Neurocomputing 128 (2014) 304–315

Contents lists available at ScienceDirect

Neurocomputing

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

Locality and similarity preserving embedding for feature selection

Xiaozhao Fang [a], Yong Xu [a,b], Xuelong Li [c,*], Zizhu Fan [a], Hong Liu [d], Yan Chen [e]

[a] Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, Guangdong, PR China
[b] Key Laboratory of Network Oriented Intelligent Computation, Shenzhen 518055, Guangdong, PR China
[c] Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, PR China
[d] Engineering Lab on Intelligent Perception for Internet of Things, Shenzhen Graduate School, Peking University, Shenzhen 518055, Guangdong, PR China
[e] Shenzhen Sunwin Intelligent Co., Ltd, Shenzhen 518057, Guangdong, PR China

ARTICLE INFO

Article history:
Received 24 June 2013
Received in revised form
16 August 2013
Accepted 17 August 2013
Communicated by G.-B. Huang
Available online 14 November 2013

Keywords:
Feature selection
Locality and similarity preserving
Sparse reconstruction
Transformation matrix
Discriminating information

ABSTRACT

Feature selection (FS) methods have commonly been used as a main way to select the relevant features. In this paper, we propose a novel unsupervised FS method, *i.e.*, locality and similarity preserving embedding (LSPE) for feature selection. Specifically, the nearest neighbor graph is firstly constructed to preserve the locality structure of data points, and then this locality structure is mapped to the reconstruction coefficients such that the similarity among these data points is preserved. Moreover, the sparsity derived by the locality is also preserved. Finally, the low dimensional embedding of the sparse reconstruction is evaluated to best preserve the locality and similarity. We impose $\ell_{2,1}-$norm on the transformation matrix to achieve row-sparsity, which allows us to select relevant features and learn the embedding simultaneously. The selected features have good stability due to the locality and similarity preserving, and more importantly, they contain natural discriminating information even if no class labels are provided. We present the optimization algorithm and analysis of convergence of the proposed method. The extensive experimental results show the effectiveness of the proposed method.

1. Introduction

In the fields of computer vision, data mining and machine learning, a mass of data is represented by high dimensional feature vectors. The original high dimensional feature vector might contain a large portion of redundant information, even corrupted noises. The direct way to deal with the problem is dimensionality reduction (DR). In the literature, there are two different ways to perform DR: feature selection and feature learning (or 'feature extraction') [1,2]. Feature selection aims to select a few relevant features to represent the original high dimensional feature vector meanwhile removing unfavorable features that seriously affect the performance of the algorithm [3]. Generally speaking, feature selection can produce three benefits: speeding up the learning process, improving the mode generalization capability and alleviating the effect of the curse of dimensionality [4]. Compared with feature learning which may introduce some new features for original data representation, feature selection does not change the original representation of data. Consequently, feature selection is preferred if the original physical meaning of each feature is demanded to retain in a task. For example, in molecular biology research, it is easy to identify a set of genes that are relevant to a key biological process by using feature selection. However, it is hard to interpret the results of feature learning because the features learned from the original data are a combination of all the original features. Thus, results of feature selection can well interpret which features are important to a given task.

In the past two decades, many effective feature selection algorithms have been proposed [5,6], which can be classified into three different categories: filter, wrapper, and embedded methods [7]. The filter methods commonly filter out some features that possess poor information by using statistical properties [8–12]. The filter methods do not directly optimize the performance of any specific learning algorithm. Thus they usually do not perform as well as some state-of-the-art methods. In wrapper methods, feature selection is performed, and simultaneously, the performance of algorithms is optimized [13,14]. Wrapper methods usually outperform filter methods in the performance. However, wrapper methods have high computational complexity because they need to train a large number of classifiers [15]. Many heuristic algorithms and hybrid methods have been proposed to alleviate this issue [16]. Nevertheless, these heuristic algorithms also have to take a large amount of time to perform the search [15]. To reduce the complexity, in practice, a simple classifier is used to evaluate the goodness of feature subsets and then the selected

* Corresponding author.
E-mail addresses: xzhfang168@gmail.com (X. Fang), laterfall2@yahoo.com.cn (Y. Xu), xuelong_li@opt.ac.cn (X. Li), zzfan3@yahoo.com.cn (Z. Fan), hongliu@pku.edu.cn (H. Liu), jadechenyan@gmail.com (Y. Chen).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.08.040

features are sent into a complicated classifier for ultimate data analysis. Another disadvantage of wrapper methods is that they are required to manually specify the parameters of the trained classifiers. This is probably one of the main reasons why filter methods are more popular in practical applications than wrapper methods [17,18]. Embedded methods usually incorporate feature selection into the learning process of the designed classifier [15,19–21] and show good performance.

Recently, several manifold learning-based algorithms were developed to perform DR, such as locally liner embedding (LLE) [22], isometric feature mapping (ISOMAP) [23] and Laplacian eigenmaps (LE) [24]. These methods are based on the idea that data points are actually sampled from a low-dimensional manifold that is embedded in a high-dimensional space. However, as pointed out in [25], all these manifold learning methods suffer from the problem that a new data point cannot easily find its low-dimensional embedding by utilizing the low-dimensional embedding results of the training data points (out of sample) because of the implicitness of the nonlinear mapping. Locality preserving projections (LPP) [26], locality preserving discriminant projections (LPDP) [27] and neighborhood preserving embedding (NPE) [28] were proposed to address this problem. Some novel methods, which integrates the theory of sparse representation and subspace learning, have also been proposed and successfully applied in many real-world applications [29,30]. The representative methods include sparse neighborhood preserving embedding (SNPE) [31], sparsity preserving projections (SPP) [32] and local coordinate coding (LCC) [33]. It should be noted that, in [30,31,33], the locality constraint is imposed on sparse coding (SC). Moreover, in [33], the theoretical analysis pointed out that under certain assumptions locality is more essential than sparsity and helpful for successful nonlinear function learning. To achieve good classification performance, the coding scheme should generate similar codes for similar descriptors [34]. Such locality and similarity is useful for producing good discriminative ability of the designed algorithm [30,34]. For example, if two data points x_i and x_j are close in the intrinsic geometry of the data distribution, then the optimal reconstruction coefficients of these two data points are also close to each other. The above-mentioned methods ignore the problem that there are many unfavorable features in the original high dimensional feature representation. Most previousalgorithms perform the sparse reconstruction task in the original high dimensional feature space, *e.g.*, SNPE, Laplacian sparse coding (LSc) [30] and LCC. However, it is difficult to perform the sparse reconstruction in a high dimensional feature space due to the fact that the high dimensional feature representation is not always reliable and even corrupted by noises. Intuitively, the sparse reconstruction task may benefit from the feature extraction process because it may remove the unfavorable features and noises. Therefore, a scheme which simultaneously integrates both the sparse reconstruction and optimal feature representation is demanded.

The above observations motivate us to consider how to devise an elegant method which can achieve the above purposes. In this paper, we propose a novel unsupervised feature selection method, *i.e.*, locality and similarity preserving embedding (LSPE) for feature selections. Specifically, in the proposed method, the nearest neighbor graph G is firstly constructed to preserve the locality and similarity among data points to be reconstructed, and then the low dimensional embedding of the reconstruction is generated with the goal to best preserve such locality and similarity. As suggested by LCC [33], locality is more essential than sparsity, as locality must lead to sparsity but not necessary vice versa. Therefore, the reconstruction coefficients of our method are sparse in the case where similar data points have nearly same reconstruction coefficients. Generally speaking, LSPE seeks the projections which cannot only preserve the locality and similarity but also the sparse reconstruction relationship. We impose $\ell_{2,1}-$norm minimization on the transformation matrix to simultaneously select relevant features and learn the embedding. By preserving the locality and similarity, LSPE can alleviate the instability of selected features. This will be confirmed by the subsequent experimental results. Although no class labels are provided, LSPE tends to select the discriminative features due to the sparsity [32]. We can learn a sparse transformation matrix from the $\ell_{2,1}-$norm minimization for feature ranking. We provide an effective algorithm to solve this $\ell_{2,1}-$norm minimization problem. And the analysis of convergence of the proposed method is presented.

The most important contributions of our proposed method are as follows.

(1) The sparse reconstruction is finally performed on the derived optimal low dimensional space, which can effectively eliminate the influence of the unfavorable features.
(2) Unlike most previous feature selection algorithms which separately treat the embedding learning and the feature selection, LSPE unifies these two objectives.
(3) Unlike SPP [32], which uses a two-stage strategy to learning the sparse reconstruction coefficient matrix and the transformation matrix, our method optimizes them simultaneously.
(4) Although supervised information is not needed, LSPE can select discriminative features in comparison with some similar unsupervised feature selection algorithms.
(5) Compared with other unsupervised feature selection algorithms, the features selected by LSPE have good stability.

The remaining of this paper is organized as follows: Section 2 briefly reviews some methods that are closely related to our method. Section 3 introduces the basis idea of locality and similarity preserving embedding (LSPE) for feature selection; Section 4 provides some discussion of the proposed method including the analysis of convergence of the proposed method. Extensive experiments are conducted in Section 5. Finally, we conclude the paper in Section 6.

2. Related methods

In this section, we will introduce some notations. The $\ell_{2,1}-$norm of a matrix is first introduced in [35] as a rotational invariant ℓ_1-norm and has attracted increasing attention [36,37]. For the matrix $A \in \Re^{m\times d}$, let $A_{i\cdot}$ the ith row of A. The $\ell_{2,1}-$norm of A is defined as

$$\|A\|_{2,1} = \sum_{i=1}^{m} \|A_{i\cdot}\|_2 \tag{1}$$

We consider an original set of n data points $X=[x_1, x_2, \ldots, x_n] \in \Re^{m\times n}$. The task of dimension reduction is to find a linear transformation matrix $A \in \Re^{m\times d}$ to transform the original high dimensional data point $x_i \in \Re^m$ into a low dimensional form $y_i \in \Re^d$ $(d<m)$ by using $y_i = A^T x_i$.

Our method is fundamentally based on two of the most popular manifold learning methods, NPE and SPP. We will review these two methods briefly in next subsections. It should be noted that there is a distinct difference between the sparse matrix learned by $\ell_{2,1}-$norm and ℓ_1-norm. Using the unified sparse subspace learning framework (SSL) [38] as an example, we respectively impose $\ell_{2,1}-$norm and ℓ_1-norm on the transformation matrix. Fig. 1(a) gives a toy example of the transformation matrix learned by $\ell_{2,1}-$norm. Each row of this transformation matrix corresponds to a feature, while each column corresponds to a dimension of the embedding. We can see that the 3rd and 5th rows are all zeros, which indicate that the 3rd and 5th rows correspond to the irrelevant features and they should be

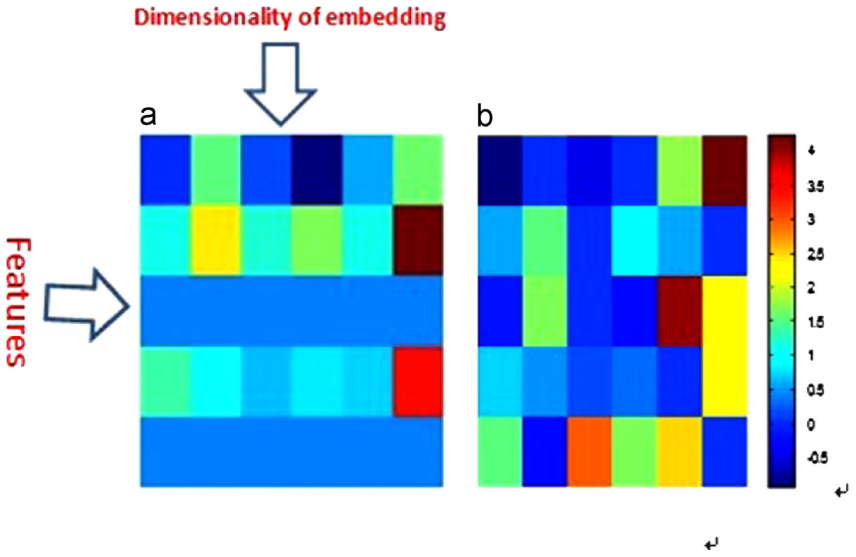

Fig. 1. A toy example for the transformation matrix learned by (a) $\ell_{2,1}-$norm and (b) ℓ_1-norm.

discarded. Hence it is very clear which features are really useful to the task. Fig. 1(b) is the toy example of the transformation matrix learned by ℓ_1-norm. We can see that for the first dimension of the embedding, the 2nd and 4th features are not selected. However, for the second dimension of the embedding, all the features are selected expect for the 1st and 4th ones. Therefore, it is still unclear which features are really useful as a whole. In this paper, we aim to use $\ell_{2,1}-$norm to learn a transformation matrix with the similar row-sparsity property as the one shown in Fig. 1(a).

2.1. Neighborhood preserving embedding (NPE)

Different from Principal Component Analysis (PCA) [39,40], NPE [28] aims at preserving the local neighborhood structure of the data points. NPE evaluates the affinity weight matrix using local least squares approximation [41]. The first step of NPE constructs an adjacency graph by using k-nearest neighbors (KNN) algorithm [42]. Then, it uses the local approximation error to compute the weights on these edges

$$\min_P \quad \sum_j \|x_i - \sum_j P_{i,j}x_j\|^2$$
$$\text{s.t.} \quad \sum_j P_{i,j} = 1, \quad j = 1,2,\ldots,n \tag{2}$$

where P is the reconstruction coefficients matrix (the affinity weight matrix). The second step of NPE is to utilize a reasonable criterion for determining a linear projection. This can be converted into the following generalized eigenvector problem [28]:

$$XMX^T z_i = \lambda XX^T z_i \tag{3}$$

where

$$M = (1-P)^T(1-P)$$

$$I = diag(1,\ldots,1)$$

Let z_i $(i = 1,2,\ldots,d)$ be the eigenvectors respectively corresponding to the first d smallest eigenvalues of the above eigenvector problem. The desirable optimal low-dimensional representation of the original data is as follows:

$$x_i \to y_i = Z^T x_i \tag{4}$$

where y_i is the desirable representation. From the description of NPE, we can see that NPE is indeed a linear version of LLE [22].

2.2. Sparsity preserving projections (SPP) [32]

SPP constructs the affinity weight matrix in a completely different way from LLE. SPP first uses as few as possible data points from X to reconstruct each data point $x_i \in X$. Hence a sparse reconstruction vector s_i for x_i is sought to perform the following reconstruction task:

$$\min_{s_i} \quad \|s_i\|_1$$
$$\text{s.t.} \quad x_i = Xx_i, \quad 1 = 1^T s_i \tag{5}$$

where $\|\cdot\|_1$ is the ℓ_1-norm [43]. $1 \in \Re^n$ is a vector of all ones. After computing the sparse reconstruction vector s_i for each x_i $(i = 1,2,\ldots n)$, SPP obtains the sparse reconstruction matrix $S = [s_1,\ldots,s_n]$. The element $s_{i,j}$ in S essentially reflects a close relation between x_i and x_j and it is reasonable to use S as the affinity weight matrix. Similar to LLE and NPE, SPP seeks the projections which best preserve the sparse reconstruction relationship. SPP has the following objective function [32]:

$$\min_Q \sum_{i=1}^n \|Q^T x_i - Q^T X s_i\|^2 \tag{6}$$

where Q is the projection matrix. The problem defined by (6) can be converted into the problem to minimize the following formulation:

$$\sum_{i=1}^n \|Q^T x_i - Q^T X s_i\|^2 = Q^T \left(\sum_{i=1}^n (x_i - Xs_i)(x_i - Xs_i)^T \right) Q \tag{7}$$

The optimal projection vectors Q can be obtained by solving the following generalized eigenvalue problem:

$$X(I - S - S^T + S^T S)X^T q_i = \lambda XX^T q_i \tag{8}$$

Specifically, let $q_1,\ldots,q_d$ be the eigenvectors of (8) corresponding to the first d smallest eigenvalues, $\lambda_1 \le,\ldots,\le \lambda_d$. Then, the transformation matrix of SPP is $Q = [q_1,\ldots,q_d]$.

3. Locality and similarity preserving embedding for feature selection

In this section, we will present the basic idea of our method. To achieve good classification performance, the reconstruction scheme should follow the rule that similar data points should

have similar reconstruction coefficients [33,44]. To obtain this purpose, we reformulate the problem as follows. For the set of m-dimensional data points $X=[x_1,\ldots,x_n]\in\Re^{m\times n}$, we can construct a nearest neighbor graph G with n vertices each of which denotes a data point [45]. Let W be the weight matrix of G. The weight setting is subject to the following criterion: if x_i is among the k-nearest neighbors of x_j or x_i is among the k-nearest neighbors of x_i, $W_{i,j}=\exp(-\|x_i-x_j\|^2/\sigma)$ (σ is the heat kernel parameter), otherwise $W_{i,j}=0$. To map the weight matrix to the sparse reconstruction coefficients, an ideal mapping is to minimize the following objective function:

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|s_i-s_j\|^2W_{i,j}=\mathrm{Tr}(SLS^T) \tag{9}$$

where S is the reconstruction coefficient matrix. Let D be a diagonal matrix whose entries are column or row sums of W, $D_{i,i}=\sum_j W_{j,i}$. $L=D-W$ is the graph Laplacian. We expect that the desirable characteristics (the locality and similarity) in the original high dimensional feature space can be preserved in the low dimensional embedding space. In other words, the low dimension embedding of the sparse reconstruction can best preserve the locality and similarity. Unlike SPP [32], where the sparse reconstruction coefficient matrix S is firstly learned in the original high dimensional feature space, and then the projection is sought to best preserve this optimal S, we optimize S and the transformation matrix simultaneously. Therefore, we define the following objective function:

$$\min_{A,S}\sum_{i=1}^{n}\|A^T(x_i-Xs_i)\|^2+\frac{1}{2}\beta\sum_{i=1}^{n}\sum_{j=1}^{n}\|s_i-s_j\|^2W_{i,j} \tag{10}$$

where $A\in\Re^{m\times d}$ is the transformation matrix and d is the dimensionality of embedding. We utilize $\ell_{2,1}-$norm minimization constraint to select the relevant features which can best preserve the locality and similarity among data points to be reconstructed. Denote $A_{i\cdot}$ ($i=1,\ldots,m$) as the ith row vector of A which is used to measure the importance of the ith feature. We expect that the transformation matrix holds the sparsity property for feature ranking. In other words, we expect that only a few numbers of $A_{i\cdot}$ are non-zeros. To this end, we impose $\ell_{2,1}-$norm on A and try to minimize $\|A\|_{2,1}$. Therefore, our objective function can be formulated as follows:

$$\min_{A,S}\sum_{i=1}^{n}\|A^T(x_i-Xs_i)\|^2+\frac{1}{2}\beta\sum_{i=1}^{n}\sum_{j=1}^{n}\|s_i-s_j\|^2W_{i,j}+\alpha\|A\|_{2,1} \tag{11}$$

where β and α are two balance parameters.

3.1. Solution

It seems that solving $\ell_{2,1}-$norm problem defined in (1) is difficult since it is hard to derive its closed solution directly. Inspired by [4], we divide the problem in (11) into two steps: learning the reconstruction coefficient matrix S while fixing the transformation matrix A, and learning A while fixing S. For convenience, the problem in (11) can be rewritten as follows:

$$\begin{aligned}
&\sum_{i=1}^{n}\|A^T(x_i-Xs_i)\|^2+\frac{1}{2}\beta\sum_{i=1}^{n}\sum_{j=1}^{n}\|s_i-s_j\|^2W_{i,j}+\alpha\|A\|_{2,1}\\
&=\mathrm{Tr}(\sum_{i=1}^{n}A^T(x_i-Xs_i)(x_i-Xs_i)^TA)+\beta\,\mathrm{Tr}(SLS^T)+\alpha\|A\|_{2,1}\\
&=\mathrm{Tr}(A^T(\sum_{i=1}^{n}(x_i-Xs_i)(x_i-Xs_i)^T)A)+\beta\,\mathrm{Tr}(SLS^T)+\alpha\|A\|_{2,1}\\
&=\mathrm{Tr}(A^T(XX^T-XSX^T-XS^TX^T+XS^TSX^T)A)+\beta\,\mathrm{Tr}(SLS^T)+\alpha\|A\|_{2,1}\\
&=\mathrm{Tr}(A^TX(I-S-S^T+S^TS)X^TA)+\beta\,\mathrm{Tr}(SLS^T)+\alpha\|A\|_{2,1}
\end{aligned} \tag{12}$$

If S is fixed, we denote $L(A)=\mathrm{Tr}(A^TXKX^TA)+\alpha\|A\|_{2,1}$, where $K=(I-S-S^T+S^TS)$. By constructing an auxiliary function, $L(A)$ can be rewritten as $L(A)=\mathrm{Tr}(A^TXKX^TA)+\alpha\,\mathrm{Tr}(A^TUA)$, where $U\in\Re^{m\times m}$ is a diagonal matrix whose ith diagonal element is

$$U_{i,i}=\frac{1}{2\|A_{i\cdot}\|_2} \tag{13}$$

To avoid degenerated solution, the orthogonal constraint $A^TA=I$ is imposed. Thus, the objective function becomes

$$\begin{aligned}
&\arg\min_{A}\quad \mathrm{Tr}(A^T(XKX^T+\alpha U)A)\\
&\text{s.t.}\quad A^TA=I
\end{aligned} \tag{14}$$

The solution of (14) can be obtained by solving the following eigenvalue problem:

$$(XKX^T+\alpha U)a_i=\lambda a_i \tag{15}$$

Let $A=[a_1,\ldots,a_d]$ be the solution of (15). These column vectors a_i ($i=1,2,\ldots,d$) correspond to the eigenvectors associated with the first d smallest eigenvalues.

Recalling the definition of U_{ii} in (13), we know that $\mathrm{Tr}(A^TUA)=\|A\|_{2,1}/2$ if $A_{i\cdot}\neq 0$. Thus we can say that $\min_A \mathrm{Tr}(A^TUA)$ is a sparse constraint on A. If $\|A_{i\cdot}\|_2$ is small, then U_{ii} is large and thus the minimization of $L(A)$ trends to force $\|A_{i\cdot}\|_2$ to be a very small value. After several times of iteration, some $\|A_{i\cdot}\|_2$'s may be close to zero and thus we obtain a sparse A. Since problem (14) is solved in an iteration way, we can initialize U by an identity matrix. In practice, the traditional regularization way can be used to redefine $U_{ii}=1/(2\|A_{i\cdot}\|_2+\zeta)$ (ζ is a very small constant) because $\|A_{i\cdot}\|_2$ could be zero theoretically. In summary, we present Algorithm 1 for optimizing (14) as follows.

Algorithm 1. Optimizing (14).

Initialize: $S=1_{n\times n}$, where $1_{n\times n}$ is a matrix of ones;
Compute $K=(I-S-S^T+S^TS)$;
Set $t=0$ and initialize $U_0\in\Re^{m\times m}$ as an identity matrix;
repeat
  Compute $P_t=(XKX^T+\alpha U)$;
  Compute $A_t=[p_1,\ldots,p_d]$, where $p_1,\ldots,p_d$ are the eigenvectors of P_t corresponding to the first d smallest eigenvalues;
  Update the diagonal matrix U_{t+1} as

$$U_{t+1}=\begin{bmatrix}\frac{1}{2\|A^t_{1\cdot}\|_2} & & \\ & \cdots & \\ & & \frac{1}{2\|A^t_{m\cdot}\|_2}\end{bmatrix};$$

  $t=t+1$;
until *Convergence*

When A is fixed, we would like to take the derivative of $C(S)=\min_S(\mathrm{Tr}(D(I-S-S^T+S^TS)D^T)+\beta\,\mathrm{Tr}(SLS^T))(A^TX=D)$ with respect to S and set it to zeros, namely

$$\frac{\partial C(S)}{\partial S}=-2D^TD+2SD^TD+2\beta SL=0 \tag{16}$$

or equivalently,

$$S=D^TD(D^TD+\beta L)^{-1} \tag{17}$$

After deriving A and S, we use ℓ_2-norm of $A_{i\cdot}$, *i.e.*, $\|A_{i\cdot}\|_2$, to rank the features. The larger $\|A_{i\cdot}\|_2$ is, the more important this feature is. We can select a number of features whose $\|A_{i\cdot}\|_2$ are larger than a threshold which is set in advance.

In summary, we describe the detailed procedure of LSPE in Algorithm 2 as follows.

Algorithm 2. The detailed procedure of LSPE.

set $t=0$;
repeat
Compute A^t based on ***Algorithm 1***;
Compute $S^t=(D^t)^TD^t((D^t)^TD^t+\beta L)^{-1}$,
where $D^t=(A^t)^TX$;
$t=t+1$;
until *Convergence*
Sort each feature $f_i|_{i=1}^m$ according to $\|A_{i.}\|_2$ in descending order and select the top ranked ones

4. Discussions

In this section, we will analyze the convergence behavior of LSPE and then give comparisons between LSPE and some related works.

4.1. Convergence analysis

Before starting our analysis, we give a lemma [4].

Lemma 1. *For any non-zero vectors q, $p\in\Re^m$, the following result holds:*

$$\|q\|_2-\frac{\|q\|_2^2}{2\|p\|_2}\le\|p\|_2-\frac{\|p\|_2^2}{2\|p\|_2} \tag{18}$$

Proof. The detailed proof is similar as that in [4]. □

In our method, solving A usually requires computationally demanding optimization procedures whereas the solution of S can be derived analytically by the analytical solution: $S=D^TD(D^TD+\beta L)^{-1}$. So the solution of S can be performed fast. In practice, we only need to prove that the solution A in Algorithm 1 can monotonically decrease the objective function value in (11) in each iteration.

Theorem 1. *The optimization procedure in solving (14) will monotonically decrease the objective function value in (11) in each iteration.*

Proof. When we fix U as U^t in the ith iteration and compute A^{t+1} and S^{t+1}, the following inequality holds:

$$\begin{aligned}&\mathrm{Tr}((A^{t+1})^TXK^{t+1}X^TA^{t+1})+\beta\,\mathrm{Tr}(S^{t+1})L(S^{t+1})^T)\\&\quad+\alpha\,\mathrm{Tr}((A^{t+1})^TU^tA^{t+1})\\&\quad\le\mathrm{Tr}((A^t)^TXK^tX^TA^t)+\beta\,\mathrm{Tr}(S^tL(S^t)^T)\\&\quad+\alpha\,\mathrm{Tr}((A^t)^TU^tA^t)\end{aligned} \tag{19}$$

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

$$\begin{aligned}&\mathrm{Tr}((A^{t+1})^TXK^{t+1}X^TA^{t+1})+\beta\,\mathrm{Tr}(S^{t+1}L(S^{t+1})^T)\\&\quad+\alpha\|A^{t+1}\|_{2,1}+\alpha\sum_{i=1}^m\left(\frac{\|A_{i.}^{t+1}\|_2^2}{2\|A_{i.}^t\|_2}-\|A_{i.}^{t+1}\|_2\right)\\&\quad\le\mathrm{Tr}((A^t)^TXK^tX^TA^t)+\beta\,\mathrm{Tr}(S^tL(S^t)^T)\\&\quad+\alpha\|A^t\|_{2,1}+\alpha\sum_{i=1}^m\left(\frac{\|A_{i.}^t\|_2^2}{2\|A_{i.}^t\|_2}-\|A_{i.}^t\|_2\right)\end{aligned} \tag{20}$$

According to Lemma 1, we have

$$\frac{\|A_{i.}^{t+1}\|_2^2}{2\|A_{i.}^t\|_2}-\|A_{i.}^{t+1}\|_2\ge\frac{\|A_{i.}^t\|_2^2}{2\|A_{i.}^t\|_2}-\|A_{i.}^t\|_2 \tag{21}$$

Combining (20) with (21), we have the following inequality:

$$\begin{aligned}&\mathrm{Tr}((A^{t+1})^TXK^{t+1}X^TA^{t+1})+\beta\,\mathrm{Tr}(S^{t+1}L(S^{t+1})^T)\\&\quad+\alpha\|A^{t+1}\|_{2,1}\\&\quad\le\mathrm{Tr}((A^t)^TXK^tX^TA^t)+\beta\,\mathrm{Tr}(S^tL(S^t)^T)\\&\quad+\alpha\|A^t\|_{2,1}\end{aligned} \tag{22}$$

which indicates that the objective function value in (11) will monotonically decrease using the updating rule in Algorithm 1. Besides, since the two items in (14) are convex function and thus (14) has a lower bound. Thus, the above iteration will converge to the global solution. □

4.2. Comparison to other methods

Undoubtedly, LSPE is closely related to SPP. In other words, LSPE is an improved version of SPP. Both LSPE and SPP seek the projections that best preserve the sparse reconstruction relationship. However, SPP uses a two-stage strategy to construct the sparse reconstruction coefficient matrix and the transformation matrix, our method optimize them simultaneously. In this way, LSPE can learn them optimally. Moreover, LSPE maps the locality among data points to the sparse reconstruction coefficients such that these reconstruction coefficients vary smoothly along the geodesics of the data manifold. Moreover, the features selected by LSPE have good stability because they consistently guarantee that the similar data points always have nearly the same reconstruction coefficients. LSPE can select the relevant features by imposing $\ell_{2,1}$−norm on the transformation matrix. However, SPP does not lead to feature selection.

Considering the deduction of LSPE, we know that LSPE is also related to Laplacian score for feature selection (LapScore) [8] and spectral feature selection (SPEC) [46]. LapScore and LSPE construct the graph to characterize the data manifold. LapScore selects features which can best preserve the locality relationship revealed by weight matrix W. However, LSPE select features which can best preserve both the locality and the similarity among data points to be reconstructed. SPEC can be regarded as an extension of LapScore. LSPE focuses on the unsupervised feature selection. SPEC, however, mainly emphasizes the supervised case. Although the locality pays an important role in developing various kinds of algorithms, *e.g.*, DR, semi-supervised learning algorithm, the features selected by the locality preserving-based feature selection algorithms may not contain discriminant information due to the lack of label information. The reconstruction coefficient of LSPE is sparse because the locality restraint is imposed on the reconstruction coefficient [34]. This entitles the features selected by LSPE to more discriminant ability than those by using LapScore and SPEC, which is proved by the subsequent experimental results.

Feature selection via joint embedding learning and sparse regression (JELSR) [47] also has somewhat relationship with LSPE. JELSR unifies the procedures of the embedding learning and the sparse regression into a framework. More precisely, JELSR can be regarded as solving the following problem:

$$\begin{aligned}\min_{W,Y}\quad&\mathrm{Tr}(YLY^T)+\beta(\|W^TX-Y\|_2^2+\alpha\|W\|_{2,1})\\\text{s.t.}\quad&YY^T=I\end{aligned} \tag{23}$$

where Y is the low dimension representation of the original data X and W is the projection matrix. JELSR mainly focus on the issue that nearby points, in the desired low dimensional space, should have similar properties. Similarly, LSPE also seeks to this purpose. We set $y_i=A^Tx_i$. Our objective function (11) can be formulated as follows:

$$\begin{aligned}\min_{A,S}\quad&\mathrm{Tr}(Y(I-S-S^T+S^TS)Y^T)+\beta\,\mathrm{Tr}(SLS^T)+\alpha\|A\|_{2,1}\\\text{s.t.}\quad&A^TA=I\end{aligned} \tag{24}$$

From (24), we know that LSPE imposes locality and similarity preserving on the reconstruction coefficients S and simultaneously delivers such preserving to the low dimensional representation Y by virtual of S. Thus, we can say the first terms in (23) and (24) share the similar purpose. Comparing the formulations in (23) and (24), it is easy to find out that JELSR selects the features which can best preserve the locality. However, LSPE selects features which simultaneously best preserve the locality and the similarity. This somewhat consistent with the purpose of Laplacian sparse coding (LSc) [44]. Thus, it outperforms JELSR in many cases. Note that LSc performs the sparse reconstruction in the original high dimensional feature space while LSPE does in the desirable low dimensional embedding space.

5. Experimental results

In this section, we evaluate the performance of LSPE on several real data sets. We perform three groups' experiments. The first group evaluates LSPE using K-means clustering [47] and clustering using local discriminant models and global integration (LDMGI) [48] as the metrics. The second group evaluates LSPE using Nearest Neighbor (NN) classifier [42] and Multiple Kernel Learning (MKL) method proposed in [49,50] for classification. We discuss the influence of the parameters used in LSPE in the last group. We compare LSPE with the following unsupervised feature selection algorithms, LapScore [8], SPEC [46], Unsupervised feature selection for multi-cluster data (MCFS) [51], JELSR [47] and Efficient spectral feature selection with minimum redundancy (MRSF) [52]. We use all features as the baseline in our experiments. The code of the proposed method is available at http://www.yongxu.org/lunwen.html. For some graph-based algorithms, such as LapScore, MCFS, SPEC and LSPE, we tune k which specifies the size of neighborhood, by selecting the most suitable value from $\{3,5,7,10,15\}$ for all the data sets. Similarly, we tune the heat kernel parameter σ from $\{10^0, 10^3, 10^5\}$. For LSPE, we tune parameters α from $\{300,500,800,1000,2000,3000,4000,5000,6000,7000,8000\}$ and β from $\{0.01,0.1,0.5,1.0,3.0,5.0,7.0,9.0,11.0,13.00,15.00,17.00\}$. We begin with a description of these data sets.

Table 1
A summary of characteristics of these data sets.

Data set	Dimensionality	Size	Class
Umist	644	575	20
Isolet	617	1560	26
ORL	1024	400	40
LFW	127	1251	86
Sonar	60	208	2
BC	30	569	2
Ionosphere	34	351	2
Vehicle	18	846	4
Yale	105	165	15

5.1. Data sets descriptions

Nine different data sets, including Umist [47], Isolet [47], Sonar [53], Breast Cancer (BC) [54], Ionosphere [55], ORL [28], Vehicle [56] , Yale [26] and LFW [57], are used in our experiments. Some data sets in Matlab format after being preprocessed are available at: http://www.cad.zju.edu.cn/home/dengcai/Data/data.html A summary of characteristics of these data sets is presented in Table 1.

5.2. Clustering results with K-means clustering and LDMGI

In the first group experiment, K-means clustering and LDMGI are employed on the several data sets to evaluate the performance of LSPE with fixed selected features. Two metrics, the accuracy (AC) and the normalized mutual information metric (MI), are used to measure the clustering performance. Given a data point x_i, let r_i and l_i be the obtained cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(l_i, map(r_i))}{n} \tag{25}$$

where n is the total number of data points and $\delta(x,y)$ is the delta function that equals one if $x=y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus [51]. The best mapping can be determined by using the Kuhn–Munkres algorithm [58]. Let C denote the set of clusters obtained from the ground truth and C' obtained from the algorithms used in this section. Their mutual information metric $MI(C,C')$ is defined as follows [49]:

$$MI(C,C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)} \tag{26}$$

where $p(c_i)$ and $p(c'_j)$ are respectively the probabilities that a sample arbitrarily selected from the data set belongs to the clusters c_i and c'_j and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected sample belongs to the clusters c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information NMI as follows:

$$NMI(C,C') = \frac{MI(C,C')}{\max(H(C), H(C'))} \tag{27}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $NMI(C,C')$ ranges from 0 to 1. $NMI=1$ if the two sets of clusters are identical, and $NMI=0$ if the two sets are independent.

For the other feature selection algorithms, we select their best results as the final results. We set different numbers of selected features for different data sets. In our experiment, each feature selection algorithm is first performed to select features. Then K-means clustering algorithm is performed based on the selected features. Since the results of K-means clustering depend on initializations, we repeated 100 times experiments with random

Table 2
Clustering results of different algorithms on seven data sets (MEAN ± STD%).

Data set	All features	LapScore	SPEC	MCFS	JELSR	MRSF	LSPE
Umist	44.23 ± 1.02	37.30 ± 0.93	42.56 ± 1.20	46.55 ± 1.00	48.90 ± 1.03	48.38 ± 1.05	49.26 ± 1.12
Isolet	50.58 ± 0.85	48.79 ± 0.56	49.50 ± 0.63	54.48 ± 0.84	55.08 ± 0.45	50.80 ± 0.69	56.11 ± 0.63
ORL	50.00 ± 0.43	44.50 ± 0.73	49.88 ± 0.23	49.40 ± 0.93	50.02 ± 0.56	49.78 ± 0.69	50.25 ± 0.80
LFW	18.78 ± 0.33	19.50 ± 2.00	16.60 ± 1.76	19.66 ± 1.50	20.90 ± 2.03	20.40 ± 2.05	22.14 ± 2.50
Ionosphere	63.81 ± 0.50	66.94 ± 2.20	67.70 ± 2.33	57.26 ± 3.00	67.90 ± 2.81	63.00 ± 2.30	70.00 ± 2.66
Sonar	54.32 ± 1.20	58.80 ± 1.14	61.00 ± 1.26	54.20 ± 0.84	64.20 ± 0.94	60.33 ± 1.40	66.25 ± 1.67
BC	72.27 ± 0.20	70.17 ± 0.36	74.00 ± 0.23	71.00 ± 0.58	74.20 ± 0.30	72.79 ± 0.22	75.86 ± 0.24

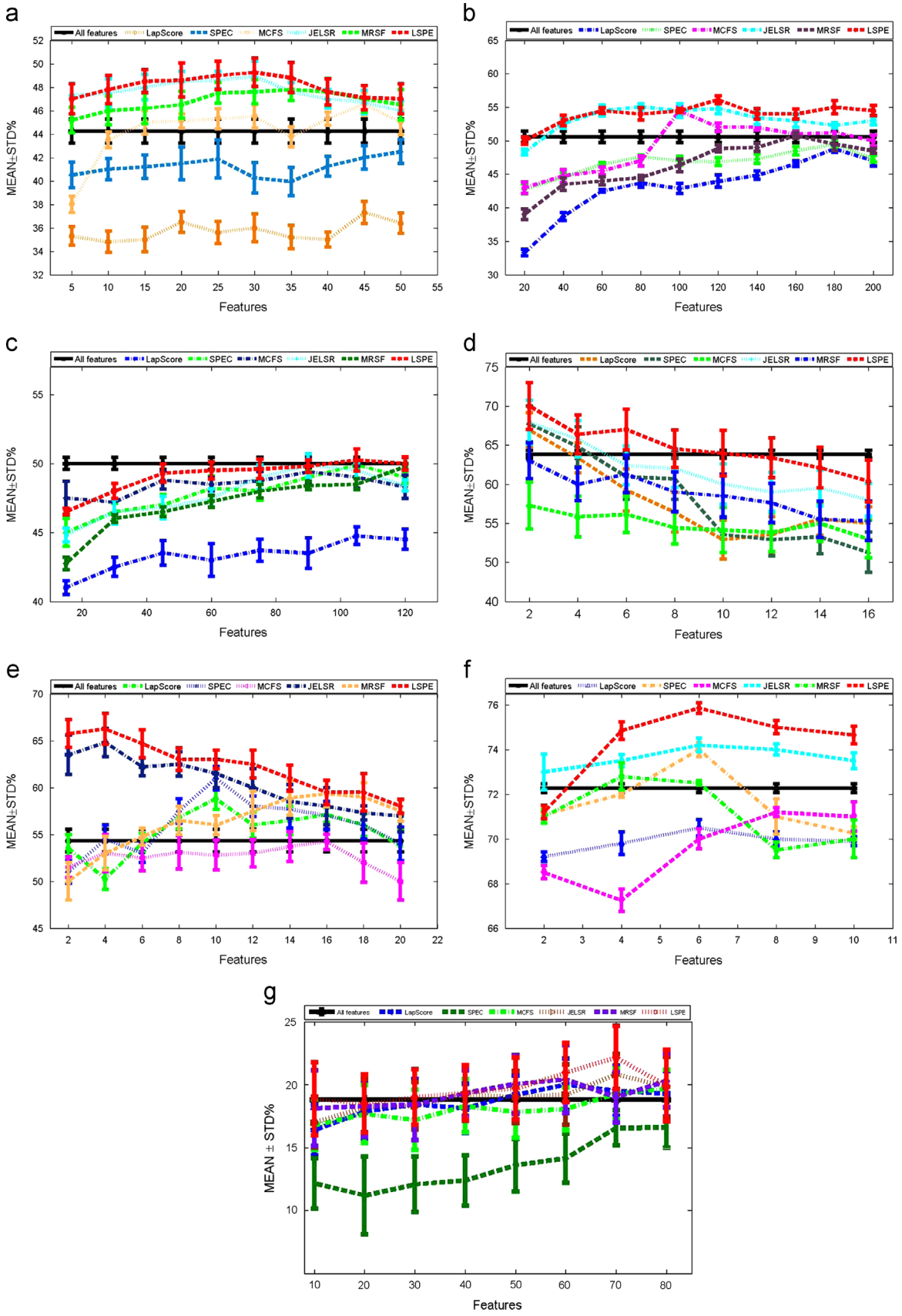

Fig. 2. The detained clustering results of *K*-means clustering on seven different data sets: (a) Umist, (b) Isolet, (c) ORL, (d) Ionosphere, (e) Sonar, (f) BC, (g) LFW.

Table 3
Mean *NMI* with standard deviation of different algorithms on three data sets (MEAN ± STD).

Data set	All features	LapScore	SPEC	MCFS	JELSR	MRSF	LSPE
Umist	0.6030 ± 0.0145	0.5632 ± 0.0152	0.5704 ± 0.0124	0.6920 ± 0.0131	0.7018 ± 0.0164	0.6667 ± 0.0143	0.7091 ± 0.0155
Isolet	0.7302 ± 0.0092	0.6680 ± 0.0120	0.6690 ± 0.0149	0.7043 ± 0.0193	0.7050 ± 0.0134	0.6835 ± 0.0167	0.7101 ± 0.0185
ORL	0.7036 ± 0.0117	0.6780 ± 0.0176	0.7026 ± 0.0165	0.7098 ± 0.0178	0.7020 ± 0.0172	0.7050 ± 0.0181	0.7104 ± 0.0111

Table 4
Clustering results (LDMGI) of different algorithms on three data sets (MEAN ± STD%).

Data set	All features	LapScore	SPEC	MCFS	JELSR	MRSF	LSPE
ORL(189)	68.00 ± 2.03	71.50 ± 2.77	71.50 ± 2.10	70.50 ± 2.52	71.40 ± 2.00	70.30 ± 2.86	72.25 ± 2.18
Isolet(266)	47.92 ± 2.05	49.71 ± 2,02	49.58 ± 2.17	50.74 ± 2.25	51.30 ± 2.22	50.55 ± 2.12	52.28 ± 2.45
Yale(105)	47.88 ± 2.33	50.30 ± 2.11	49.70 ± 2.53	51.52 ± 2.78	51.00 ± 2.06	50.69 ± 2.16	53.33 ± 2.58

Table 5
Clustering results of two algorithms on three sets (MEAN ± STD%).

Algorithms	ORL(189)	Isolet(266)	Yale(105)
All features	52.25 ± 2.10	52.33 ± 2.44	40.70 ± 2.83
SSPE	58.25 ± 1.81	55.16 ± 2.20	44.97 ± 1.50
LSPE	59.00 ± 2.03	54.65 ± 2.19	45.66 ± 1.30

Table 6
Classification error on Sonar data set (MEAN ± STD%).

Method	One third train	One second train	Two third train
All features	22.77 ± 2.09	18.20 ± 2.90	16.47 ± 2.63
LapScore	21.61 ± 2.72	17.67 ± 2.56	14.74 ± 2.50
SPEC	22.68 ± 2.42	18.50 ± 2.13	16.22 ± 2.18
LSPE	21.10 ± 2.58	17.30 ± 2.10	14.20 ± 2.00

initialization and report the mean performance with standard deviation (MEAN ± STD%). Table 2 gives the best clustering results of different feature selection algorithms using different parameters. As seen from Table 2, LSPE outperforms other algorithms. We can also see from this table that JELSR is the second best algorithm. From the analysis in [47], we know that SPEC, MCFS and MRSF adopt a two-step strategy for feature selection. For example, SPEC analyzes features separately and selects features one after another whereas MCFS selects features in batch-mode. For MRSF, it separates embedding learning and sparse regression. However, JELSR integrates the two objectives into one step, which can lead to a good performance. Similarly, LSPE unifies the two objectives of embedding learning and feature selection. Moreover, LSPE imposes the similarity preserving on the reconstruction coefficients. Thus, LSPE perform better than JELSR in our experiments. This observation validates that it is a better way to implement embedding learning and feature selection jointly for feature selection. Fig. 2 gives the detailed clustering results of different number of the selected features. As we can see, LSPE consistently requires few features to achieve reasonably good results whereas the most of other algorithms need more features. We also note that the change curve of the performance of LSPE is more smooth than ones of the most of other algorithms, which indicates that the stability of the features selected by LSPE is superior to ones of other algorithms. Moreover, from the results in Fig. 2, it is easy to conclude that more features do not lead to the best results. This may be caused by the adding of redundant features when we select more features. Table 3 gives the best results of *NMI* of different algorithms on the range of selected features. For Sonar, BC and Ionosphere data sets, the values of *NMI* are so small that they are not persuasive. We, here, give the mean *NMI* with standard deviation on three data sets. A big value of *NMI* implies good performance. LSPE always outperforms all its competitors. Table 4 gives the best clustering results of LDMGI on three data sets. We can also see that LSPE can obtain the best performance by using LDMGI. This indicates that LSPE selects the best effective features compared with other methods.

Table 7
Classification error on BC data set (MEAN ± STD%).

Method	One third train	One second train	Two third train
All features	11.39 ± 1.34	11.21 ± 1.52	10.50 ± 1.78
LapScore	11.33 ± 1.46	9.83 ± 1.61	9.19 ± 1.38
SPEC	9.80 ± 1.29	8.79 ± 1.20	8.20 ± 1.82
LSPE	8.50 ± 1.40	7.30 ± 1.57	6.90 ± 1.37

Table 8
Classification error on Ionosphere data set (MEAN ± STD%).

Method	One third train	One second train	Two third train
All features	17.97 ± 2.27	16.80 ± 2.50	16.50 ± 2.40
LapScore	17.58 ± 2.45	15.90 ± 2.23	14.95 ± 2.26
SPEC	17.07 ± 2.74	14.81 ± 2.68	14.01 ± 2.82
LSPE	16.00 ± 2.10	14.00 ± 2.22	13.00 ± 2.12

Table 9
Classification error on Vehicle data set (MEAN ± STD%).

Method	One third train	One second train	Two third train
All features	36.89 ± 1.68	35.23 ± 1.72	34.31 ± 2.23
LapScore	36.62 ± 1.77	34.46 ± 1.93	33.64 ± 2.39
SPEC	35.79 ± 1.82	33.99 ± 1.62	33.17 ± 2.28
LSPE	32.81 ± 1.90	30.56 ± 1.76	29.69 ± 2.06

Next, we substitute the locality preserving with the sparsity preserving in our method (SSPE) and compare these two methods on *K*-means clustering. The comparative results can be found in Table 5. The dimension of the data set is reduced to 189 (ORL), 266 (Isolet), and 105 (Yale) dimensions by PCA. LSPE obtains the better results on ORL and Yale data sets without sacrificing slight performance on Isolet data set.

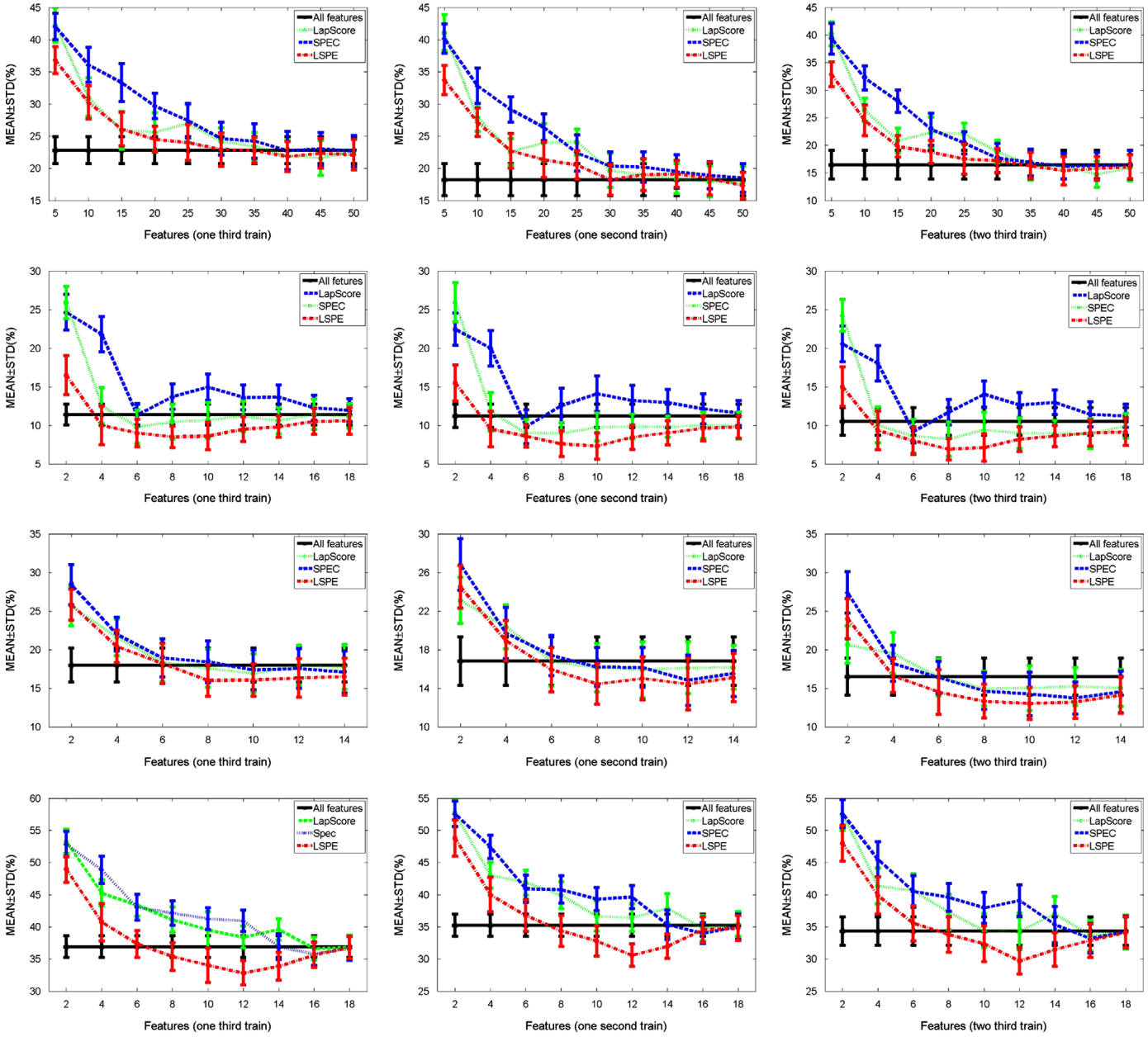

Fig. 3. The detailed classification performance of different algorithms on Sonar (the first row), BC (the second row), Ionosphere (the third row) and Vehicle (the fourth row).

Table 10
Classification error (MKL) of LSPE on four data sets (MEAN $\pm$ STD%).

Method	One third train	One second train	Two third train
Sonar	16.33 ± 2.51	13.00 ± 1.57	10.28 ± 2.14
BC	6.50 ± 1.41	5.20 ± 1.51	3.80 ± 1.90
Ionosphere	4.00 ± 2.83	3.89 ± 2.18	3.34 ± 1.27
Vehicle	18.21 ± 1.50	15.53 ± 1.02	14.20 ± 2.21

5.3. *Classification results with NN classifier and MKL method*

In the second group experiment, we carry out NN algorithm and MKL method with selected different features on the some data sets. In order to evaluate the experimental results better, for each data set, we randomly choose one third, one second, and two third of the total samples as training set and the rest are used as test set. The experiments are repeated 100 times on the best parameter combination. The mean classification error with standard deviation (MEAN $\pm$ STD%) is used as the final result.

The best results on range of selected features are shown in Tables 6–9. The detailed classification performance for each selected feature is presented in Fig. 3. As can be seen from these tables, in the range of the selected features, the best results of FSPE are better than those of other algorithms. However, from the results in Fig. 3, it would be interesting to note that the stability of features selected by LSPE is consistently better than all the other algorithms. However, the change curves of the classification performance of LapScore and SPEC are very volatile. This is attributed to the using of the locality and similarity perverting [29]. Moreover, we also notice that LSPE obtains reasonable results with less features. For example, on BC and Ionosphere data sets, FSPE obtains the reasonably results with typically around 4 and 6 features, respectively. For the other three algorithms, they usually require more features to achieve a reasonable result. It is easy conclude that LSPE can achieve better classification

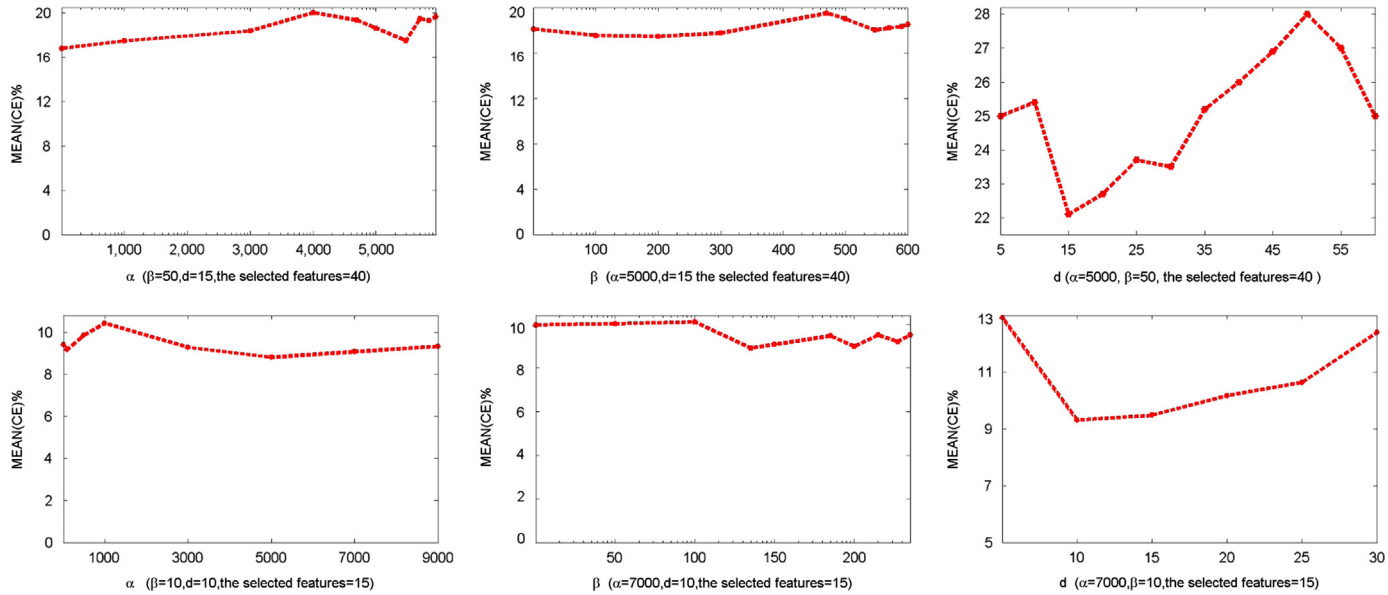

Fig. 4. The mean classification error variation of LSPE versus different parameters on Sonar (the first row) and BC (the second row).

performance with the least amount of features. In other words, the features selected by LSPE have better discriminant ability than those by using other algorithms. Table 10 represents the best classification results of LSPE on four data sets by using MKL method. We can see that, compared with NN, the classification error is reduced significantly by using MKL method. This confirms that the features selected by LSPE have good discriminability.

5.4. Parameters selection

In the third group experiment, there are five parameters, *i.e.*, k, σ, α, β, and d. In our method, we can obtain reasonable results when we tune k from $\{5, 10\}$ and set $\sigma = 1.0$ for all data sets. Therefore, in this section, we do not discuss these two parameters. It is time-consuming to select α, β, and d based on the gird search. Fortunately, α and β affect the performance of LSPE slightly if they are set in feasible range. However, the performance of LSPE is comparatively sensitive to d, the dimensionality of the low dimensional embedding. We set the range of d as $[(\frac{1}{5}) \times (\#f), (\frac{1}{2}) \times (\#f)]$, where $\#f$ is the number of the features of a data set. We select two data sets, *i.e.*, Sonar and BC, and perform NN algorithm on these two data sets to validate this strategy. We randomly choose one second of the total samples as training set and the rest are used as test set. These trails are independently performed 100 times, and the mean classification error (MEAN(CE)%) is reported. Fig. 4 shows this strategy works well on the selected two data sets. The performance is consistent when each of α and β is selected from a wide range. Specifically, for Sonar and BC, we respectively set $d = (\frac{1}{4}) \times (\#f) = (\frac{1}{4}) \times 60 = 15$ and $d = (\frac{1}{3}) \times (\#f) = (\frac{1}{3}) \times 30 = 10$. Form Fig. 4, we can see that the performance of LSPE is not very sensitive to α and β in the wide range, when we fix d. However, the performance is comparatively sensitive to d, when we fix α and β. Moreover, we also see that, for Sonar, a valley appears when $d = 15$. For BC (Fig. 4 (the second row)), there appears a valley when $d = 10$. This indicates that the proposed method performs well under this setting $d \in [(1/5) \times (\#f), (1/2) \times (\#f)]$. To our knowledge, previous literature does not propose a very feasible method to resolve the problem that how to determine the suitable number of the selected features, and thus, in this experiment, it is set by experience. For example, Fig. 4 gives the performance of LSPE versus α or β with the number of the selected features fixed to 40 and 15 for Sonar and BC, respectively.

6. Conclusion

In this paper, we propose a novel feature selection method, *i.e.*, locality and similarity preserving embedding (LSPE) for feature selection, which unifies embedding learning and feature selection. We introduce an iterative algorithm to optimize LSPE and theoretically show its convergence. LSPE seeks an optimal transformation matrix by determining the sparse reconstruction coefficient matrix and transformation matrix simultaneously. The major advantage of the proposed LSPE method is that the selected features have good stability by preserving locality and similarity among data points. Moreover, LSPE trends to select discriminative features because of the sparsity, which leads LSPE to achieve better performance with the least amount of features. In the future, we attempt to extend LSPE to the supervised case for obtaining better performance.

Acknowledgment

This work is supported by the National Basic Research Program of China (973 Program) (Grant No. 2012CB316400), by the National Natural Science Foundation of China (Grant Nos 61125106 and 91120302), and by the Shaanxi Key Innovation Team of Science and Technology (Grant No. 2012KCT-04).

References

[1] Z. Zhao, L. Wang, H. Liu, J.P. Ye, On similarity preserving feature selection, IEEE Trans. Knowl. Data Eng. 24 (March (3)) (2013) 619–632.
[2] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extension: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.
[3] J.B. Yang, C.J. Ong, An effective feature selection method via mutual information estimation, Cybern. B Cybern. 42 (6) (2012) 1550–1559.
[4] F.P. Nie, H. Huang, X. Cai, Chris H.Q. Ding, Efficient and robust feature selection via joint $\ell_{2,1}$−norm s minimization, Adv. Neural Inf. Process. Syst. (2012) 1813–1821.
[5] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[6] R. Diao, Q. Shen, Two new approaches to feature selection with harmony search, in: Proceedings of 19th International Conference on Fuzzy Systems, 2010, pp. 3161–3167.
[7] Z.C. Li, Y. Yang, J. Liu, X.F. Zhou, H.Q. Lu, Unsupervised feature selection using nonnegative spectral analysis, in: Proceedings of 26th AAAI Conference on Artificial Intelligence, 2012, pp. 1026–1032.
[8] X.F. He, D. Cai, P. Niyogi, Laplacian score for feature selection, Adv. Neural Inf. Process. Syst. 18 (2005).

[9] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd ed., Wiley-Interscience, New York, 2000.
[10] K.Z. Mao, Identifying critical variables of principal components for unsupervised feature selection, IEEE Trans. Syst. Man Cybern. B Cybern. 35 (2) (2005) 339–344.
[11] R. Battiti, Using mutual information for selection features in supervised neural net learning, IEEE Trans. Neural Netw. 5 (4) (1994) 537–550.
[12] M. Vasconcelos, N. Vasconcelos, Natural image statistics and low complexity feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 228–244.
[13] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[14] A. Rakotomamonjy, Variable selection using svm-based criteria, J. Mach. Learn. Res. 3 (2003) 1357–1370.
[15] Y.J. Sun, S. Todorovic, S. Goodison, Local learning based feature selection for high dimensional data analysis, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2005) 1610–1626.
[16] P. Pudil, J. Novovicov, Novel methods for subset selection with respect to problem knowledge, IEEE Intell. Syst. 13 (2) (1998) 66–74.
[17] L. van't, H. Dai, M. van de Vijver, et al., Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 536–550.
[18] Y. Wang, J. Klijin, Y. Zhang, et al., Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer, Lancet 365 (2005) 671–679.
[19] T.Y. Zhou, D.C. Tao, Double shrinking sparse dimension reduction, IEEE Trans. Image Process. 22 (1) (2001) 668–674.
[20] T.Y. Zhou, D.C. Tao, X.D. Wu, Manifold elastic net: a unified framework for sparse dimension reduction, Data Min. Knowl. Discov. 46 (1) (2002) 131–159.
[21] T.Y. Zhou, D.C. Tao, GoDec: randomized lowrank & sparse matrix decomposition in noisy case, in: ICML 11: Proceedings of the 28th International Conference on Machine Learning, 2011.
[22] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[23] J. Tenenbaum, V. de Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[24] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[25] Y. Bengio, J.F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, M. Ouimet, Out-of-sample extensions for LLE, isomap, mds, eigenmaps, and spectral clustering, Adv. Neural Inf. Process. Syst. (2003) 177–184.
[26] X.F. He, S.C. Yan, Y.X. Hu, P. Niyogi, H.J. Zhang, Face recognition using Laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.
[27] J. Gui, W. Jia, L. Zhu, S.L. Wang, D.S. Huang, Locality preserving discriminant projections for face and palmprint recognition, Neural Comput. 73 (2010) 2696–2707.
[28] X.F. He, D. Cai, S.C. Yan, H.J. Zhang, Neighborhood preserving embedding, in: Proceedings of the Tenth IEEE International Conference on Computer Vision, 2007, pp. 823–830.
[29] Y. Xu, D. Zhang, J. Yang, J.-Y. Yang, A two-phase test sample sparse representation method for use with face recognition, IEEE Trans. Circuits Syst. Video Technol. 21 (9) (2011) 1255–1262.
[30] S.H. Gao, I.W. Tsang, L.-T. Chia, Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 92–104.
[31] B. Cheng, J.C. Yang, S.C. Yan, Y. Fu, T.S. Huang, Learning with ℓ_1−graph for image analysis, IEEE Trans. Image Process. 19 (2010) 858–866.
[32] L.S. Qiao, S.C. Chen, X.Y. Tan, Sparsity preserving projections with applications to face recognition, Pattern Recognit. 43 (2010) 331–341.
[33] K. Yu, T. Zhang, Y.H. Gong. Nonlinear learning using local coordinate coding. In: Proc. of NIPS'09, 2009.
[34] J.J. Wang, J.C. Yang, K. Yu, F.J. LV, T. Huang, Y.H. Gong, Locality-constrained linear coding for image classification, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2010, pp. 13–18.
[35] C. Ding, D. Zhou, X.F. He, H.Y. Zha, R_1-PCA: rotational invariant ℓ_1−norm principal component analysis for robust subspace factorization, in Proceedings of 23rd International Conference on Machine Learning, ACM, 2006, pp. 281–288.
[36] F. Bach, Consistency of the group lasso and multiple kernel learning, J. Mach. Learn. Res. 9 (2008) 1179–1225.
[37] H. Huang, C. Ding, Robust tensor factorization using ℓ_1−norm, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[38] D. Cai, X.F. He, J.W. Han, Spectral regression: a unified approach for sparse subspace learning, in: Proceedings of 2007 International Conference on Data Mining (ICDM'07), Omaha, NE, October 2007.
[39] Y. Xu, D. Zhang, F.X. Song, J.Y. Yang, Z. Jing, M. Li, A method for speeding up feature extraction based on KPCA, Neural Comput. 70 (2007) 1056–1061.
[40] Y. Xu, D. Zhang, J. Yang, Z. Jin, J.Y. Yang, Evaluate dissimilarity of samples in feature space for improving KPCA, Int. J. Inf. Technol. Decision Making 10 (3) (2011) 479–495.
[41] Y. Xu, A.N. Zhong, J. Yang, D. Zhang, LPP solution schemes for use with face recognition, Pattern Recognit. 43 (12) (2010) 4165–4176.
[42] Y. Xu, Q. Zhu, Z.Z. Fan, M.N. Qiu, Y. Chen, H. Liu, Coarse to fine K nearest neighbor classifier, Pattern Recognit. Lett. 34 (2013) 980–986.
[43] Z.H. Lai, W. Wong, Z. Jin, J. Yang, Y. Xu, Sparse approximation to the eigensubspace for discrimination, IEEE Trans. Neural Netw. Learn. Syst. 23 (12) (2012) 1948–1960.
[44] S.H. Gao, I.W. Tsang, L.-T. Chia, P. Zhao, Local features are not lonely Laplacian sparse coding for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3555–3561.
[45] J. Liu, D. Cai, F.X. He, Gaussian mixture model with local consistency, in: Proceedings of 24th Conference on Artificial Intelligence, 2010, pp. 512–517.
[46] 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.
[47] C.P. Hou, F.P. Nie, D.Y. Yi, Y. Wu, Feature selection via joint embedding learning and sparse regression, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 1324–1329.
[48] Y. Yang, D. Xu, F.P. Nie, S.C. Yang, Y.T. Zhang, Image clustering using local discriminant models and global integration, IEEE Trans. Image Process. 19 (10) (2010) 2761–2773.
[49] L.X. Duan, I.W. Tsang, D. Xu, Domain transfer multiple kernel learning, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 465–479.
[50] L. Duan, D. Xu, I.W. Tsang, J. Luo, Visual event recognition in videos by learning from web data, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1667–1680.
[51] D. Cai, C. Zhang, X.F. He, Unsupervised feature selection for multi-cluster data, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 333–342.
[52] 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.
[53] [Online]. Available: ⟨http://archive.ics.uci.edu/ml/machine-learning-databases/undocumented/connectionist-bench/sonar/⟩.
[54] [Online]. Available: ⟨http://archive.ics.uci.edu/ml/machine-learning-databases/breastcancer-wisconsin/⟩.
[55] [Online]. Available: ⟨http://bengio.abracadoudou.com/lectures/dbases/index.html⟩.
[56] [Online]. Available: ⟨http://bengio.abracadoudou.com/lectures/dbases/vehicle⟩.
[57] S.J. Wang, J. Yang, M.F. Sun, X.J. Peng, M.M. Sun, C.G. Guang, Sparse tensor discriminant color space for face verification, IEEE Trans. Neural Netw. 23 (6) (2012) 876–887.
[58] L. Lovsz, M. Plummer, Matching Theory, North Holland, Amsterdam, The Netherlands, 1986.

Xiaozhao Fang received the M.S. degree in computer science from Guangdong University of Technology, Guangzhou, China, in 2008. He is currently pursuing the Ph.D. degree in computer science and technology at Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. He has published more than seven journal papers. His current research interests include pattern recognition and machine learning.

Yong Xu was born in Sichuan, China, in 1972. He received his B.S. and M.S. degrees in 1994 and 1997, respectively. He received the Ph.D.degree in pattern recognition and intelligence system at NUST (China) in 2005. Now he works at Shenzhen Graduate School, Harbin Institute of Technology. He is an IEEE member. His current interests include pattern recognition, biometrics, machine learning and video analysis.

Xuelong Li is a Full Professor with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, Shaanxi, PR China.

Zizhu Fan received the M.S. degree in computer science from Hefei University of Technology, Hefei, China, in 2003. Now he is working for his Ph.D. degree in computer science & technology at Shenzhen Graduate School, Harbin Institute of Technology (HIT). His current interests include pattern recognition and machine learning. He has published more than 10 journal papers.

Hong Liu received Bachelor degree in computer science in 1990, Master degree in computer science in 1993 and Doctor degree in electrical control and automation engineering in 1996, post-doctoral in computer science and technology in 1996. Professor Liu is currently the supervisor of doctoral students, the director of Research Department of Shenzhen Graduate School and director of Intelligent Robot Laboratory of Peking University. He is also an IEEE member, an executive director and vice secretary-Intelligent Automation Committee of Chinese Automation Association (IACAA). His expertise is in the areas of image processing and pattern recognition, intelligent robots and computer vision, intelligent micro-systems hardware and software co-design. He has published more than 100 papers in the important scholarly journals and international conferences, and access to subsidized Korean Academy of an Jung-geun Awards, Department of Space Science and Technology Progress Award, and Peking University Teaching Excellence Award, Aetna award and candidates of Peking University Top Ten Teachers. He has done exchange visits in many famous universities and research institutions in several countries and regions, including the United States, Canada, France, the Netherlands, Japan, Korea, Singapore, Hong Kong and so on.

Yan Chen received her B.E and M.E degrees in computer science from Northeastern University, China, in 1997 and 2000, respectively, and her Ph.D. in 2010 from University of Technology, Sydney (UTS), Australia. Currently, she is a Post-Doctoral Researcher with Harbin Institute of Technology (HIT) at Shenzhen, China. Her research interests include computer vision and pattern recognition.