60.96**±3.30	47.31±1.83	54.46±2.05	55.88±1.86	53.46±2.73	59.65±2.89
COIL20	**65.75**±4.16	60.94±2.17	63.38±3.04	62.96±2.82	62.00±2.40	65.61±2.67
TOX_171	43.13±2.35	41.84±1.98	41.35±0.57	40.18±4.35	38.45±1.31	**44.82**±1.36
Umist	42.97±2.21	42.09±1.94	44.83±1.52	47.24±2.25	44.07±2.38	**52.36**±2.61
Lung	70.10±8.22	66.67±2.12	55.44±4.75	71.75±2.04	62.82±4.74	**84.70**±0.87
Lung_dis	73.63±5.26	62.81±5.59	70.07±5.83	74.25±4.84	74.52±4.86	**81.03**±4.34

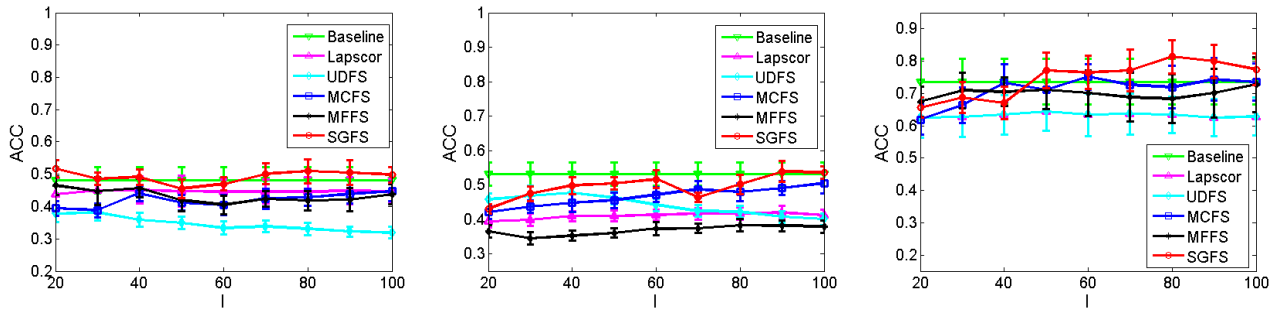
Table 5 Normalized Mutual Information of six algorithms on twelve datasets (NMI±STD%)

Dataset	Baseline	LapScor	UDFS	MCFS	MFFS	SGFS
AR10P	21.42±5.62	32.81±2.42	32.00±3.22	22.95±2.93	39.63±2.89	**47.98**±2.60
Isolet	76.06±1.26	68.40±1.34	52.90±0.79	69.92±0.96	72.42±2.12	**76.78**±0.98
Usps	61.13±0.86	56.53±0.70	48.01±1.28	58.89±2.03	60.02±1.72	**62.23**±0.14
Yale64	54.79±3.39	51.46±2.58	44.95±1.79	50.11±2.23	**56.17**±4.77	55.20±1.88
Orl64	**73.56**±1.50	63.71±0.96	68.68±1.64	70.94±1.58	61.55±1.03	73.41±0.97
Orlraws	79.87±5.31	72.64±3.80	72.64±3.80	82.52±3.07	81.44±6.57	**83.54**±3.56
AT&T	**79.96**±1.37	71.04±0.93	73.90±1.32	74.52±1.02	73.73±1.21	76.83±1.27
COIL20	**76.69**±1.99	69.94±2.05	72.43±1.31	73.94±1.44	71.93±2.19	74.76±1.49
TOX_171	**14.55**±2.64	11.03±1.14	10.12±2.03	10.20±1.98	11.31±0.13	12.45±191
Umist	64.83±2.04	62.33±1.91	58.31±1.47	67.06±1.38	62.59±1.87	**68.91**±1.40
Lung	54.47±2.84	47.17±1.94	38.01±1.87	52.52±0.63	44.53±1.14	**58.23**±1.01
Lung_dis	69.27±4.21	59.76±3.77	65.02±4.12	70.30±3.72	65.88±3.77	**72.96**±3.64

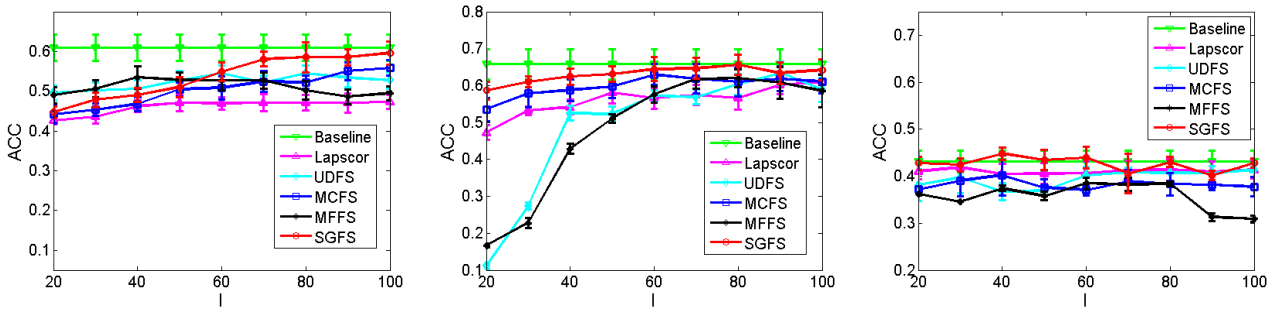
From Table 4 and Table 5, we can see that most of the results of SGFS on 12 datasets are better than the results of five comparison algorithms. In addition, the results of SGFS are even better than the results of Baseline on some datasets. **Fig. 3** shows the clustering accuracy of SGFS and five comparison algorithms on twelve datasets with different number of selected features. The horizontal axis represents the number of selected features *l*, the vertical axis represents clustering accuracy (ACC) and standard deviation (STD).



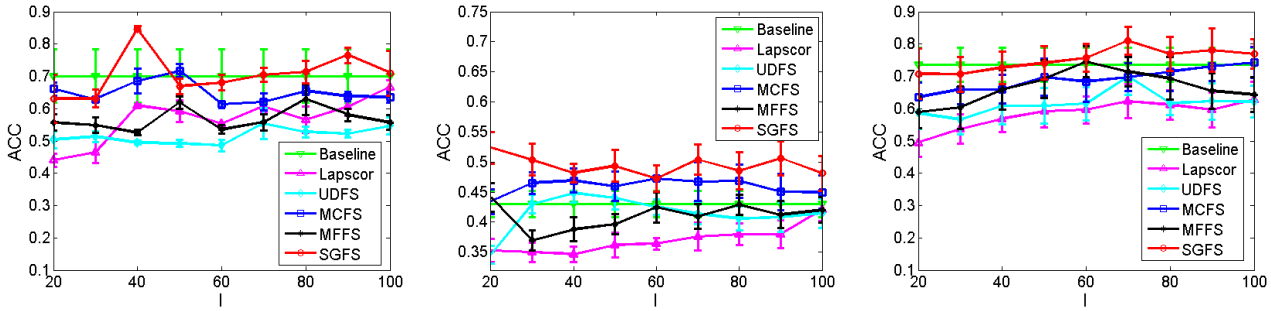
(a) AR10P (b) Isolet (c) Usps



(d) Yale64 (e) Orl64 (f) Orlraws



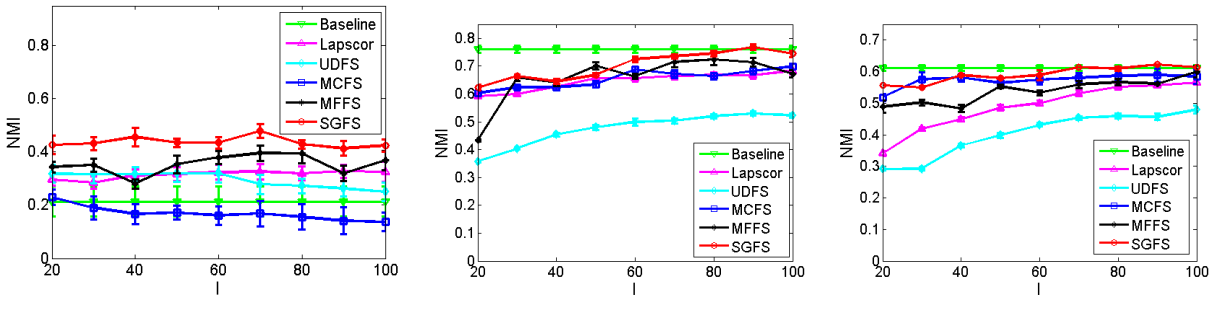
(g) AT&T (h) COIL20 (i) TOX_171



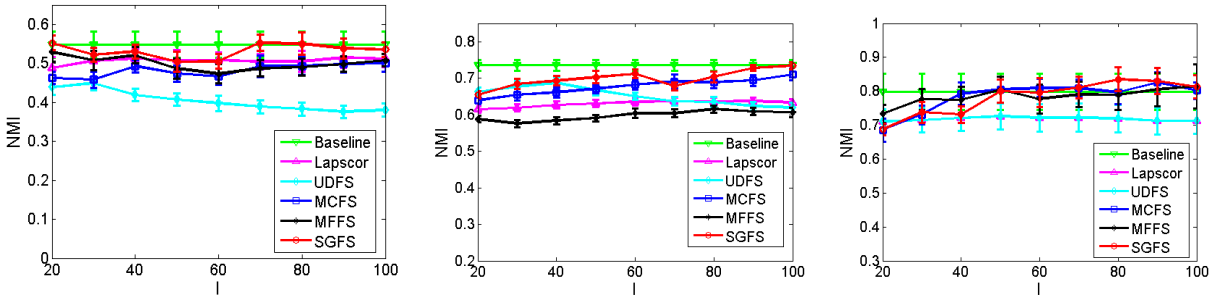
(j) Lung (k) Umist (l) Lung_dis

Fig. 3. Clustering accuracy of six algorithms on twelve datasets with different number of selected features.

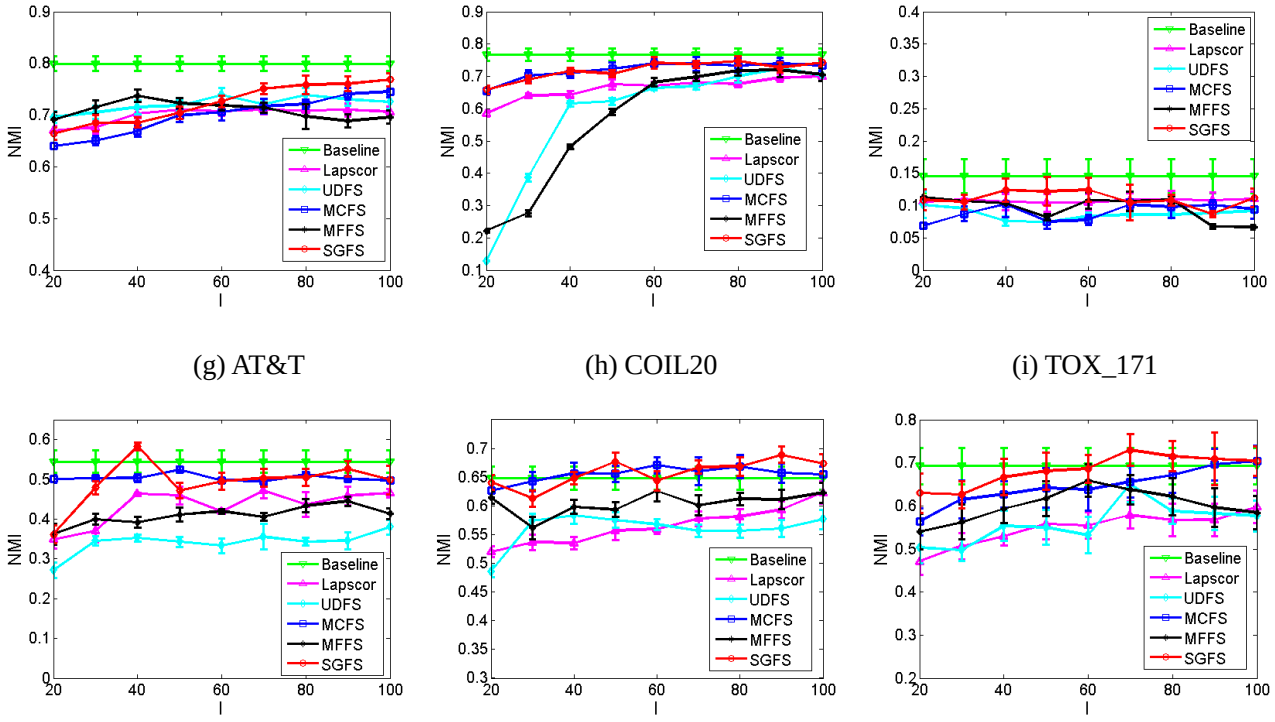
From **Fig. 3**, we can see that the ACC of SGFS are higher than that of four comparison algorithms on all datasets, and in some cases even higher than the results of Baseline. It is worth mentioning that the results of SGFS are better than all the comparison algorithms on the AR10P dataset, which suggests that SGFS improves the clustering accuracy.



(a) AR10P (b) Isolet (c) Usps



(d) Yale64 (e) Orl64 (f) Orlraws



(g) AT&T (h) COIL20 (i) TOX_171

(j) Lung (k) Umist (l) Lung_dis

Fig. 4. Normalized Mutual Information of six algorithms on twelve datasets with different number of selected features.

Fig.4 shows the normalized mutual information of SGFS and five comparison algorithms on twelve datasets with different numbers of selected features. The horizontal coordinate also represents the number of selected features *l*, while the vertical coordinate represents normalized mutual information (NMI) and standard deviation (STD). We also can see that the NMI of SGFS are higher than that of four comparison algorithms on all datasets. On datasets Isolet, Usps, Yale64, Orl64 and Lung, only SGFS has results better than Baseline. It also shows that SGFS is more effective than the comparison algorithms.

3.4.5 Robustness test of algorithms

In order to verify the robustness of the proposed algorithm against noise, we artificially corrupted six datasets by adding Gaussian noise of various different magnitudes. In this experiment, we set the variance of the added Gaussian noise to 10, 20, 30. In Table 6 and Table 7, we show the ACC and NMI values of MFFS and SGFS on six test datasets.

Table 6 Clustering accuracy of SGFS and MFFS on six datasets with different variance of the Gaussian noise (ACC±STD%)

Variance	10		20		30	
Dataset	MFFS	SGFS	MFFS	SGFS	MFFS	SGFS
AR10P	35.38±2.79	45.62±4.07	35.04±2.82	43.35±3.39	35.46±3.61	41.04±3.08
Orl64	36.47±1.31	52.14±2.96	36.01±1.32	52.41±2.72	36.43±2.04	53.12±1.80
AT&T	53.81±2.48	58.80±2.81	53.31±1.77	57.98±2.26	53.04±2.46	58.27±2.72
TOX_171	34.50±0.01	42.98±1.55	34.50±0.01	43.04±1.51	34.50±0.01	43.01±1.52
Umist	42.62±2.72	48.49±2.41	42.59±2.20	47.99±3.05	41.62±2.13	46.15±2.97
Lung_dis	69.59±7.16	77.19±3.78	68.08±6.92	76.51±6.79	68.22±6.87	76.16±6.61

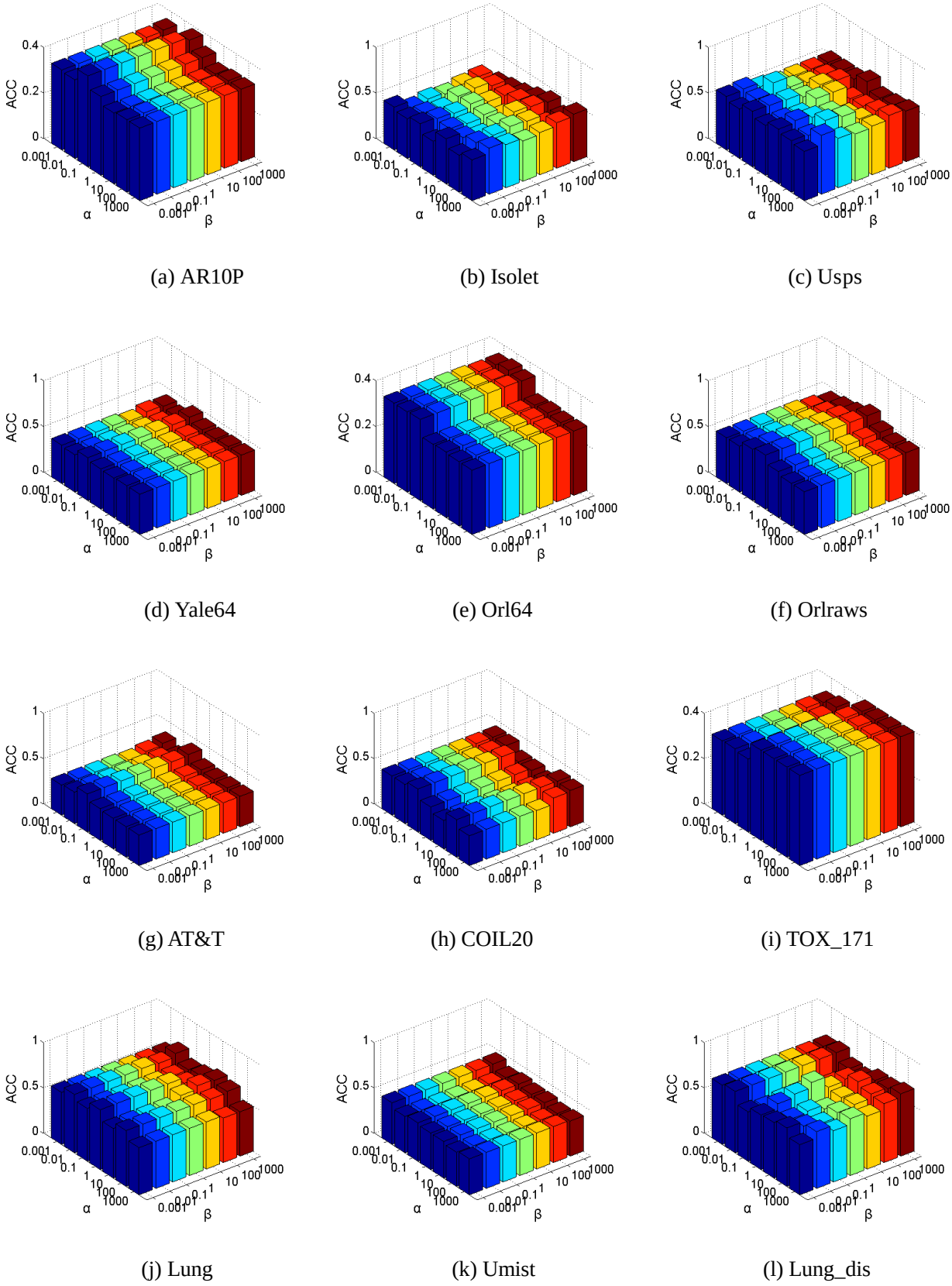
Table 7 Normalized Mutual Information of SGFS and MFFS on six datasets with different variance of the Gaussian noise (NMI±STD%)

Variance	10		20		30	
Dataset	MFFS	SGFS	MFFS	SGFS	MFFS	SGFS
AR10P	37.75±2.97	46.75±4.18	37.07±2.94	43.83±3.65	37.73±3.36	43.51±3.82
Orl64	59.56±1.31	72.35±1.40	59.01±1.11	71.68±1.50	59.11±1.22	72.25±0.75
AT&T	72.48±1.48	76.16±1.26	72.08±1.19	75.60±1.39	71.95±1.29	75.79±1.58
TOX_171	7.96±0.01	10.83±1.68	7.96±0.01	10.84±1.67	7.96±0.01	10.84±1.67
Umist	62.31±1.69	67.55±1.65	62.11±1.18	66.46±1.71	61.11±1.67	64.88±1.34
Lung_dis	63.59±5.56	70.22±3.38	62.38±4.46	71.32±4.05	62.16±4.64	69.89±5.18

From Table 6 and Table 7, we can see that the performance of the proposed algorithm is affected very little by the added Gaussian noise. This suggests that the proposed algorithm is highly robust to noise. Furthermore, on each test data set, the results of SGFS are significantly better than those of MFFS.

3.4.6 Parameter sensitivity analysis

The parameters of SGFS mainly include: the neighborhood parameter k, the Gaussian scale parameter σ, and the balance parameters α, β, λ. In this paper, we only discuss the sensitivity of parameters α and β, since the other parameters are relatively stable. We explore the parameters α and β in the range of $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{+1}, 10^{+2}, 10^{+3}\}$, and record the clustering accuracy (ACC) and normalized mutual information (NMI) under different parameter combinations. We draw the results into 3-D diagrams, as shown in **Fig. 5** and **Fig 6**.



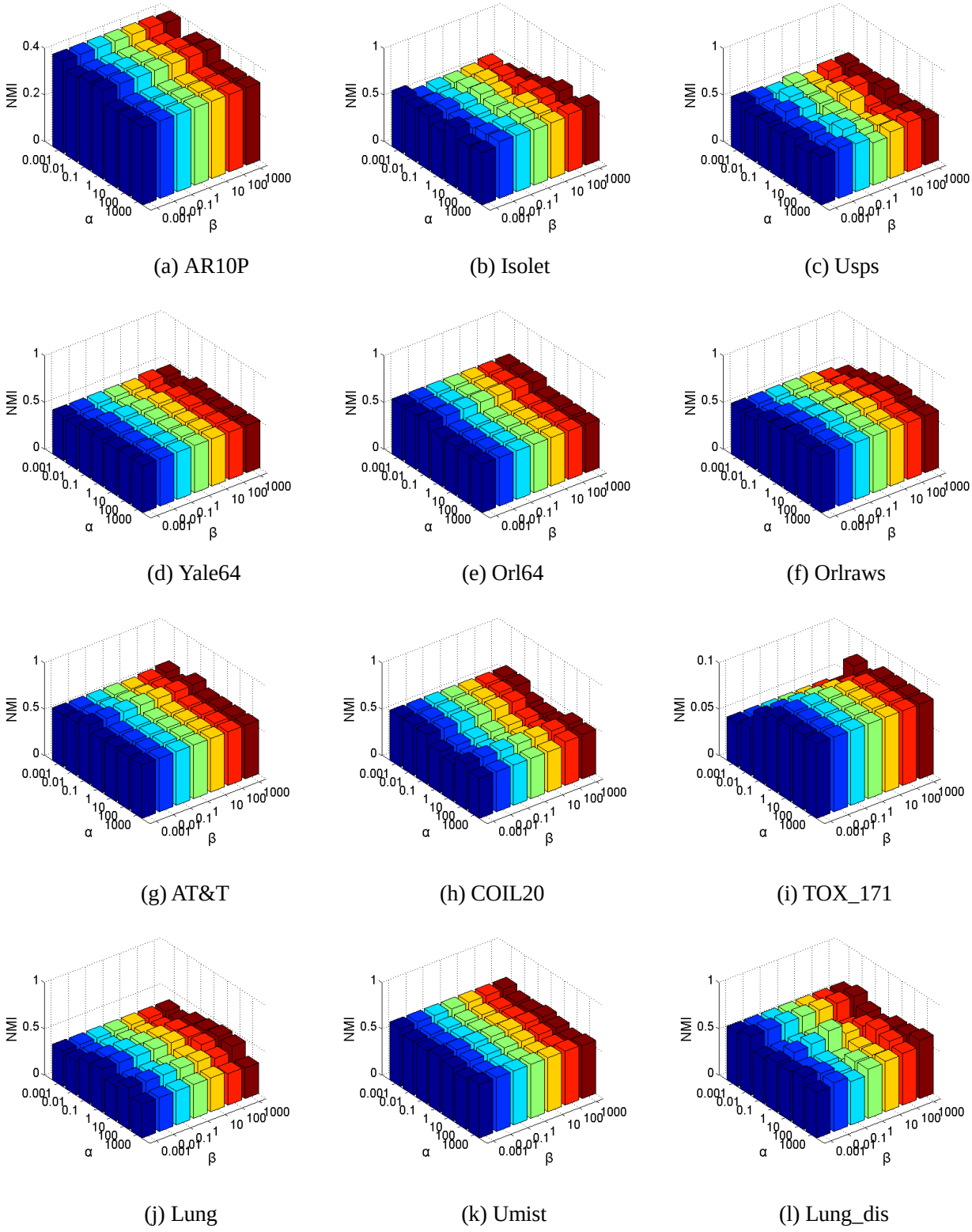
(a) AR10P (b) Isolet (c) Usps

(d) Yale64 (e) Orl64 (f) Orlraws

(g) AT&T (h) COIL20 (i) TOX_171

(j) Lung (k) Umist (l) Lung_dis

Fig. 5. Clustering accuracy of SGFS on twelve datasets under different parameter combinations.



(a) AR10P (b) Isolet (c) Usps

(d) Yale64 (e) Orl64 (f) Orlraws

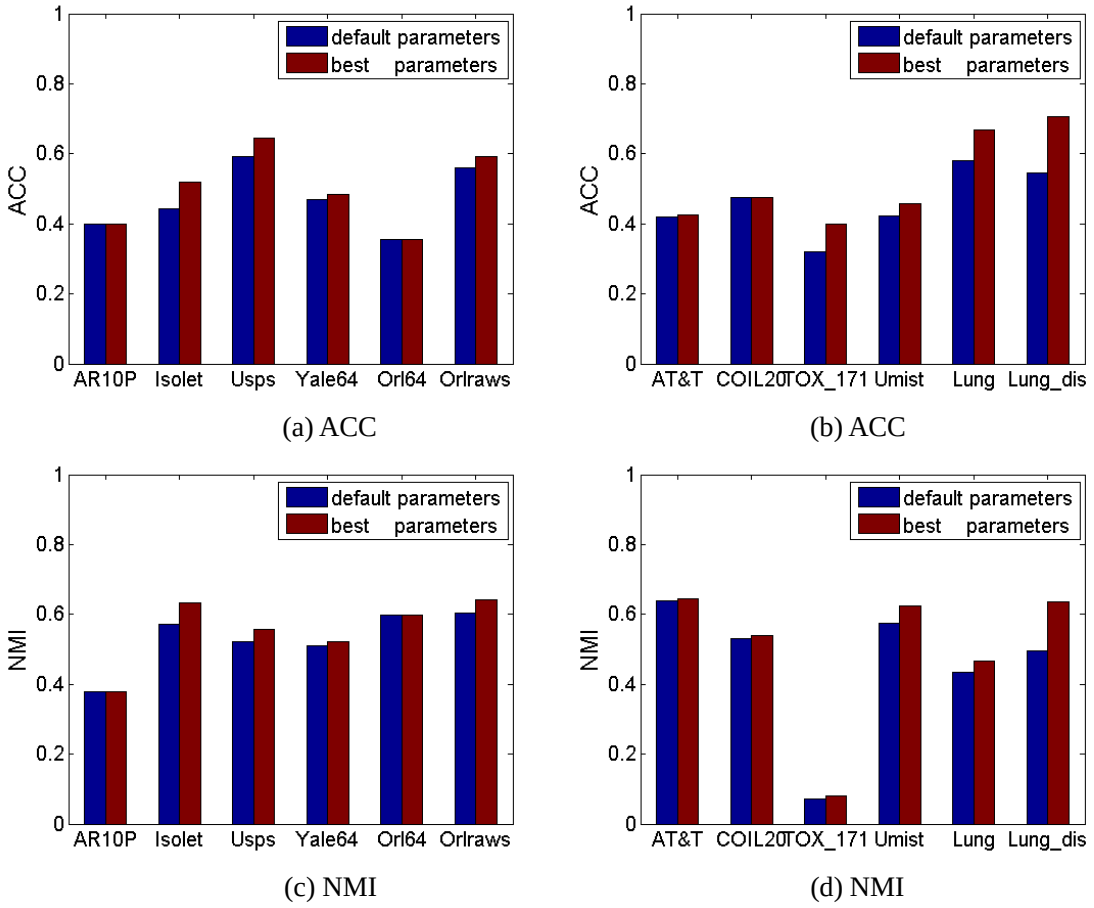
(g) AT&T (h) COIL20 (i) TOX_171

(j) Lung (k) Umist (l) Lung_dis

Fig. 6. Normalized Mutual Information of SGFS on twelve datasets under different parameter combinations.

From **Fig.5** and **Fig.6**, we can see that the parameters α and β on most datasets are relatively stable, especially on datasets Yale64, AT&T and Umist. This suggests that the proposed algorithm is not sensitive to the values of parameters α and β.

To further analyze the parameter sensitivity of the proposed algorithm, we compare the clustering results of SGFS under the default parameters and the best parameters. We set the default parameters α and β to a fixed value of 0.1. The best parameters are searched in the range of α and β. The comparison results are given in **Fig. 7**.



(a) ACC (b) ACC

(c) NMI (d) NMI

Fig. 7. Clustering results under the default parameters and the best parameters (default α=0.1, β=0.1).

Fig. 7 shows that the clustering results on all datasets under the default parameters and the best parameters lead to only small differences in performance. On datasets AR10P and Orl64, the performance that results from default parameters and the best parameters is the same. These results suggest that overall performance is not strongly dependent on the choice of parameter values. Good performance can be achieved with arbitrary parameter values.

4. Conclusions

In this paper, we have proposed a novel algorithm, called subspace learning-based graph regularized feature selection (SGFS). The proposed algorithm is based on the framework of subspace

learning feature selection via matrix factorization (MFFS). We have extended this approach, by first introducing an $L_{2,1}$-*norm* sparse constraint, which ensures the sparsity of the feature selection matrix $\boldsymbol{W}$ and increases the discriminating ability of the selected features. Furthermore, we have shown how the idea of graph regularization can be incorporated into feature selection, to preserve the local structure information of the feature space. This structure is then used to guide the learning of the coefficient matrix $\boldsymbol{H}$ and the feature selection matrix $\boldsymbol{W}$. We have presented a variety of experimental results, which suggest that SGFS outperforms several well known comparison algorithms from the literature.

A deficiency of the proposed algorithm is that the alternative iterative optimization method can sometimes be prone to converging on local optima. In future work, we hope to incorporate a mechanism for optimizing $\boldsymbol{W}$ and $\boldsymbol{H}$ simultaneously to obtain the globally optimal solution. We also intend to apply the idea of dual-graph regularization to the feature selection framework, and use the geometric structure information of the data space and the feature space simultaneously.

Acknowledgement

We would like to express our sincere appreciation to the editors and the anonymous reviewers for their insightful comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Natural Science Foundation of China, under Grants 61371201, the National Basic Research Program (973 Program) of China under Grant 2013CB329402, the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT_15R53.

References

[1] H. Yan, J. Yang, Sparse discrimiminative feature selection, Pattern Recogn. 48 (5) (2015) 1827-1835.

[2] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint $L_{2,1}$-norms minimization, in: Advances in Neural Information Processing Systems, 2010.

[3] I. Koprinska, M. Rana, V. G. Agelidis, Correlation and instance based feature selection for electricity load forecasting, Knowl.-Based Syst. 82 (2015) 29-40.

[4] A. G. Wang, N. An, G. L. Chen, L. Li, G. Alterovitz, Accelerating wrapper-based feature

selection with k-nearest-neighbor, Knowl.-Based Syst. 83 (2015) 81-91.

[5] Y. J. Li, H. X. Guo, X. Liu, Y. N. Li, J. L. Li, Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data, Knowl.-Based Syst. 94 (2016) 88-104.

[6] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, Incremental Support Vector Learning for Ordinal Regression, IEEE Transactions on Neural Networks and Learning Systems, 26(7) (2015): 1403-1416.

[7] B. Gu and V. S. Sheng, A Robust Regularization Path Algorithm for ν-Support Vector Classification, IEEE Transactions on Neural Networks and Learning Systems, DOI : 10.1109/TNNLS.2016.2527796, 2016

[8] Z. J. Chen, C. Z. Wu, Y. S. Zhang, Z. Huang, B. Ren, Feature selection with redundancy complementariness dispersion, Knowl.-Based Syst. 89 (2015) 203-217.

[9] P. Moradi, M. Rostami, Integration of graph clustering with ant colony optimization for feature selection, Knowl.-Based Syst. 84 (2015) 144-161.

[10] F. Nie, S. Xiang, Y. Liu, C. Hou, C. Zhang, Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction, Pattern Recogn. Lett. 33 (5) (2012) 485-491.

[11] B. Gu, X. M. Sun, and V. S. Sheng, Structural Minimax Probability Machine, IEEE Transactions on Neural Networks and Learning Systems, DOI : 10.1109/ TNNLS.2016. 2544779, 2016

[12] 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.

[13] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, Incremental learning for ν-Support Vector Regression, Neural Networks, 67(2015):140-150.

[14] M. Sikonja, I. Kononenko, Theoretical and empirical analysis of relief and relieff, Mach Learn. 53 (1-2) (2003) 23-69.

[15] Z. Xu, I. King, M. R.-T. Lyu R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Netw. 21 (7) (2010) 1033-1047.

[16] J. G. Dy, C. E. Brodley, Feature selection for unsupervised learning, J. Mach Learn. Res. 5 (2004) 845-889.

[17] M. Banerjee, N. R. Pal, Unsupervised Feature Selection with Controlled Redundancy (UFeSCoR), IEEE Trans. Knowl. Data Eng. 27 (12) (2015) 3390-3403.

[18] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings International Conference on Machine Learning, 2007, pp. 1151-1157.

[19] I. Guyon, A. Elisseeffi, An introduction to variable and feature selection, J. Mach Learn. Res. 3 (2003) 1157-1182.

[20] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1-2) (1997) 273-324.

[21] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, 2005, pp. 507-514.

[22] C. Hou, F. Nie, D. Yi, Y. Wu, Feature selection via joint embedding learning and sparse regression, in IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 2011, pp. 1324-1329.

[23] V. Vapnik, Statistical Learning Theory, Wiley, New York, NY, USA, 1998.

[24] Y. W. Wang, Y. N. Liu, L. Z. Feng, X. D. Zhu, Novel feature selection method based on harmony search for email classification, Knowl.-Based Syst. 73 (2015) 311-323.

[25] L. G. Zhou, D. Lu, H. Fujita, The performance of corporate financial distress prediction models with feature selection guided by domain knowledge and data mining approaches, Knowl.-Based Syst. 85 (2015) 52-61.

[26] Q. F. Zhou, H. Zhou, T. Li, Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features, Knowl.-Based Syst. 95 (2016) 1-11.

[27] P. Paatero, U. Tapper, Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, Environmetrics. 5 (2) (1994) 111-126.

[28] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature. 401 (1999) 788-791.

[29] I. Jolliffe, Principle Component Analysis, Springer, 1986.

[30] S. Lipovetsky, PCA and SVD with nonnegative loadings, Pattern Recogn. 42 (1) (2009) 68-76.

[31] G. Golub, C. Reinsch, Singular value decomposition and least squares solutions, Numer. Math. 14 (5) (1970) 403-420.

[32] S. Wang, W. Pedrycz, Q. Zhu, W. Zhu, Subspace learning for unsupervised feature selection via matrix factorization, Pattern Recogn. 48 (1) (2015) 10-19.

[33] S. Wang, W. Pedrycz, Q. Zhu, W. Zhu, Unsupervised feature selection via maximum projection and minimum redundancy, Knowl.-Based Syst. 75 (2015) 19-29.

[34] 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.

[35] K. Yu, T. Zhang, Y. H. Gong, Nonlinear learning using local coordinate coding, in: Advances in Neural Information Processing Systems, 2009, pp. 2223-2231.

[36] X. He, P. Niyogi, Locality Preserving Projections, in: Advances in Neural Information Processing Systems, 2003, pp. 153-160.

[37] S. Roweis, L. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science. 290 (5500) (2000) 2323-2326.

[38] M. Belkin, P. Niyogi, Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, in: Advances in Neural Information Processing Systems, 2001, pp. 585-591.

[39] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, C. Zhang, Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering, IEEE Trans. Neural Netw. 22 (11) (2011) 1796-1808.

[40] F. Shang, Y. Liu, F. Wang, Learning spectral embedding for semi-supervised clustering, 2011 IEEE 11th International Conference on Data Miming, 2011, pp. 597-606.

[41] 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.

[42] W. Xu, Y. Gong, Document clustering by concept factorization, in: Proceedings of International Conference on Research and Development in Information. Retrieval (SIGIR'04), Sheffield, UK, 2004, pp. 202-209.

[43] D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, IEEE Trans. Knowl. Data Eng. 23 (6) (2011) 902-913.

[44] F. Shang, L.C. Jiao, F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, Pattern Recogn. 45 (6) (2012) 2237-2250.

[45] J. Bu, P. Li, C. Chen, Z. He, D. Cai, Relational multi-manifold co-clustering, IEEE Trans. Cybern. 43 (6) (2013) 1871-1881.

[46] J. Ye, Z. Jin, Dual-graph regularized concept factorization for clustering, Neurocomputing. 138 (2014) 120-130.

[47] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res., 7 (2006) 2399-2434.

[48] L. Chova, G. Valls, J. Mari, J. Calpe, Semi-supervised image classification with Laplacian

support vector machines, IEEE Geosci. Remote Sens. Lett, 5 (2008) 336-340.

[49] Z. Yang and Y. Xu, Laplacian twin parametric-margin support vector machine for semi-supervised classification, Neurocomputing, 171 (2016) 325-334.

[50] Z. Zhao, L. Wang, H. Liu, Efficient spectral feature selection with minimum redundancy, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010, pp. 673-678.

[51] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multicluster data, in: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010, pp. 333-342.

[52] 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.

[53] 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) 2168-2267.

[54] D. D. Lee, H. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556-562.

[55] A. Rakhlin, A. Caponnetto, Stability of K-Means clustering, in: Advances in Neural Information Processing Systems, 2007, pp. 216-222.

[56] H. Lu, Z. Fu, X. Shu, Non-negative and sparse spectral clustering, Pattern Recogn. 47 (1) (2014) 418-426.

[57] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, X. Zhou, l2,1-Norm regularized discriminative feature selection for unsupervised learning, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 1589-1594.

[58] F. Bach, Consistency of the group lasso and multiple kernel learning, J. Mach Learn. Res. 9 (2008) 1179-1225.

[59] M. Wu, B. Schölkopf, A local learning approach for clustering. In: Advances in Neural Information Processing Systems, 2007, pp. 1529-1536.

[60] A. Strehl, J. Ghosh, Cluster ensembles-a knowledge reuse framework for combining multiple partitions. J. Mach Learn. Res. 3 (2002) 583-617.

[61] C. Papadimitriou, K. Steiglitz, Combinatorial optimization: Algorithms and complexity, Dover, New York, NY, USA, 1998.