Multimed Tools Appl (2018) 77:29605–29622
https://doi.org/10.1007/s11042-017-5381-7

Unsupervised feature selection via local structure learning and sparse learning

Cong Lei[1] · **Xiaofeng Zhu**[1]

Received: 31 July 2017 / Revised: 24 October 2017 / Accepted: 1 November 2017 /
Published online: 28 November 2017

Abstract Feature self-representation has become the backbone of unsupervised feature selection, since it is almost insensitive to noise data. However, feature selection methods based on feature self-representation have the following drawbacks: 1) The self-representation coefficient matrix is fixed and can not be fine-tuned according to the structure of data. 2) they do not consider the manifold structure of data, thus unable to further increase the performance of feature selection. To solve the above problems, this paper proposes an unsupervised feature selection algorithm that combines feature self-representation and manifold learning. Specifically, we first utilize feature self-representation to construct the model. After that, the self-representation coefficient matrix is dynamically adjusted to the optimal state based on the similarity matrix. Then, we use low-rank representation to explore the global manifold structure of the data. Finally, we combine sparse learning with feature selection. The experimental results on twelve datasets show that the proposed method outperforms all the competing methods.

Keywords Feature selection · Subspace learning · Sparse feature selection · Hypergraph representation

1 Introduction

With the rapid development of information technology and database technology, people can easily access and store huge amount of data. Traditional data analysis tools can not satisfy

✉ Xiaofeng Zhu
seanzhuxf@gmail.com

[1] Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin, 541004, Guangxi, People's Republic of China

people's increasing needs any more [10, 42, 66]. In order to efficiently extract useful information from massive data, many data mining technologies have been around. In real applications, however, data usually is high-dimensional and may contain lots of redundant and useless information, which seriously undermines effectiveness of the data mining methods [13, 17, 19]. Therefore, it is crucial to conduct dimensionality reduction on the data before applying any data mining algorithm [15, 16].

Dimensionality reduction is one of the most important research areas of machine learning [9, 21, 41]. In real applications, data is often high-dimensional, and it is difficult to be understood, represented and processed [20, 37, 39]. By employing dimensionality reduction methods to break the *curse of dimensionality*, people can easily process and thus fully understand the data [11, 29, 49]. There are a number of ways to reduce dimensions of the data, which can be divided into two categories. The first one contains linear methods, such as Principal Component Analysis (PCA) [50] and Classical Multidimensional Scaling (CMDS) [52]. Low-dimensional data obtained by these linear methods can usually maintain the linear relationship between high-dimensional data points, but this kind of methods can not reveal complex nonlinear manifold structure of the data. The second one, based on manifold learning theory, consists of non-linear methods, such as Isometric Mapping (Isomap) [64], Local Linear Embedding (LLE) [63]. This type of algorithms can learn inherent geometry structure of the data, facilitating data processing and analysis [16, 24].

Dimensionality reduction methods based on manifold learning can usually be divided into two categories. One is based on local manifold structure of data. The most classical method is Locality Preserving Projections (LPP) [62], which can preserve neighborhood relations between data after dimensionality reduction. Another consists of methods that consider global structure of the data. Linear Discriminant Analysis (LDA) [7] is the most commonly used in many areas and it can produce the best projection result, meaning that after projection samples with the same class label have the largest distance while those with different class labels have the smallest distance [18, 40, 61, 63]. That is, LDA produces the best capability of separability for data samples. However, most feature selection algorithms only take into account one of the two manifold structures of the data, i.e., either global or local, so the performance is not satisfactory [12, 28, 30].

Because of some good properties, such as insensitive to noise data, feature self-representation has been widely used in machine learning and computer vision [60, 65]. However, since the self-representation coefficient matrix obtained by these methods is fixed, feature self-representation based methods do not adapt well to data with complex structure. To this end, this paper presents a feature selection algorithm that combines structure learning with sparse learning (LSS_FS). The main contributions of this paper include

- Due to the fact that our method uses feature self-representation to build the model, this method is not very sensitive to noise and outliers, exhibiting good robustness.
- This model utilizes a low-rank constraint on self-representation coefficient matrix to explore global manifold structure of the data, which had been proven that it has the ability to take advantage of the correlation among features effectively.
- The model proposed in this paper can dynamically adjust the structure of data according to the similarity matrix of the data, and can make it more accurate, so as to improve the classification accuracy.
- In this paper, we propose a novel method to solve the objective function. We perform low-rank feature selection and dynamically adjust the graph matrix to optimize the

objective function according to the similarity matrix iteratively. The optimal results are achieved in each iteration, and finally the global optimal solution is thus obtained.

2 Related work

The original purpose of sparse learning is to compress and express a signal, because it has a lower sampling rate than Shannon theorem. And it gradually becomes popular, and successfully used to solve many practical problems. For example, in signal and image processing, signal encoding, representation and compression, image denoising, image super-resolution, etc [34, 48, 57].

With the continuous innovation and practice of many scientists and researchers around the world, sparse learning theory has been successfully applied in signal processing, data mining, machine learning, information retrieval, pattern recognition, biological computing and other fields. And it has become the research hotspot of these areas. So far, a large number of experts and scholars are still carrying out an in-depth study of sparse learning's theory and application. For sparse learning itself, because it has a more natural discriminant nature, it is more suitable for face recognition [27, 36]. Specifically, based on the theory of Compressive Sensing, by using the sparse learning to reconstruct an approximate signal for the original one to find the dimensionality reduce matrix, so that the original high-dimensional signal projection to low-dimensional space can also maintain its original features as much as possible [32, 35]. Sparse learning methods have achieved very high recognition accuracy in face recognition. Although this method has achieved good performance, it needs to store all training images for recognition, significantly increasing the storage overhead [43, 46]. Moreover, when the sparse learning deals with independent signals, it only considers the inherent relevance of the signal, ignoring the correlation between the same type of signals.

Sparse learning evaluates the importance of an element in terms of the weight of its coefficient, that is, the weight of an unimportant element is zero and the important one is nonzero [31]. The more important an element, the greater the corresponding weight. Therefore, sparse learning can be used as a natural identification information into the model [55] by using the coefficient weights between the data samples or features. This enables sparse learning to reduce the impact of noise on the model and improve the efficiency of the learning model. Therefore, sparse learning has been widely used in the field of data mining and machine learning. For example, Zhu et al. proposed a joint sparse learning method to achieve block sparse of data by considering the correlation between data and global information of the data, and it has been used to deal with multi-label data for classification. Gao et al. used the histogram intersection kernel (HIk) to implement kernel sparse learning, and it utilizes the HIk basis to consider a feature quantization of soft-allocated extended to conduct sparse coding [8]. Xia et al. proposed a sparse projection algorithm to conduct binary coding for high-dimensional data [27]. Sparse Subspace Learning (SSL) can also be seen as a special dimensionality reduction method [2]. Some researchers proposed a series of sparse subspace learning algorithms based on sparse learning theory, including Nonnegative Sparse Principal Component Analysis (NSPCA) [45], Sparse Nomegative Matrix Factorization (SNMF) [23], and so on.

The essence of sparse learning is to introduce the coefficient weight between samples or features as the authentication information into the model. By using the sparse constraint to punish the model, the coefficient weight of some irrelevant data approaches zero, and useful

information is preserved, so the sparse learning is robust [52]. However, sparse learning does not take into account the internal structure of the data.

To overcome the shortcomings of sparse learning, manifold learning has been proposed. Linear discriminant analysis is the most common technique used in manifold learning. Zhong et al. proposed a method to utilize ℓ_1-norm to maximize the objective function (LDA-ℓ_1) [47], where LDA-ℓ_1 captures the global manifold structure via discriminant analysis, and then selects the most critical features. Unsupervised Feature Selection via Unified Trace Ratio Formulation and K-means Clustering (TRACK) was proposed as a unified framework for feature selection and selected discriminative features according to trace ratio formulation and K-means clustering [25]. Cai et al. proposed Multi-Cluster Feature Selection (MCFS), which explores local manifold structure of the data via spectral analysis and then searches the features that can better preserve the clustering structure [3]. However, these algorithms are sensitive to outliers, and if the data contains too much noise, performance of the algorithm will degenerate significantly.

There are many robust feature selection methods. For example, Unsupervised feature selection by regularized self-representation (RSR), which uses all the features to linearly represent each feature, thus weakening the impact of outliers [56]. Sun et al. put forward an Unsupervised Robust Bayesian Feature Selection method, where the feature selection capability is realized by estimating the feature saliencies associated with the features [22]. But these methods do not consider the manifold structure of data, so their performance are not satisfactory. To obtain better stability and performance, this paper combines manifold learning with feature self-representation, and then introduces the sparse regularization factor to conduct feature selection.

3 Our method

In this section, we first introduce some notations that are used in this paper, and then explain the detail of the proposed LSS_FS method, in Sections 3.1 and 3.2, respectively, and then elaborate the proposed optimization method in Section 3.3. Finally, we analysis the convergence of the objective function in Section 3.4.

3.1 Notations

For a matrix $X \in \mathbb{R}^{n\times d}$, its i-th row and j-th column are denoted as X^i and X_j, respectively. The trace of a matrix X is denoted as $\mathrm{tr}(X)$. X^T means the transpose of X and X^{-1} means the inverse of X, respectively. Also we denote the ℓ_F-norm and $\ell_{2,1}$-norm of X respectively as $||X||_F = \sqrt{\sum_i^n ||x^i||_2^2} = \sqrt{\sum_j^d ||x_j||_2^2}$, $||X||_{2,1} = \sum_i^n \sqrt{\sum_j^d x_{i,j}^2}$.

3.2 Structure learning for feature selection

Many previous literatures have indicated that the local structures of the samples may provide complementary information to boost the ability of dimensionality reduction [44, 54]. So, this paper proposes to utilize the local structures of the samples by learning a graph matrix $S \in \mathbb{R}^{n\times n}$ on a low-dimensional space of the original data. Given the feature matrix X, we can get the following formula according to [51, 59]

$$\min_Z \sum_{i,j}^n ||x^i Z - x^j Z||_2^2 s_{i,j} \tag{1}$$

where $||\cdot||_2$ is the ℓ_2-norm of a vector, $Z \in \mathbb{R}^{d\times d}$ is a transformation matrix of the high-dimensional data X in its low-dimensional space, and the element $s_{i,j}$ of the graph matrix S represents the similarity between sample X^i and sample X^j. If sample X^i is one of the k-nearest neighbors of sample X^j, then the value of the heat kernel, i.e., $f(x^i, x^j) = exp(-\frac{||x^i - x^j||_2^2}{2\sigma^2})$ where σ is a tuning parameter, is regarded as the value of $s_{i,j}$; otherwise $s_{i,j} = 0$.

Actually, the character of S has been demonstrated very sensitive to σ [49, 53]. This inspires us to learn the relatively correct graph matrix from the 'clean' data and to decrease number of parameters that need to adjust. By clean, we mean that a low-dimensional space with as less noise and redundancy as possible [33]. However, in real applications, we cannot know the graph matrix and low-dimensional space in advance [38]. In order to deal with these problems, we combine graph matrix learning with low-dimensional space learning to iteratively optimize them so as to achieve the optimal results. Therefore, we may learn the graph matrix by following the distribution of the samples. Then we have the following objective function:

$$\begin{aligned} &\min_{S,Z} \sum_{i,j}^{n} (||x^i Z - x^j Z||_2^2 s_{i,j} + \lambda_1 ||s_i||_2^2), \\ &s.t., \forall i, s_i^T \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0, \text{if} j \in \mathcal{N}(i), \text{otherwise } 0 \end{aligned} \tag{2}$$

where λ_1 is a control parameter and s_i is the i-th column of S, $||s_i||_2^2$ is used to avoid the trivial solution, $\mathbf{1}$ and $\mathcal{N}(i)$ represent an all-one-element vector and the set of the nearest neighbors of the i-th sample, respectively, and the constraint $s_i^T \mathbf{1} = 1$ is applied to gain the shift invariant similarity. Clearly, (2) assigns small value (i.e., similarity) to $s_{i,j}$ if sample i and j are far apart, and large value to $s_{i,j}$ otherwise.

Previous methods learn the graph matrix via (1) to generate an optimal similarity measurement. Different from that, (2) aims to achieve both optimal similarity measurement (i.e., S) and feature selection results (i.e., Z). It becomes obvious that (2) may receive better feature selection results (i.e., Z) than (1).

In order to reduce the adverse impacts of outliers and noise samples on the model, we propose to use the self-representation method to build an unsupervised model, then we can get the following functions:

$$\begin{aligned} &\min_{S,Z} \sum_{i,j}^{n} ||x^i Z - x^j Z||_2^2 s_{i,j} + \lambda_1 ||s_i||_2^2 + \lambda_2 ||X - XZ||_F^2 \\ &s.t., \forall i, s_i^T \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0, \text{if} j \in \mathcal{N}(i), \text{otherwise } 0 \end{aligned} \tag{3}$$

In order to improve effectiveness of the algorithm and to consider relationship among features, we may impose a constrain on the rank of Z [58]. Through this process, a low rank constraint on Z can naturally be represented as the product of two r-rank matrices as follows:

$$Z = AB \tag{4}$$

where $A \in \mathbb{R}^{d\times r}$, $B \in \mathbb{R}^{r\times d}$, $r \leq \min(n, d)$, and our function becomes

$$\begin{aligned} &\min_{S,A,B} \sum_{i,j}^{n} ||x^i AB - x^j AB||_2^2 s_{i,j} + \lambda_1 ||s_i||_2^2 + \lambda_2 ||X - XAB||_F^2 \\ &s.t., \forall i, s_i^T \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0, \text{ if } j \in \mathcal{N}(i), \text{otherwise } 0 \end{aligned} \tag{5}$$

According to [56], the feature selection issue can be converted to the following form:

$$\min_W l(X - XW) + \lambda R(W) \tag{6}$$

where W is the feature weight matrix, $l(X - XW)$ is the loss function, $R(W)$ is the regularization on W and λ is a positive constant. In the item $||X - XAB||_F^2$, by treating XA as X of (6), then B can be naturally regarded as W of (6). Inspired by (6), we naturally make the regularization on matrix of B to further improve the performance of the algorithm. Therefore, our ultimate objective function is:

$$\begin{aligned}&\min_{S,A,B} \sum_{i,j}^{n} ||x^i AB - x^j AB||_2^2 s_{i,j} + \lambda_1 ||s_i||_2^2 + \lambda_2 ||X - XAB||_F^2 + \lambda_3 ||B||_{2,1} \\ &s.t., \forall i, s_i^T \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0, \text{if} j \in \mathcal{N}(i), \text{otherwise} 0, A^T A = I \end{aligned} \tag{7}$$

The constraint $A^T A = I$ (where $A \in \mathbb{R}^{d \times r}$ and $I \in \mathbb{R}^{r \times r}$) is introduced for identifiability purpose, where λ_1, λ_2 and λ_3 are tuning parameters.

3.3 Optimization

Equation (7) is not jointly convex with respect to all the variables (i.e., A, B, and S), but it is convex for each variable while fixing the rest. In this paper, we utilize the alternative optimization method to optimize (7), i.e., iteratively optimizing each variable respectively.

Update A by fixing B and S

When B and S are fixed, the second and fourth terms of (7) can be viewed as constants, thus we can get:

$$\min_{A} \sum_{i,j}^{n} ||x^i AB - x^j AB||_2^2 s_{i,j} + \lambda_2 ||X - XAB||_F^2, s.t., A^T A = I \tag{8}$$

By following the IRLS framework, we rewrite (8) as:

$$\begin{aligned}&\min_{A} tr(B^T A^T X^T L X A B) + \lambda_2 tr(X^T X - X^T X A B \\ &- B^T A^T X^T X + B^T A^T X^T X A B), s.t., A^T A = I \end{aligned} \tag{9}$$

where $tr(\cdot)$ is a trace operator, $L = Q - S \in \mathbb{R}^{n \times n}$ is a Laplacian matrix and Q is a diagonal matrix with its i-th element $q_{i,i} = \sum_{j=1}^{n} s_{i,j}$. By fixing B, we get the derivative of (9) as follows:

$$2X^T L X A B B^T - 2\lambda_2 X^T X B^T + 2\lambda_2 X^T X A B B^T \tag{10}$$

Since A is orthogonal, we can employ an existing method in [26] to optimize it.

Update B by fixing A and S

When A and S are fixed, the second term of (7) can be viewed as constants, we can get:

$$\min_{B} \sum_{i,j}^{n} ||x^i AB - x^j AB||_2^2 s_{i,j} + \lambda_2 ||X - XAB||_F^2 + \lambda_3 ||B||_{2,1} \tag{11}$$

Which is equivalent to

$$\begin{aligned}&\min_{B} tr(B^T A^T X^T L X A B) + \lambda_2 tr(X^T X - X^T X A B \\ &- B^T A^T X^T X + B^T A^T X^T X A B) + \lambda_3 tr(B^T P B) \end{aligned} \tag{12}$$

Where $P \in \mathbb{R}^{r \times r}$ is diagonal matrix with $p_{i,i} = \frac{1}{2||B^i||_2}$, $(i = 1, \ldots, r)$. Then setting the derivative of B in (12) to zero, we obtain:

$$B = \lambda_2 (A^T X^T L X A + \lambda_2 A^T X^T X A + \lambda_3 P)^{-1} A^T X^T X \tag{13}$$

Update S by fixing A and B

Given that A and B are fixed, (7) is changed to:

$$\min_{S} \sum_{i,j}^{n} ||x^i AB - x^j AB||_2^2 s_{i,j} + \lambda_1 ||s_i||_2^2$$
$$s.t., \forall i, s_i^T \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0, \text{if} j \in \mathcal{N}(i), \text{otherwise } 0 \tag{14}$$

We first calculate the Euclidean distance between each sample to produce k nearest neighbors of all samples, and then set the value of $s_{i,j}$ to 0 if the j-th sample does not belong to one of the k nearest neighbors of the i-th sample, otherwise, we utilize (17) to determine the values of $s_{i,j}$.

Since optimizing S is equal to solely optimizing each vector $s_i (i = 1, \ldots, n)$, we further change the optimization problem in (14) to individually optimize $s_i (i = 1, \ldots, n)$ as follows:

$$\min_{s_i^T 1 = 1, s_{i,i} = 0, s_{i,j} \geq 0} \sum_{j}^{n} (||x^i AB - x^j AB||_2^2 s_{i,j} + \lambda_1 s_{i,j}^2) \tag{15}$$

By denoting $F \in \mathbb{R}^{n \times n}$ where $f_{i,j} = ||x^i AB - x^j AB||_2^2$, we rewrite (15) as follows:

$$\min_{s_i^T 1 = 1, s_{i,i} = 0, s_{i,j} \geq 0} ||s_i + \tfrac{1}{2\lambda_1} f_i||_2^2 \tag{16}$$

On the basis of the Karush-Kuhn-Tucker (KKT) conditions [1], we are able to get the closed-form solution of $s_{i,j} (j = 1, \ldots, n)$ as:

$$s_{i,j} = (-\tfrac{1}{2\lambda_1} f_{i,j} + \tau)_+ \tag{17}$$

We suppose that there are k nearest neighbors for each sample. By denoting $\hat{f}_i = \{\hat{f}_{i,1}, \ldots, \hat{f}_{i,n}\}$ as a descend order of $f_i, (i = 1, \ldots n)$, (17) reveals that $s_{i,k+1} = 0$ and $s_{i,k} > 0$, where k is the number of nearest neighbors of the i-th samples and can be tuned by cross-validation methods. That is,

$$-\tfrac{1}{2\lambda_1} \hat{f}_{i,k+1} + \tau \leq 0 \tag{18}$$

Since $s_i^T \mathbf{1} = 1$, we can get

$$\sum_{j=1}^{k} (\tfrac{1}{2\lambda_1} \hat{f}_{i,k} + \tau) = 1 \Rightarrow \tau = \tfrac{1}{k} + \tfrac{1}{2k\lambda_1} \sum_{j=1}^{k} \hat{f}_{i,k} \tag{19}$$

Based on the above discussion, we summarize the process for solving (7) in Algorithm 1.

Algorithm 1 Pseudo code for solving (7).

Input: $X \in \mathbb{R}^{n \times d}$, λ_1, λ_2, λ_3 and r;
Output: $A \in \mathbb{R}^{d \times r}$, $B \in \mathbb{R}^{r \times d}$, and $S \in \mathbb{R}^{n \times n}$;
Randomly initialize $A(0)$, $B(0)$, initialize $S(0)$ to zero matrix;
Initialize t=0;
repeat
  Update $A(t+1)$ via (8) and [26];
  Update $B(t+1)$ via (11);
  Update $S(t+1)$ by solving (14);
  Compute the diagonal matrix p as $p_{ii} = \frac{1}{2||B^i||_2}, i = 1, \ldots, r$;
  $t = t+1$;
until *converge*;

3.4 Convergence analysis

The framework IRLS has been proven to converge in [6], so we only need to prove the convergence of Algorithm 1 via Theorem 1:

Theorem 1 *The value of objective function in* (7) *monotonically decreases until Algorithm 1 converges.*

Proof After the t-th iteration, we have obtained the optimal $A^{(t)}$, $B^{(t)}$ and $S^{(t)}$. In the $(t+1)$-th iteration, we need to optimize $S^{(t+1)}$ by fixing $A^{(t)}$ and $B^{(t)}$.

According to (17), $s_{i,j}^{(t+1)}$ has a closed-form solution for all $i, j = 1, \ldots, n$, thus we have:

$$\begin{aligned}
&\sum_{i,j}^{n} ||x^i A^{(t)} B^{(t)} - x^j A^{(t)} B^{(t)}||_2^2 s_{i,j}^{(t+1)} + \lambda_1 ||X - X A^{(t)} B^{(t)}||_F^2 \\
&+\lambda_2 \sum_{i}^{n} ||s_i^{(t+1)}||_2^2 + \lambda_3 ||B^{(t)}||_{2,1} \\
\leq\ &\sum_{i,j}^{n} ||x^i A^{(t)} B^{(t)} - x^j A^{(t)} B^{(t)}||_2^2 s_{i,j}^{(t)} + \lambda_1 ||X - X A^{(t)} B^{(t)}||_F^2 \\
&+\lambda_2 \sum_{i}^{n} ||s_i^{(t)}||_2^2 + \lambda_3 ||B^{(t)}||_{2,1}
\end{aligned} \tag{20}$$

While fixing $S^{(t+1)}$ to update $A^{(t+1)}$ and $B^{(t+1)}$, we follow [6] to get

$$\begin{aligned}
&\sum_{i,j}^{n} ||x^i A^{(t+1)} B^{(t+1)} - x^j A^{(t+1)} B^{(t+1)}||_2^2 s_{i,j}^{(t+1)} \\
&+\lambda_1 ||X - X A^{(t+1)} B^{(t+1)}||_F^2 + \lambda_2 \sum_{i}^{n} ||s_i^{(t+1)}||_2^2 + \lambda_3 ||B^{(t+1)}||_{2,1} \\
\leq\ &\sum_{i,j}^{n} ||x^i A^{(t)} B^{(t)} - x^j A^{(t)} B^{(t)}||_2^2 s_{i,j}^{(t+1)} \\
&+\lambda_1 ||X - X A^{(t)} B^{(t)}||_F^2 + \lambda_2 \sum_{i}^{n} ||s_i^{(t+1)}||_2^2 + \lambda_3 ||B^{(t)}||_{2,1}
\end{aligned} \tag{21}$$

By integrating (20) with (21), we obtain:

$$\begin{aligned}
&\sum_{i,j}^{n} ||x^i A^{(t+1)} B^{(t+1)} - x^j A^{(t+1)} B^{(t+1)}||_2^2 s_{i,j}^{(t+1)} \\
&+\lambda_1 ||X - X A^{(t+1)} B^{(t+1)}||_F^2 + \lambda_2 \sum_{i}^{n} ||s_i^{(t+1)}||_2^2 + \lambda_3 ||B^{(t+1)}||_{2,1} \\
\leq\ &\sum_{i,j}^{n} ||x^i A^{(t)} B^{(t)} - x^j A^{(t)} B^{(t)}||_2^2 s_{i,j}^{(t)} \\
&+\lambda_1 ||X - X A^{(t)} B^{(t)}||_F^2 + \lambda_2 \sum_{i}^{n} ||s_i^{(t+1)}||_2^2 + \lambda_3 ||B^{(t)}||_{2,1}
\end{aligned} \tag{22}$$

From (22), we know that the objective function value of (7) decreases after each iteration of Algorithm 1. Hence, Theorem 1 has been proven. □

4 Experimental results

We use 5 binary-class and 7 multi-class benchmark datasets to verify the performance of our method and the competing dimensionality reduction methods. The selected datasets include glass, wine, Parkinsons, Ionosphere, LungCancer, Sonar, Movements, Arrhythmia, LVST, ecoli and Yeast, all downloaded from the UCI Machine Learning Repository.[1] The Isolet

[1] http://archive.ics.uci.edu/ml/.

Table 1 Statistics of the datasets used

Datasets	Samples	Dimensions	Classes
Glass	214	9	6
Wine	178	13	3
Parkinsons	195	22	2
Ionosphere	351	34	2
LungCancer	32	56	2
Sonar	208	60	2
Movements	360	90	15
Arrhythmia	452	279	13
LVST	126	310	2
ecoli	336	343	8
Isolet	1560	617	26
Yeast	1484	1470	10

from the website of Feature Selection Datasets.[2] We summarize detailed information of the datasets in Table 1.

To prove the performance of LSS_FS, we compare it with six state-of-art feature selection methods. We list details of the competitor methods as follows:

TRACK: It selects discriminative features via unified trace ratio formulation and k-means clustering [25]. RSR: Regularized self-representation selects representative features via the $\ell_{2,1}$-norm to characterize the self-representation coefficient matrix and ensure the robustness to outilers [56]. CSFS: Convex Semi-supervised multi-label Feature Selection can conduct the feature selection via the $\ell_{2,1}$-norm regularization [5]. FSR_ALM : Exact Top-k Feature Selection via $\ell_{2,0}$-norm constraint and Augmented Lagrangian Multiplier (ALM) to avoid the heavy burden of tuning regularization parameters and make it more practically [4]. LDA: Liner Discriminant Analysis (LDA) aims at minimizing the within class distance while maximizing the between class distance when conducting feature selection. Hence, LDA is a global subspace learning method [7]. PCA: It maps high-dimensional data into low-dimensional space by preserving the covariance of the data matrix [50].

We set $\{\lambda_2, \lambda_3\} \in \{10^{-2}, \ldots, 10^2\}$, while the value of λ_1 is automatically adjusted according to [14], and the rank of the self-representation coefficient matrix $r \in \{1, \ldots, \min(n, d)\}$. Moreover, $\{c, g\} \in \{2^{-5}, \ldots, 2^5\}$ in SVM a 5-fold inner cross-validation is used to distinguish different types of samples. According to the cross-validation method, we can choose the best parameters for the experiments. In order to reduce impact of randomness, a 10-fold outer cross-validation has been used to get the average results. For the sake of fairness, we use the same strategy for all the competing methods.

We use three kinds of evaluation metrics, such as classification accuracy, standard deviation and coefficient of variation, to evaluate the classification performance of all methods. We define classification accuracy (ACC) as follows:

$$ACC = N_{correct}/N \tag{23}$$

where N is the number of samples and $N_{correct}$ is the number of correctly classified samples.

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

We utilize Standard Deviation (STD) to reflect the classification accuracy results and give the definition as follows:

$$STD = \sqrt{(1/n_{run}) \sum_{i=1}^{N} (ACC_i - u)} \tag{24}$$

where n_{run} denotes the runs of experiments, i.e., n_{run}=10 in our experiments, and u stands for the average value of ACC. A large ACC means good performance, and a small STD shows excellent stability.

We also use Coefficient of Variation (CV) as the evaluation index, which is defined as

$$CV = STD/ACC \tag{25}$$

where STD is the square root of an unbiased estimate of each sample, i.e., standard deviation. A small CV means better robustness.

4.1 Experiment results and analysis

In Table 2, we display the classification accuracy of all methods on twelve datasets in Table 1. Besides, we report the results of classification accuracy for all datasets in Fig. 1.

The proposed LSS_FS method outperforms all competing methods in all classification tasks. For instance, the ACC of our method increases on average by 10.97%, compared with TRACK which does not learn the relatively correct graph matrix of the high-dimensional data. ACC of out method raises on average by 9.45%, compared with the PCA method which is not able to take into account enough information for feature selection. On the other hand, ACC of our method climbs on average by 8.63%, compared with the LDA method which only considers the global structure of the data. Meahwhile, ACC of our method increases on average by 8.46%, compared with the FSR_ALM method which fails to take the relationship between features into account. Also, compared with CSFS which does not construct graph matrix to consider the local correlations among the features, ACC

Table 2 Classification accuracy (ACC±STD (%))

Datasets	TRACK	RSR	CSFS	FSR_ALM	LDA	PCA	LSS_FS
Glass	60.39±2.42	66.02±1.52	55.76±2.60	64.77±0.87	63.85±0.87	65.47±1.23	**71.69±0.85**
Wine	92.83±1.04	96.14±0.31	91.73±0.86	92.26±2.02	93.55±0.97	89.44±2.23	**97.45±0.30**
Parkinsons	86.13±0.75	88.49±**0.46**	86.89±0.50	86.34±0.61	86.18±1.25	79.96±0.64	**92.35**±1.33
Ionosphere	81.82±1.43	87.88±0.90	85.67±0.71	86.53±0.59	87.09±0.91	86.86±1.23	**92.88±0.51**
LungCancer	77.33±2.91	73.58±5.20	79.33±2.91	71.83±**0.73**	74.25±3.26	73.50±3.14	**79.67**±3.25
Sonar	76.17±1.32	74.44±1.42	78.10±**0.70**	78.25±0.85	78.24±0.86	76.54±1.74	**87.00**±1.01
Movements	79.47±1.29	80.42±0.95	77.44±1.56	77.81±1.76	79.86±**0.73**	80.08±1.17	**88.50**±0.77
Arrhythmia	66.95±0.88	67.27±1.36	67.07±0.96	67.47±0.96	67.01±1.47	63.34±1.22	**70.97±0.57**
LVST	61.60±2.70	83.42±**0.80**	60.53±4.01	60.31±5.22	84.14±1.68	63.09±3.83	**84.38**±1.45
ecoli	75.85±3.24	85.74±0.48	81.13±0.72	82.28±1.19	75.55±**0.04**	84.99±0.95	**86.00**±0.50
Isolet	81.46±0.44	80.53±2.32	96.08±0.26	96.39±**0.13**	83.60±2.02	95.94±0.21	**96.87**±0.22
Yeast	36.45±2.17	41.96±0.57	48.30±0.67	42.32±0.28	31.20±**0.01**	35.47±0.61	**60.33**±0.27
Average value	73.04±1.72	77.16±1.36	75.67±1.37	75.55±1.27	75.38±1.17	74.56±1.52	**84.01±0.92**

The bold number means the best result of each row

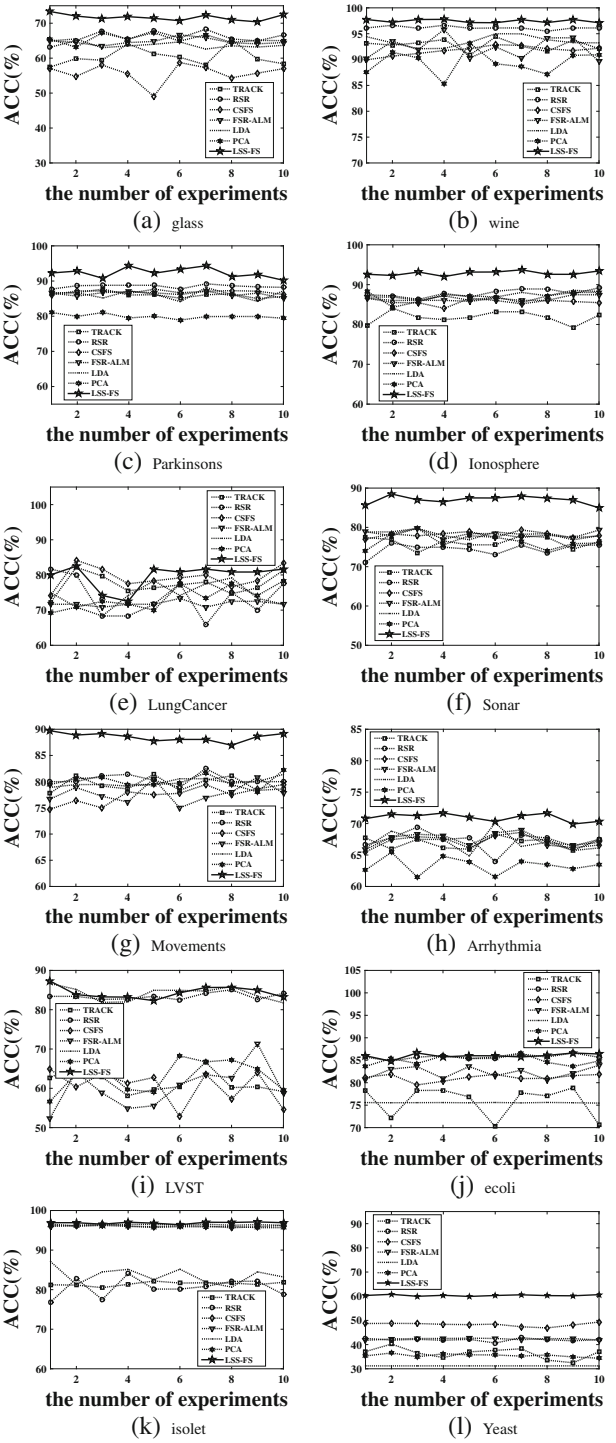


Fig. 1 Average classification accuracy of all methods on all the tested datasets

of our method increases on average by 8.34%. Compared with the RSR method that only considers the relationship between the features, the improvement on ACC of our method is 6.85% on average. The reason is that, LSS_FS method considers the following constraint while conducting feature selection: 1) two kinds of correlations, i.e., sample-level and feature-level, inherent in data; 2) iteratively adjusting the transformation matrix until it is optimal. Furthermore, the proposed method obtains the minimum standard deviation on average, compared with the other methods. Based on the above observations, our proposed method has the best robustness compared with all the other methods for classification tasks.

We show the results of coefficient of variation of all methods in Table 3. Specifically, our algorithm obtains the minimum value on the five datasets, to be specific, 1.19 on glass, 0.31 on wine, 0.55 on Ionosphere, 0.87 on Movements and 0.80 on Arrhythmia, respectively. Returning to the performance of CV in Table 3, though the proposed method does not achieve the minimum variation on each of the datasets, it reaches the minimum average variation on all the datasets. This verifies that our method achieves the best stability.

Figure 2 reveals the characteristic of the proposed objective values of Algorithm 1 on six datasets at each iteration, where we set the stop criteria of Algorithm 1 as $\frac{||obj(t+1)-obj(t)||_2^2}{obj(t)} \leq 10^{-5}$ where $obj(t)$ represents the value of objective function in (7) at the t-th iteration. From Fig. 2, we can know that: 1) the proposed Algorithm 1 to optimize the objective function in (7) monotonically decreases the objective function value until Algorithm 1 converges; 2) the proposed Algorithm 1 on all the datasets converges within twenty iterations, revealing that the proposed algorithm has a fast convergence rate.

From Fig. 3, we can know that the classification accuracy with a low-rank constraint for the most part is better than the classification accuracy with full-rank. For instance, the average classification accuracy of LSS_FS method with low rank constraint increases by 0.74%, 0.22%, 0.21%, 0.45%, 0.41%, 0.54%, 0.56%, 0.39%, 0.28%, 0.91%, 0.14% and 0.15%, respectively, compared with the results of full-rank constraint on dataset wine, Parkinsons, Ionosphere, LungCancer, Sonar, Movements, Arrhythmia, LVST, ecoli, Isolet and Yeast. Hence, it is obvious that analyzing high-dimensional data with a low rank constraint in

Table 3 Coefficient of variation (%)

Datasets	TRACK	RSR	CSFS	FSR_ALM	LDA	PCA	LSS_FS
Glass	4.01	2.30	4.66	1.34	1.36	1.88	**1.19**
Wine	1.12	0.32	0.94	2.19	1.04	2.49	**0.31**
Parkinsons	0.87	**0.52**	0.58	0.71	1.45	0.80	1.44
Ionosphere	1.75	1.02	0.83	0.68	1.04	1.42	**0.55**
LungCancer	3.76	7.07	3.67	**1.02**	4.39	4.27	4.08
Sonar	1.73	1.91	**0.90**	1.09	1.10	2.27	1.16
Movements	1.62	1.18	2.01	2.26	0.91	1.46	**0.87**
Arrhythmia	1.31	2.02	1.43	1.42	2.19	1.93	**0.80**
LVST	4.38	**0.96**	6.62	8.66	2.00	6.07	1.72
ecoli	4.27	0.56	0.89	1.45	**0.05**	1.12	0.58
Isolet	0.54	2.88	0.27	**0.13**	2.42	0.22	0.23
Yeast	5.95	1.36	1.39	0.66	**0.03**	1.72	0.63
Average value	2.61	1.84	2.02	1.80	1.50	2.14	**1.13**

The bold number means the best result of each row

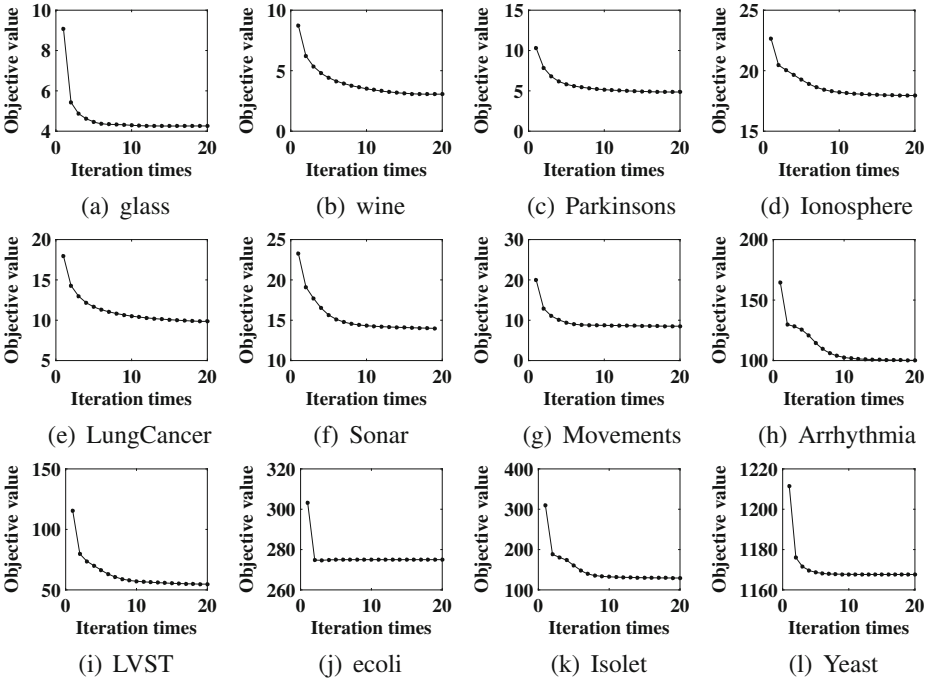


Fig. 2 Convergence rate of Algorithm 1 on all tested datasets

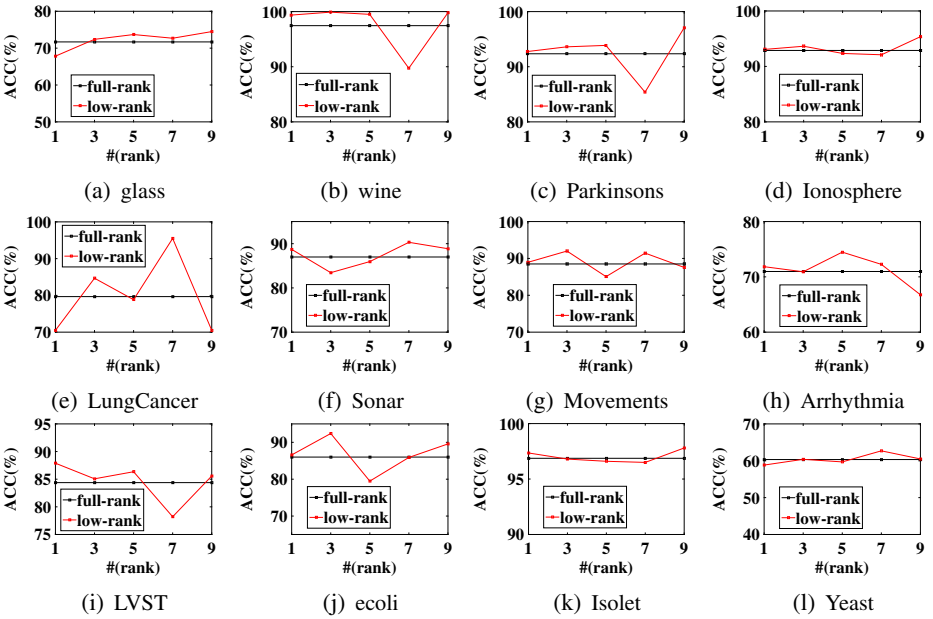
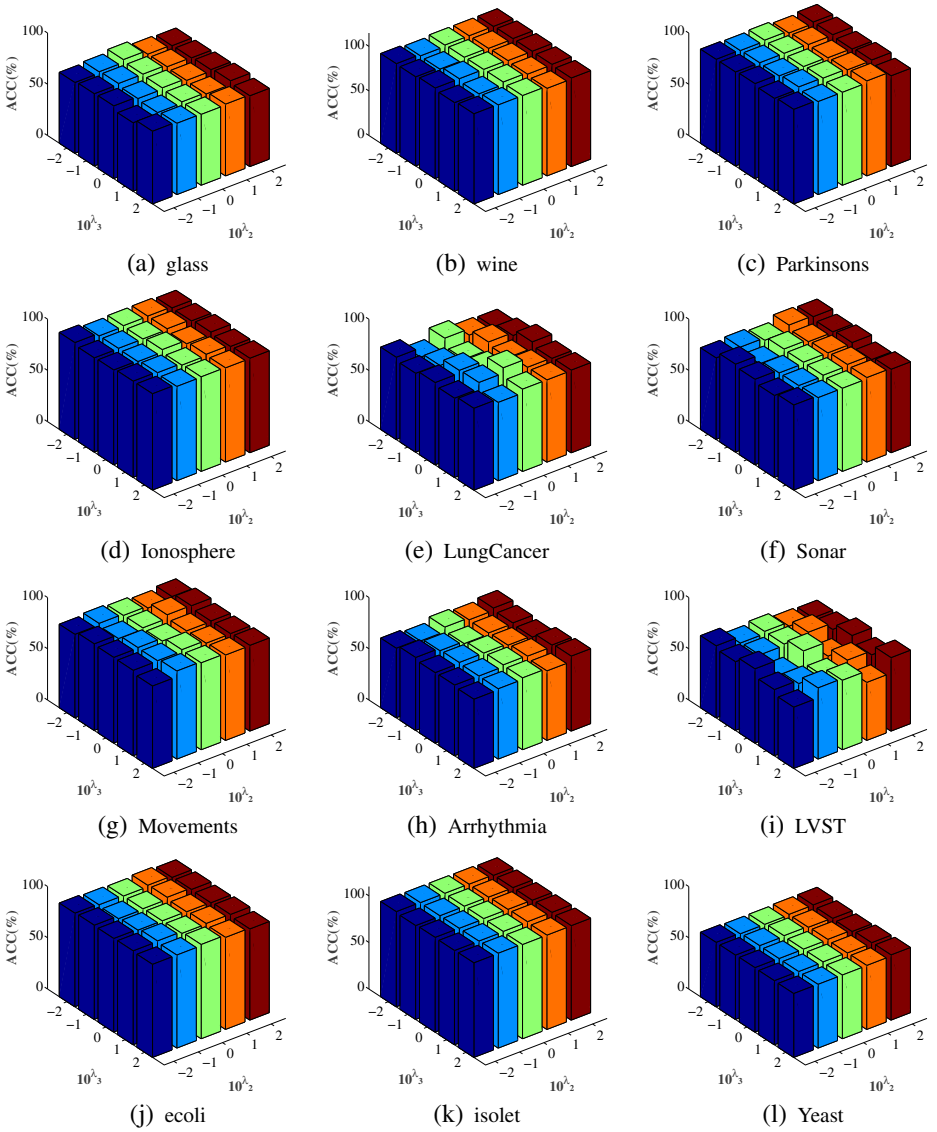


Fig. 3 Average classification accuracy of our method on all tested data sets for different number of ranks

feature selection is meaningful. Due to the fact that the low rank constraint conducting subspace learning helps search the low-dimensional space of high-dimensional data by considering the global feature correlation.

We vary parameter λ_2 and λ_3 within the range of $\{10^{-2}, \ldots, 10^2\}$ and list the results in Fig. 4. Parameter λ_2 is used to control the magnitude between the local representation term $\sum_{i,j}^{n} ||x^i AB - x^j AB||_2^2 s_{i,j}$ and the global representation term $||X - XAB||_F^2$, while λ_3 in (7) is used to adjust the sparsity of AB. In Fig. 4, our method achieves the best performance on dataset Sonar and ecoli while setting $\lambda_2 = 10$, and $\lambda_3 = 0.01$. Clearly, our method



(a) glass (b) wine (c) Parkinsons

(d) Ionosphere (e) LungCancer (f) Sonar

(g) Movements (h) Arrhythmia (i) LVST

(j) ecoli (k) isolet (l) Yeast

Fig. 4 ACC results of our proposed method for different λ_2 and λ_3

produces the best ACC, i.e., 84.38% with $\lambda_2 = 1$, and $\lambda_3 = 1$ for dataset LVST. This indicates that tuning of parameter benefits our method.

5 Conclusion

In this work, we proposed a novel feature selection method, called Unsupervised Feature Selection via Local Structure Learning and Sparse Learning (LSS_FS). LSS_FS method utilizes the similarity matrix to fine tune the self-representation coefficient matrix to output a high quality self-representation coefficient matrix. As a result, LSS_FS can have better discriminative power than traditional feature selection methods. Experimental results on real datasets show that LSS_FS method provides better feature selection performance than the competitor methods.

In the future work, we will extend the proposed method for semi-surprised feature selection tasks.

Acknowledgments This work was supported in part by the China Key Research Program (Grant No: 2016YFB1000905), the China 1000-Plan National Distinguished Professorship, the Nation Natural Science Foundation of China (Grants No: 61573270, 61672177 and 61363009), the Guangxi Natural Science Foundation (Grant No: 2015GXNSFCB139-011), the Guangxi High Institutions Program of Introducing 100 High-Level Overseas Talents, the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing, the Guangxi Bagui Teams for Innovation and Research, the Research Fund of Guangxi Key Lab of MIMS (16-A-01-01 and 16-A-01-02), the Guangxi Bagui Teams for Innovation and Research, and Innovation Project of Guangxi Graduate Education under grant XYCSZ2017064, XYCSZ2017067 and YCSW2017065.

References

1. Boyd S, Vandenberghe L (2013) Convex optimization
2. Cai D, He X, Han J (2007) Spectral regression: a unified approach for sparse subspace learning. In: ICDM, pp 73–82
3. Cai D, Zhang C, He X (2010) Unsupervised feature selection for multi-cluster data. In: KDD, pp 333–342
4. Cai X, Nie F, Huang H (2013) Exact top-k feature selection via l 2,0 -norm constraint. In: IJCAI, pp 1240–1246
5. Chang X, Nie F, Yi Y, Huang H (2014) A convex formulation for semi-supervised multi-label feature selection. In: AAAI, pp 1171–1177
6. Daubechies I, Devore R, Fornasier M, SiNan Gntk C (2008) Iteratively reweighted least squares minimization for sparse recovery. Commun Pure Appl Math 63(1):1–38
7. Fan Z, Yong X, Zhang D (2011) Local linear discriminant analysis framework using sample neighbors. IEEE Trans Neural Netw 22(7):1119–1132
8. Gao S, Tsang IW, Chia L-T (2013) Sparse representation with kernels. IEEE Trans Image Process 22(2):423–434
9. Gao L, Song J, Liu X, Shao J, Liu J, Shao J (2017) Learning in high-dimensional multimedia data: the state of the art. Multimed Syst 23(3):303–313
10. Gao L, Wang Y, Li D, Shao J, Song J (2017) Real-time social media retrieval with spatial, temporal and social constraints. Neurocomputing 253:77–88
11. Hu R, Zhu X, Cheng D, He W, Yan Y, Song J, Zhang S (2017) Graph self-representation method for unsupervised feature selection. Neurocomputing 220:130–137
12. Jayasena KPN, Li L, Xie Q (2017) Multi-modal multimedia big data analyzing architecture and resource allocation on cloud platform. Neurocomputing
13. Ling CX, Yang Q, Wang J, Zhang S (2004) Decision trees with minimal costs. In: ICML, pp 69
14. Nie F, Zhu W, Li X (2016) Unsupervised feature selection with structured graph optimization. In: AAAI, pp 1302–1308

15. Qian B, Wang X, Cao N, Gang Jiang Y, Davidson I (2014) Learning multiple relative attributes with humans in the loop. IEEE Trans Image Process 23(12):5573–5585
16. Qian B, Wang X, Cao N, Li H, Gang Jiang Y (2015) A relative similarity based method for interactive patient risk prediction. Data Mining Knowl Discov 29(4):1070–1093
17. Qin Y, Zhang S, Zhu X, Zhang J, Zhang C (2007) Semi-parametric optimization for missing data imputation. Appl Intell 27(1):79–88
18. Song J, Yi Y, Zi H, Shen HT, Luo J (2013) Effective multiple feature hashing for large-scale near-duplicate video retrieval. IEEE Trans Multimed 15(8):1997–2008
19. Song J, Gao L, Nie F, Shen HT, Yan Y, Sebe N (2016) Optimized graph learning using partial tags and multiple features for image and video annotation. IEEE Trans Image Process 25(11):4999–5011
20. Song J, Gao L, Zou F, Yan Y, Sebe N (2016) Deep and fast: deep learning hashing with semi-supervised graph construction. Image Vis Comput 55:101–108
21. Song J, Shen HT, Wang J, Zi H, Sebe N, Wang J (2016) A distance-computation-free search scheme for binary code databases. IEEE Trans Multimed 18(3):484–495
22. Sun J, Zhou A (2014) Unsupervised robust bayesian feature selection, pp 558–564
23. Wang T, Qin Z, Zhang S, Zhang C (2012) Cost-sensitive classification with inadequate labeled data. Inf Syst 37(5):508–516
24. Wang X, Qian B, Davidson I (2012) On constrained spectral clustering and its applications. Data Mining Knowl Discov 28(1):1–30
25. Wang D, Nie F, Huang H (2014) Unsupervised feature selection via unified trace ratio formulation and k-means clustering (track). In: Ecml/pkdd, pp 306–321
26. Wen Z, Yin W (2013) A feasible method for optimization with orthogonality constraints. Math Program 142(1):397–434
27. Xia Y, He K, Kohli P, Sun J (2015) Sparse projections for high-dimensional binary codes. In: Computer vision and pattern recognition, pp 3332–3339
28. Xie Q, Pang C, Zhou X, Zhang X, Ke D (2014) Maximum error-bounded piecewise linear representation for online stream approximation. Vldb J 23(6):915–937
29. Xie QS, Wang JZ, Zhang X (2016) Modeling and predicting ad progression by regression analysis of sequential clinical data. Neurocomputing 195(C):50–55
30. Xie Q, Zhang X, Li Z, Zhou X (2016) Optimizing cost of continuous overlapping queries over data streams by filter adaption. IEEE Trans Knowl Data Eng 28(5):1258–1271
31. Xindong W, Zhang S (2003) Synthesizing high-frequency rules from different data sources. IEEE Trans Knowl Data Eng 15(2):353–367
32. Xindong W, Zhang C, Zhang S (2004) Efficient mining of both positive and negative association rules. Acm Trans Inf Syst 22(3):381–405
33. Xindong W, Zhang C, Zhang S (2005) Database classification for multi-database mining. Inf Syst 30(1):71–88
34. Yan X, Zhang C, Zhang S (2009) Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Syst Appl 36(2):3066–3076
35. Zhang S (2011) Shell-neighbor method and its application in missing data imputation. Appl Intell 35(1):123–133
36. Zhang S (2012) Nearest neighbor selection for iteratively knn imputation. J Syst Softw 85(11):2541–2552
37. Zhang C, Zhang S (2002) Association rule mining: models and algorithms 2307
38. Zhang S, Zhang C (2002) Anytime mining for multiuser applications. IEEE Trans Syst Man Cybern-Part Syst Humans 32(4):515–521
39. Zhang S, Zhang C, Yang Q (1999) Data preparation for data mining. Academic Press
40. Zhang S, Zhang C, Yan X (2003) Post-mining: maintenance of association rules by weighting. Inf Syst 28(7):691–707
41. Zhang S, Wu X, Zhang C (2003) Multi-database mining 2:5–13
42. Zhang S, Qin Z, Ling CX, Sheng S (2005) Missing is useful?: missing values in cost-sensitive decision trees. IEEE Trans Knowl Data Eng 17(12):1689–1693
43. Zhang S, Jin Z, Zhu X (2011) Missing data imputation by utilizing information within incomplete instances. J Syst Softw 84(3):452–459
44. Zhang S, Li X, Zong M, Zhu X, Cheng D (2017) Learning k for knn classification. ACM Trans Intell Syst Technol 8(3):43
45. Zhang S, Li X, Zong M, Zhu X, Wang R (2017) Efficient knn classification with different numbers of nearest neighbors. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2017.2673241
46. Zhao Y, Zhang S (2005) Generalized dimension-reduction framework for recent-biased time series analysis. IEEE Trans Knowl& Data Eng 18(2):231–244

47. Zhong F, Zhang J (2013) Linear discriminant analysis based on l1-norm maximization. IEEE Trans Image Process 22(8):3018–3027
48. Zhu Y, Lucey S (2015) Convolutional sparse coding for trajectory reconstruction. IEEE Trans Pattern Anal Mach Intell 37(3):529–540
49. Zhu X, Zhang S, Jin Z, Zhang Z (2011) Missing value estimation for mixed-attribute data sets. IEEE Trans Knowl Data Eng 23(1):110–121
50. Zhu X, Zi H, Shen HT, Cheng J, Changsheng X (2012) Dimensionality reduction by mixed kernel canonical correlation analysis. Pattern Recogn 45(8):3003–3016
51. Zhu X, Zi H, Shen HT, Zhao X (2013) Linear cross-modal hashing for efficient multimedia search. In: ACM International conference on multimedia, pp 143–152
52. Zhu X, Zi H, Cheng H, Cui J, Shen HT (2013) Sparse hashing for fast multimedia search. ACM Trans Inf Syst 31(2):9
53. Zhu X, Zi H, Yang Y, Shen HT, Changsheng X, Luo J (2013) Self-taught dimensionality reduction on the high-dimensional small-sized data. Pattern Recogn 46(1):215–229
54. Zhu X, Zhang L, Huang Z (2014) A sparse embedding and least variance encoding approach to hashing. IEEE Trans Image Process 23(9):3737
55. Zhu X, Suk HI, Shen D (2014) A novel matrix-similarity based loss function for joint regression and classification in ad diagnosis. Neuroimage 100:91–105
56. Zhu P, Zuo W, Zhang L, Qinghua H, Shiu SC (2015) Unsupervised feature selection by regularized self-representation. Pattern Recogn 48(2):438–446
57. Zhu X, Xie Q, Zhu Y, Liu X, Zhang S (2015) Multi-view multi-sparsity kernel reconstruction for multi-class image classification. Neurocomputing 169:43–49
58. Zhu X, Suk HI, Lee SW, Shen D (2015) Canonical feature selection for joint regression and multi-class identification in alzheimer's disease diagnosis. Brain Imag Behav 10(3):1–11
59. Zhu X, Li X, Zhang S (2016) Block-row sparse multiview multilabel learning for image classification. IEEE Trans Cybern 46(2):450
60. Zhu X, Suk H-I, Lee S-W, Shen D (2016) Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification. IEEE Trans Biomed En. 63(3):607–618
61. Zhu Y, Zhu X, Kim M, Shen D, Guorong W (2016) Early diagnosis of alzheimers disease by joint feature selection and classification on temporally structured support vector machine. In: MICCAI, pp 264–272
62. Zhu X, He W, Li Y, Yang Y, Zhang S, Rongyao H, Zhu Y (2017) One-step spectral clustering via dynamically learning affinity matrix and subspace. In: AAAI, pp 2963–2969
63. Zhu X, Li X, Zhang S, Chunhua J, Xindong W (2017) Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE Trans Neural Netw Learn Syst 28(6):1263–1275
64. Zhu X, Li X, Zhang S, Xu Z, Yu L, Wang C (2017) Graph PCA hashing for similarity search. IEEE Multimed Multimed 19(9):2033–2044
65. Zhu X, Suk HII, Huang H, Shen D (2017) Low-rank graph-regularized structured sparse regression for identifying genetic biomarkers. IEEE Transact Big Data. https://doi.org/10.1109/TBDATA.2017.2735991
66. Zhu X, Suk H-I, Wang L, Lee S-W, Shen D (2017) A novel relational regularization feature selection method for joint regression and classification in AD diagnosis. Med Image Anal 38:205–214

Cong Lei is with the Guangxi Key Lab of Multisource Information Mining & Security, Guangxi Normal University, P. R. China.

Xiaofeng Zhu is with the Guangxi Normal University, P. R. China. His research interests include data mining and machine learning.