



Joint self-representation and subspace learning for unsupervised feature selection

Ruili Wang¹ · Ming Zong¹

Received: 4 September 2017 / Revised: 8 October 2017 / Accepted: 16 October 2017 /

Published online: 6 November 2017

© Springer Science+Business Media, LLC 2017

Abstract This paper proposes a novel unsupervised feature selection method by jointing self-representation and subspace learning. In this method, we adopt the idea of self-representation and use all the features to represent each feature. A Frobenius norm regularization is used for feature selection since it can overcome the over-fitting problem. The Locality Preserving Projection (LPP) is used as a regularization term as it can maintain the local adjacent relations between data when performing feature space transformation. Further, a low-rank constraint is also introduced to find the effective low-dimensional structures of the data, which can reduce the redundancy. Experimental results on real-world datasets verify that the proposed method can select the most discriminative features and outperform the state-of-the-art unsupervised feature selection methods in terms of classification accuracy, standard deviation, and coefficient of variation.

Keywords Unsupervised feature selection · Self-representation · Subspace learning

1 Introduction

Feature selection is a research hotspot in the fields of pattern analysis, machine learning, and data mining. The earliest feature selection studies mainly focused on statistical and signal processing problems. Since large-scale machine learning emerged in 1990s, existing feature

This article belongs to the Topical Collection: *Special Issue on Deep Mining Big Social Data*
Guest Editors: Xiaofeng Zhu, Gerard Sanroma, Jilian Zhang, and Brent C. Munsell

✉ Ruili Wang
ruili.wang@massey.ac.nz

Ming Zong
M.Zong@massey.ac.nz

¹ Institute of Natural and Mathematical Sciences (INMS), Massey University, Albany, Auckland, New Zealand

selection algorithms had diffract to meet the challenge [26]. When the number of features reaches a certain size, the accuracy of a classifier is declining, which is called the curse of dimensionality [1, 19, 36]. Therefore, there is an urgent need to develop better feature selection algorithms to increase the accuracy and efficiency for the large-scale data.

Feature selection is a method of selecting some of features that have more discriminative ability from a set of features to reduce the dimension of a feature space. It is an important component of a classification system [21, 27]. For a classification system, a good learning sample is the key in training a classifier. The quality of the data, for example, whether the sample contains irrelevant or redundant features can directly affect the performance of the classifier [29, 30]. Therefore, it is important to develop an effective feature selection method.

In general, based on the combination of subset evaluation criteria in feature selection and follow-up learning algorithms, the feature selection approaches can be categorized into three groups: the filter approach, wrapper approach and embedded approach. The filter approach [10] is independent of the follow-up learning algorithm and it uses the statistical performance of all training samples to evaluate the features [17]. The time cost of the filter approach is low, but the evaluation may have a deviation with the follow-up learning algorithm. While the wrapper approach [10, 18] uses the follow-up learning algorithm to evaluate the accuracy of the training features, the deviation is thus small, but the computation cost is large and thus not suitable for a large-scale data set [31]. In the embedded feature selection approach [7], a feature selection method itself is embedded as a component into a learning algorithm. The most typical embedded method is a decision tree [30]. However, the key of feature selection methods depend on the efficient selection of a useful subset of features. The features in this selected feature subset are kept while remaining features are abandoned. However, the abandoned features may also be related to other features, while the abandonment may lose some useful relevant information. Further, it is helpful for feature selection to effectively utilize the relation between data.

In order to utilize the relation between data, firstly, self-representation has been widely used for feature selection [33], according to self-similarity, i.e., a feature can be represented by all other features. Then, subspace learning is also introduced for keeping the relevance between data [9, 12, 35]. Subspace learning is designed to maintain specific statistical properties such as Principal Component Analysis (PCA) [37], Linear Discriminant Analysis (LDA) and so on when performing feature space transformation. These subspace learning methods can effectively mitigate the so-called curse of dimensionality and preserve the inherent relevance of the data. Thus, the above two points motivate this paper to consider taking advantages of the merits of both self-representation and subspace learning as a whole.

Therefore, this paper proposed a new Unsupervised Feature selection method by jointing Self-representation and Subspace learning, which is called UFSS for short. Firstly, because of the correlation between features, we consider using all features to represent each feature, that is, each feature is reconstructed by all features. Correspondingly, we use the least squares method as a loss function to evaluate the reconstruction error. Then, in order to overcome the over-fitting problem and select the most discriminative features, we use the Frobenius norm to constrain the reconstruction coefficient matrix to overcome the over-fitting problem and select the most discriminative features. On the other hand, we introduce Locality Preserving Projection (LPP) as a regularization term to maintain the local adjacent relation of the data when performing feature space transformation during the reconstruction process [11]. At the same time, we consider further applying a low-rank constraint to find the effective low-dimensional structures of the data, which can reduce the redundancy [34]. Finally, we proposed an effective optimization method to solve the objective function fast.

In summarization, the core of feature selection methods is to select the most effective features from the original features to reduce the feature dimension, which is a key data preprocessing step in pattern recognition. Based on the selected optimal feature subset, we use a classic classifier, i.e., Support Vector Machine (SVM), to classify the test samples.

The rest of this paper is organized as follows: We briefly review the previous feature selection methods and subspace learning methods in Section 2. After that, in Section 3, we give the details of the proposed new unsupervised feature selection method UFSS. Then, we present the experimental results in Section 4. Finally, we summarize our work and future work in Section 5.

2 Related work

In this section, we briefly review three important items: unsupervised feature selection, subspace learning and self-representation, because our proposed algorithm is based on them and they play different roles during the reconstruction.

2.1 Unsupervised feature selection

Supervised feature selection methods use the class labels as a guide to achieve feature selection. However, a lot of data may be unlabeled in a practical application. Therefore, unsupervised feature selection methods are useful and more difficult because they do not have class labels to use. Tabakhi et al. [19] proposed an ant colony algorithm, which can provide a well approximate solution based on previous iterations and the time cost is acceptable. Liu et al. [13] drew lessons from the Laplacian Score method. They considered replacing the k-means clustering method with a distance-based entropy measure in the Laplacian Score (LS) for automatically selecting the optimal subset of features. Qian and Zhai [16] took advantages of local learning and nonnegative matrix factorization. The proposed method can select the most discriminative subset of features by combining robust clustering and robust feature selection at the same time. Based on manifold learning and sparse learning model, Cai et al. [3] proposed Multi-Cluster Feature Selection (MCFS)). They considered using spectral analysis methods based on the preserved multi-cluster structure of the data to measure the relevance between different features.

2.2 Subspace learning

Subspace learning has been applied in different kinds of models for reducing dimension. Zou et al. [39] proposed a new improved principal component analysis based on sparse coding, which is called sparse principal component analysis (SPCA). It uses Lasso penalty to produce sparse principal component. Yan et al. [24] considered using graph embedding technology to represent the geometric structure or properties of the sample space. Based on it, they applied it to characterize intraclass compactness and interclass separability simultaneously and can better solve the problem that the number of available projection directions is low in LDA. Recently, Nikitidis et al. [15] proposed a method called maximum margin projection pursuit. It can take advantage of maximum margin to discriminate samples when performing feature space transformation. Cai et al. [2] proposed to use both graph embedding and regression for sparse projections learning, it can solve different graph-based subspace learning methods by the proposed unified framework.

2.3 Self-representation

Self-representation stems from the natural self-similarity phenomenon [33], which means that a part of an object is similar to other parts of the object, such as coastlines, stock market movements and so on. Just like sparsity results in sparse representation, self-similarity leads to self-representation. Of course, self-representation has been widely used for high dimensional data. Zhu et al. [33] proposed to use self-representation and $\ell_{2,1}$ -norm to constrain the coefficient matrix for removing outliers and can select the most representative features to reconstruct other features. Zhang et al. [28] proposed a new improved kNN method based self-representation, which also uses training samples to reconstruct themselves and imposes ℓ_1 -norm to make representation coefficient matrix to produce sparsity.

3 Approach

In this section, firstly, we give some notations used in this paper in Section 3.1 and give some basic knowledge as preliminary in Section 3.2. The details of the proposed UFSS method is described in Section 3.3. We presented an optimization method to solve the objective function in Section 3.4. Finally, we analyze the convergence of the objective function in Section 3.5.

3.1 Notations

In this paper, we denote scalars as normal italic letters, vectors as bold lowercase letters and matrices as bold uppercase letters, respectively [29, 30]. Given a matrix $\mathbf{X} = [x_{ij}]$, we denote the i th row of \mathbf{X} by \mathbf{x}^i , and the j th column of \mathbf{X} by \mathbf{x}_j . The Frobenius norm and ℓ_1 -norm of \mathbf{X} are defined as $\|\mathbf{X}\|_F = \sqrt{\sum_i \|\mathbf{x}^i\|_2^2} = \sqrt{\sum_j \|\mathbf{x}_j\|_2^2}$ (matrix norms here are entry-wise norm), $\|\mathbf{X}\|_1 = \sqrt{\sum_i \sum_j |x_{ij}|}$ and $\|\mathbf{X}\|_{21} = \sum_i \sqrt{\sum_j x_{ij}^2}$. The trace operator, the transpose operator and the inverse of \mathbf{X} is denoted as $tr(\mathbf{X})$, \mathbf{X}^T and \mathbf{X}^{-1} , respectively.

3.2 Preliminary

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denotes a sample matrix, where n and d denote the numbers of samples and features, respectively. We also use $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ to denote the n samples where is a column vector, thus $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_n]$. On the other hand, we use $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_d$ to denote the d features and $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_d$ are the corresponding feature vectors, where $\mathbf{f}_i \in \mathbb{R}^n$ and $\mathbf{X} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_d]$.

The key of unsupervised feature selection methods is to select an optimal feature subset for all the samples. Drawn lessons from the regression problem [28, 33], we can regard the feature selection problem as a regression problem [33]:

$$\min_{\mathbf{W}} l(\mathbf{XW} - \mathbf{Y}) + \lambda R(\mathbf{W}) \tag{1}$$

where \mathbf{W} denotes a coefficient matrix, which is used to measure the weight of a feature. \mathbf{Y} usually is a response matrix and $l(\mathbf{XW} - \mathbf{Y})$ denotes a loss function. $R(\mathbf{W})$ usually is a regularization imposed on \mathbf{W} and λ is a positive constant.

3.3 UFSS method

Many developed feature selection methods are derived from the model in (1), although considering correlation between features, it is still hard to select the proper response matrix \mathbf{Y} . Drawing lessons from the merits of both self-representation and subspace learning, in this section, we proposed a new combined method for unsupervised feature selection. The self-representation denotes that the proposed method uses the sample matrix \mathbf{X} instead of the response matrix \mathbf{Y} , i.e., $\mathbf{Y} = \mathbf{X}$, which means each feature can be represented by all the features. Therefore, we can represent each feature \mathbf{f}_i as follow:

$$\mathbf{f}_i = \sum_{j=1}^d \mathbf{f}_j w_{ji} + \mathbf{e}_i \tag{2}$$

Applied to all the features, then (2) can also be represented in a matrix form as follow:

$$\mathbf{X} = \mathbf{XW} + \mathbf{eb} \tag{3}$$

where $\mathbf{W} = [w_{ji}] \in \mathbb{R}^{d \times d}$ is the self-representation coefficients matrix, $\mathbf{e} \in \mathbb{R}^{n \times 1}$ denotes a column vector with all elements are 1 and $\mathbf{b} \in \mathbb{R}^{1 \times d}$ is a bias term. Obviously, in order to use \mathbf{XW} to represent \mathbf{X} sufficiently, we should make error term \mathbf{eb} as small as possible. Frobenius norm can be adopted to measure the residual, i.e., $\min_{\mathbf{W}, \mathbf{b}} \|\mathbf{X} - \mathbf{XW} - \mathbf{eb}\|_F$. The matrix

\mathbf{W} reflects the importance of different features. To avoid over-fitting and to select the most discriminative features, absorbing the core idea from ridge regression [29, 30], a shrink regularization factor is introduced, i.e., $\|\mathbf{W}\|_F$. In addition, as the basic assumption of manifold learning, a classic method of subspace learning, we know that real data may be presented in a high-dimensional structure but actually it may exist in a very low-dimensional manifold, i.e., the data can be represented by low-dimensional structure to some extent if we can map it back into the low dimensional space and reveal its essence. Taking advantage of manifold learning, locality preserving projection (LPP) as a regularization term is introduced to maintain the local adjacent relations of the data after performing feature space transformation during the self-representation process [32, 38]. Then we can have the objective function as follow:

$$\min_{\mathbf{W}, \mathbf{b}} \|\mathbf{X} - \mathbf{XW} - \mathbf{eb}\|_F + \lambda_1 \|\mathbf{W}\|_F + \lambda_2 tr(\mathbf{W}^T \mathbf{X}^T \mathbf{LXW}) \tag{4}$$

where λ_1 and λ_2 are control parameters. The penalty term $\|\mathbf{W}\|_F$ is used for penalizing all coefficients in \mathbf{W} together; $\mathbf{L} = \mathbf{D} - \mathbf{S} \in \mathbb{R}^{d \times d}$ is called graph Laplacian, where $\mathbf{S} \in \mathbb{R}^{d \times d}$ is a similarity matrix and $\mathbf{D} \in \mathbb{R}^{d \times d}$ is a diagonal matrix. On the other hand, in order to further remove the large amount of redundancy in the data, a low rank constraint is introduced to find the effective low-dimensional structures of the data, which can guarantee to reduce the redundancy. Thus, the low rank constraint can be applied to the rank of \mathbf{W} , i.e.,

$$rank(\mathbf{W}) = r, \quad r \leq \min(n, d) \tag{5}$$

Further, (5) can be re-expressed as product of two r – rank matrices as follow:

$$\mathbf{W} = \mathbf{AB} \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{d \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times d}$. Therefore, the final objective function becomes:

$$\begin{aligned} & \min_{\mathbf{A}, \mathbf{B}, \mathbf{b}} \|\mathbf{X} - \mathbf{XAB} - \mathbf{eb}\|_F + \lambda_1 \|\mathbf{W}\|_F + \lambda_2 \text{tr}(\mathbf{B}^T \mathbf{A}^T \mathbf{X}^T \mathbf{LXAB}) \\ & \text{s.t., } \text{rank}(\mathbf{AB}) \leq \min(n, d) \end{aligned} \tag{7}$$

In brief, we uses the proposed UFSS method to select the optimal feature subset from the original feature space, which is a key data preprocessing step for reducing dimension. Further, based on the selected optimal feature subset, this paper conducts classification with a classic classifier, i.e., Support Vector Machine (SVM). The pseudo of the UFSS algorithm is described in Algorithm 1 as below.

Algorithm 1 The Pseudo of UFSS algorithm

Input: training samples \mathbf{X} , test samples \mathbf{Y}

Output: Class label

* Feature Selection Stage *

1. Solving (7) to obtain the optimal solution \mathbf{W} ;
2. According to \mathbf{W} , the most efficient features are selected;

* Classification Stage *

1. According to the selected feature subset, using SVM to conduct classification;
 2. Output the predicted label;
-

3.4 Optimization

Since with respect to the three variables, i.e., \mathbf{A} , \mathbf{B} and \mathbf{b} , the objective function (7) is not jointly convex. We propose an iterative algorithm to optimize with \mathbf{A} , \mathbf{B} and \mathbf{b} . Concretely, we iteratively excute the following two steps until the pre-set conditions are met: (i) Update \mathbf{b} with the fixed \mathbf{A} and \mathbf{B} ; (ii) Update \mathbf{A} and \mathbf{B} with the fixed \mathbf{b} .

- (i) Fix \mathbf{A} and \mathbf{B} , then update \mathbf{b} .

We set the derivative of the objective function (7) with respect to \mathbf{b} equal to 0:

$$2\mathbf{e}^T \mathbf{eb} + 2\mathbf{e}^T \mathbf{XAB} - 2\mathbf{e}^T \mathbf{X} = 0 \tag{8}$$

We have this form by transformation:

$$\mathbf{b} = (1/n)(\mathbf{e}^T \mathbf{X} - \mathbf{e}^T \mathbf{XAB}) \tag{9}$$

- (ii) Fix \mathbf{b} , then update \mathbf{A} and \mathbf{B} .

We substitute (9) into the objective function (7), then we have:

$$\begin{aligned} & \min_{\mathbf{A}, \mathbf{B}, \mathbf{b}} \|\mathbf{X} - \mathbf{XAB} - \mathbf{e}((1/n)(\mathbf{e}^T \mathbf{X} - \mathbf{e}^T \mathbf{XAB}))\|_F + \lambda_1 \|\mathbf{W}\|_F \\ & + \lambda_2 \text{tr}(\mathbf{B}^T \mathbf{A}^T \mathbf{X}^T \mathbf{LXAB}) \end{aligned} \tag{10}$$

Let $\mathbf{H} = \mathbf{I}_n - (1/n)\mathbf{ee}^T \in \mathbf{R}^{n \times n}$, where $\mathbf{I}_n \in \mathbf{R}^{n \times n}$ is an identity matrix, (10) can be rewritten as

$$\min_{\mathbf{A}, \mathbf{B}} \|\mathbf{HX} - \mathbf{HXAB}\|_F + \lambda_1 \|\mathbf{AB}\|_F + \lambda_2 \text{tr}(\mathbf{B}^T \mathbf{A}^T \mathbf{X}^T \mathbf{LXAB}) \tag{11}$$

which is equivalent to

$$\begin{aligned} & \min_{\mathbf{A}, \mathbf{B}} \text{tr}((\mathbf{HX} - \mathbf{HXAB})^T \mathbf{P}(\mathbf{HX} - \mathbf{HXAB})) + \lambda_1 \text{tr}(\mathbf{B}^T \mathbf{A}^T \mathbf{QAB}) \\ & + \lambda_2 \text{tr}(\mathbf{B}^T \mathbf{A}^T \mathbf{X}^T \mathbf{LXAB}) \end{aligned} \tag{12}$$

where $\mathbf{P} \in \mathbf{R}^{n \times n}$ and $\mathbf{Q} \in \mathbf{R}^{d \times d}$, respectively, are diagonal matrices with $\mathbf{P}_{ii} = \frac{1}{2} \|(\mathbf{H}\mathbf{X} - \mathbf{H}\mathbf{X}\mathbf{A}\mathbf{B})^i\|_F, i = 1, \dots, n$, and $\mathbf{Q}_{jj} = \frac{1}{2} \|(\mathbf{A}\mathbf{B})^j\|_F, j = 1, \dots, d$.

By setting (12) w.r.t \mathbf{B} to zero, we obtain:

$$\mathbf{B} = (\mathbf{A}^T (\mathbf{X}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \mathbf{X} + \lambda_1 \mathbf{Q} + \lambda_2 \mathbf{X}^T \mathbf{L} \mathbf{X}) \mathbf{A})^{-1} \mathbf{A}^T \mathbf{X}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \mathbf{X} \tag{13}$$

Substituting (13) into (12), we have:

$$\begin{aligned} & \max_{\mathbf{A}} \text{tr}(\mathbf{A}^T (\mathbf{X}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \mathbf{X} + \lambda_1 \mathbf{Q} + \lambda_2 \mathbf{X}^T \mathbf{L} \mathbf{X}) \mathbf{A})^{-1} \\ & \mathbf{A}^T \mathbf{X}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \mathbf{X} \mathbf{X}^T \mathbf{H}^T \mathbf{P}^T \mathbf{H} \mathbf{X} \mathbf{A} \end{aligned} \tag{14}$$

Note that

$$\begin{aligned} \mathbf{S}_t &= \mathbf{X}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \mathbf{X} + \lambda_1 \mathbf{Q} + \lambda_2 \mathbf{X}^T \mathbf{L} \mathbf{X} \\ \mathbf{S}_b &= \mathbf{X}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \mathbf{X} \mathbf{X}^T \mathbf{H}^T \mathbf{P}^T \mathbf{H} \mathbf{X} \end{aligned} \tag{15}$$

where \mathbf{S}_t and \mathbf{S}_b denote the total-class scatter matrix and the between-class scatter matrix, respectively. Then the solution of (14) can be represented as below:

$$\mathbf{A}^* = \arg \max_{\mathbf{A}} \text{tr}[(\mathbf{A}^T \mathbf{S}_t \mathbf{A})^{-1} \mathbf{A}^T \mathbf{S}_b \mathbf{A}] \tag{16}$$

The global optimal solution of (16) is the top s eigenvectors of $\mathbf{S}_t^{-1} \mathbf{S}_b$. The above analysis leads to Algorithm 2 below.

Algorithm 2 The Pseudo code of solving the objective function (7)

Input: $\mathbf{X} \in \mathbf{R}^{n \times d}, \lambda_1, \lambda_2, p, r;$

Output: $\mathbf{A} \in \mathbf{R}^{d \times r}, \mathbf{B} \in \mathbf{R}^{r \times d};$

1. Obtain \mathbf{L} by constructing the $k - nn$ graph using 0-1 weights;

2. Initialize $k = 0, \mathbf{P}(0)$ and $\mathbf{Q}(0)$ as random diagonal matrix;

3. **repeat:**

4. Update $\mathbf{A}(k + 1)$ via (16);

5. Update $\mathbf{B}(k + 1)$ via (13);

6. Update $\mathbf{b}(k + 1)$ via (9);

7. Compute the diagonal matrix $\mathbf{P}(k + 1)$ as $\mathbf{P}_{ii} = \frac{1}{2} \|(\mathbf{H}\mathbf{X} - \mathbf{H}\mathbf{X}\mathbf{A}\mathbf{B})^i\|_F, i = 1, \dots, n;$

8. Compute the diagonal matrix $\mathbf{Q}(k + 1)$ as $\mathbf{Q}_{jj} = \frac{1}{2} \|(\mathbf{A}\mathbf{B})^j\|_F, j = 1, \dots, d;$

9. $k = k + 1;$

10. **Until** The difference between the two iterations of the objective function (7) is less than $10^{-5}.$

3.5 Proving of the convergence

In each iteration, it can be proved that the objective function (7) is monotonically decreasing [4]. Note that the objective function (11) and the objective function (12) are equivalent. Thus, we have

$$\begin{aligned} & \Rightarrow \text{tr}[(\mathbf{H}\mathbf{X} - \mathbf{H}\mathbf{X}\mathbf{A}^{(t+1)} \mathbf{B}^{(t+1)})^T \mathbf{P}^{(t)} (\mathbf{H}\mathbf{X} - \mathbf{H}\mathbf{X}\mathbf{A}^{(t+1)} \mathbf{B}^{(t+1)})] \\ & \quad + \lambda_1 \text{tr}(\mathbf{B}^{(t+1)T} \mathbf{A}^{(t+1)T} \mathbf{Q}^{(t)} \mathbf{A}^{(t+1)} \mathbf{B}^{(t+1)}) \\ & \quad + \lambda_2 \text{tr}(\mathbf{B}^{(t+1)T} \mathbf{A}^{(t+1)T} \mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{A}^{(t+1)} \mathbf{B}^{(t+1)}) \\ & \leq \text{tr}[(\mathbf{H}\mathbf{X} - \mathbf{H}\mathbf{X}\mathbf{A}^{(t)} \mathbf{B}^{(t)})^T \mathbf{P}^{(t)} (\mathbf{H}\mathbf{X} - \mathbf{H}\mathbf{X}\mathbf{A}^{(t)} \mathbf{B}^{(t)})] \\ & \quad + \lambda_1 \text{tr}(\mathbf{B}^{(t)T} \mathbf{A}^{(t)T} \mathbf{Q}^{(t)} \mathbf{A}^{(t)} \mathbf{B}^{(t)}) + \lambda_2 \text{tr}(\mathbf{B}^{(t)T} \mathbf{A}^{(t)T} \mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{A}^{(t)} \mathbf{B}^{(t)}) \end{aligned} \tag{17}$$

Let $\mathbf{Z} = \mathbf{H}\mathbf{X} - \mathbf{H}\mathbf{X}\mathbf{A}\mathbf{B}$, then we have [4]:

$$\begin{aligned} &\Rightarrow \sum_{i=1}^n \frac{\|z^{i(t+1)}\|_2^2}{2\|z^{i(t)}\|_2} + \lambda_1 \sum_{j=1}^d \frac{\|w^{j(t+1)}\|_2^2}{2\|w^{j(t)}\|_2} + \lambda_2 \text{tr}(\mathbf{W}^{(t+1)T} \mathbf{X}^T \mathbf{L}\mathbf{X}\mathbf{W}^{(t+1)}) \\ &\leq \sum_{i=1}^n \frac{\|z^{i(t)}\|_2^2}{2\|z^{i(t)}\|_2} + \lambda_1 \sum_{j=1}^d \frac{\|w^{j(t)}\|_2^2}{2\|w^{j(t)}\|_2} + \lambda_2 \text{tr}(\mathbf{W}^{(t)T} \mathbf{X}^T \mathbf{L}\mathbf{X}\mathbf{W}^{(t)}) \end{aligned} \tag{18}$$

For each i , we are able to gain as follows:

$$\|z^{i(t+1)}\|_2 - \frac{\|z^{i(t+1)}\|_2^2}{2\|z^{i(t)}\|_2} \leq \|z^{i(t)}\|_2 - \frac{\|z^{i(t)}\|_2^2}{2\|z^{i(t)}\|_2} \tag{19}$$

For each j , we are able to gain as follows:

$$\|w^{j(t+1)}\|_2 - \frac{\|w^{j(t+1)}\|_2^2}{2\|w^{j(t)}\|_2} \leq \|w^{j(t)}\|_2 - \frac{\|w^{j(t)}\|_2^2}{2\|w^{j(t)}\|_2} \tag{20}$$

Then combine (19) and (20) together we have:

$$\begin{aligned} &\sum_{i=1}^n \|z^{i(t+1)}\|_2 - \frac{\|z^{i(t+1)}\|_2^2}{2\|z^{i(t)}\|_2} + \lambda_1 \sum_{j=1}^d \|w^{j(t+1)}\|_2 - \frac{\|w^{j(t+1)}\|_2^2}{2\|w^{j(t)}\|_2} \\ &\leq \sum_{i=1}^n \|z^{i(t)}\|_2 - \frac{\|z^{i(t)}\|_2^2}{2\|z^{i(t)}\|_2} + \lambda_1 \sum_{j=1}^d \|w^{j(t)}\|_2 - \frac{\|w^{j(t)}\|_2^2}{2\|w^{j(t)}\|_2} \end{aligned} \tag{21}$$

Then combine (18) and (21) we have:

$$\begin{aligned} &\sum_{i=1}^n \|z^{i(t+1)}\|_2^2 + \lambda_1 \sum_{j=1}^d \|w^{j(t+1)}\|_2^2 + \lambda_2 \text{tr}(\mathbf{W}^{(t+1)T} \mathbf{X}^T \mathbf{L}\mathbf{X}\mathbf{W}^{(t+1)}) \\ &\leq \sum_{i=1}^n \|z^{i(t)}\|_2^2 + \lambda_1 \sum_{j=1}^d \|w^{j(t)}\|_2^2 + \lambda_2 \text{tr}(\mathbf{W}^{(t)T} \mathbf{X}^T \mathbf{L}\mathbf{X}\mathbf{W}^{(t)}) \end{aligned} \tag{22}$$

From the above analyzes, and then unite (9) and $\mathbf{H} = \mathbf{I}_n - (1/n)\mathbf{e}\mathbf{e}^T \in \mathbf{R}^{n \times n}$, we have:

$$\begin{aligned} &\|\mathbf{X} - \mathbf{X}\mathbf{W}^{(t+1)} - \mathbf{e}\mathbf{b}\|_F + \lambda_1 \|\mathbf{W}^{(t+1)}\|_F + \lambda_2 \text{tr}(\mathbf{W}^{(t+1)T} \mathbf{X}^T \mathbf{L}\mathbf{X}\mathbf{W}^{(t+1)}) \\ &\leq \|\mathbf{X} - \mathbf{X}\mathbf{W}^{(t)} - \mathbf{e}\mathbf{b}\|_F + \lambda_1 \|\mathbf{W}^{(t)}\|_F + \lambda_2 \text{tr}(\mathbf{W}^{(t)T} \mathbf{X}^T \mathbf{L}\mathbf{X}\mathbf{W}^{(t)}) \end{aligned} \tag{23}$$

Obviously, the value of objective function (7) decreases in each iteration [14, 20, 23]. Further, the objective function (7) will converge globally because it is a convex function [22].

4 Experiment

In this section, we introduce experimental setting in Section 4.1 firstly and we provide a brief introduction of the methods that will be compared with our method in Section 4.2. Then we summarize and analysis the experimental results by comparing our proposed method with other comparative methods in Section 4.3.

Table 1 Datasets summarization

Datasets	Instance	Feature	Class
SPECTF_Heart	267	44	2
LungCancer	32	56	2
Sonar	208	60	2
Movements	360	90	15
USPS	9298	256	10
Arrhythmia	452	279	13
Yeast	1484	1470	10
FERET	1400	6400	200

4.1 Experimental setting

The experimental environment is a Window XP system, and Matlab 7.11.0 is used to implement all the algorithms. In our experiments, we conduct the 10-fold cross-validation method for all methods. The final result was computed by averaging the results from all experiments. We apply the proposed UFSS method and the comparison methods to the classification task and evaluate them on eight datasets in terms of three different evaluations, i.e., classification accuracy, STandard Deviation (STD) and coefficient of variation. Specifically, we compare our methods with other methods in dimension reduction for feature selection, and then we use Support Vector Machine (SVM) [25] to conduct classification via the LIBSVM toolbox.¹ These datasets contain binary datasets and multi-class datasets, including SPECTF_Heart, LungCancer, Sonar, Movements, Arrhythmia and Yeast. They are all downloaded from UCI Machine Learning Repository,² the USPS dataset is downloaded from the website of Feature Selection Data sets,³ while the FERET dataset is downloaded from the website of CSDN.⁴ We summarized datasets in Table 1.

Three kinds of evaluation metrics as the evaluations for the classification task, i.e., classification accuracy, STandard Deviation (STD for short) and Coefficient of Variation (CV for short), respectively. The higher accuracy the algorithm is, the better classification performance it is. The smaller STD and CV the algorithm is, the more stable and robust it is.

4.2 Comparison methods

The comparison methods are introduced as follows:

- PCA: The method is a common dimensionality reduction method, which is used for extracting the important feature components from data [8].
- TRACK: The method mainly takes advantages of trace ratio formulation and K-means clustering to select the most discriminative features [33].
- RSR: The method joints sparse regularization and semi-supervised learning to select the most informative features, which can make the classifier robust for outliers [6].

¹<http://www.csie.nu.edu.tw/cjlin/libsvm/>

²UCI Repository of Machine Learning Datasets, <http://archive.ics.uci.edu>

³<http://featureselection.asu.edu/datasets.php>

⁴<http://download.csdn.net/download/zh920307/6844115>

Table 2 The results of Classification Accuracy (mean±STD)

Datasets	PCA	TRACK	RSR	FSR_ALM	UFSS
SPECTF_Heart	0.7737 ± 0.91	0.7982 ± 0.85	0.7940 ± 0.03	0.8005 ± 1.55	0.8242 ± 0.04
LungCancer	0.7350 ± 3.14	0.7733 ± 2.91	0.7358 ± 5.20	0.7183 ± 0.73	0.7967 ± 2.27
Sonar	0.7654 ± 1.74	0.7617 ± 1.33	0.7444 ± 1.42	0.7825 ± 0.85	0.8506 ± 0.92
Movements	0.8009 ± 1.17	0.7947 ± 1.29	0.8042 ± 0.95	0.7781 ± 1.76	0.8286 ± 0.87
USPS	0.9482 ± 0.05	0.9323 ± 0.28	0.9614 ± 0.06	0.9613 ± 0.07	0.9663 ± 0.07
Arrhythmia	0.6334 ± 1.22	0.6695 ± 0.88	0.6727 ± 1.36	0.6747 ± 0.96	0.6839 ± 0.07
Yeast	0.3547 ± 0.61	0.3645 ± 2.17	0.4196 ± 0.57	0.4232 ± 0.28	0.4404 ± 1.24
FERET	0.5949 ± 0.57	0.6009 ± 0.28	0.5980 ± 0.63	0.6015 ± 0.85	0.6286 ± 0.59

The bold emphasis are the results from our methods

- FSR_ALM: The method directly uses $\ell_{2,0}$ -norm constraint to exact Top-k Feature Selection and augmented Lagrangian method is used to tackle the constrained optimization problem [5].

4.3 Experimental results

We presented the mean of classification accuracy and the corresponding SStandard Deviation(STD), of all algorithms on the eight datasets in Table 2. Table 3 shows the Coefficient of Variation (CV) of all algorithms on eight datasets. We listed classification accuracy (the mean of classification accuracy in ten iterations) of all algorithms on eight datasets in Figure 1 where the horizontal axis denotes the iterations and the vertical axis denotes the classification accuracy.

In regard to classification accuracy accuracy and SStandard Deviation (STD) in Table 2 and Figure 1, we have the following observations:

- the proposed method UFSS improves the classification accuracies on average over eight datasets about by 4.2% (vs. PCA), 3.3% (vs. TRACK), 2.9% (vs. RSR), 2.8% (vs. FSR-ALM). In addition, according to Figure 1, we can also easily find that the proposed method UFSS almost has higher accuracy than four comparison algorithms in each of iteration. The reason is that, all the four comparison algorithms, i.e., PCA, TRACK, RSR and FSR-ALM, are subspace learning methods and they mainly only consider one kind of correlation inherent in data, while the proposed UFSS considers jointing self-representation and subspace learning for unsupervised feature selection to

Table 3 The result of Coefficient of Variation (mean)

Datasets	PCA	TRACK	RSR	FSR_ALM	UFSS
SPECTF_Heart	1.18	1.06	0.04	1.94	0.05
LungCancer	4.27	3.76	7.07	1.02	2.85
Sonar	2.27	1.73	1.91	1.09	1.08
Movements	1.46	1.62	1.18	2.26	1.05
USPS	0.05	0.30	0.06	0.07	0.07
Arrhythmia	1.93	1.31	2.02	1.42	0.89
Yeast	1.72	5.95	1.36	0.66	2.82
FERET	0.96	0.47	1.05	1.41	0.94

The bold emphasis are the results from our methods

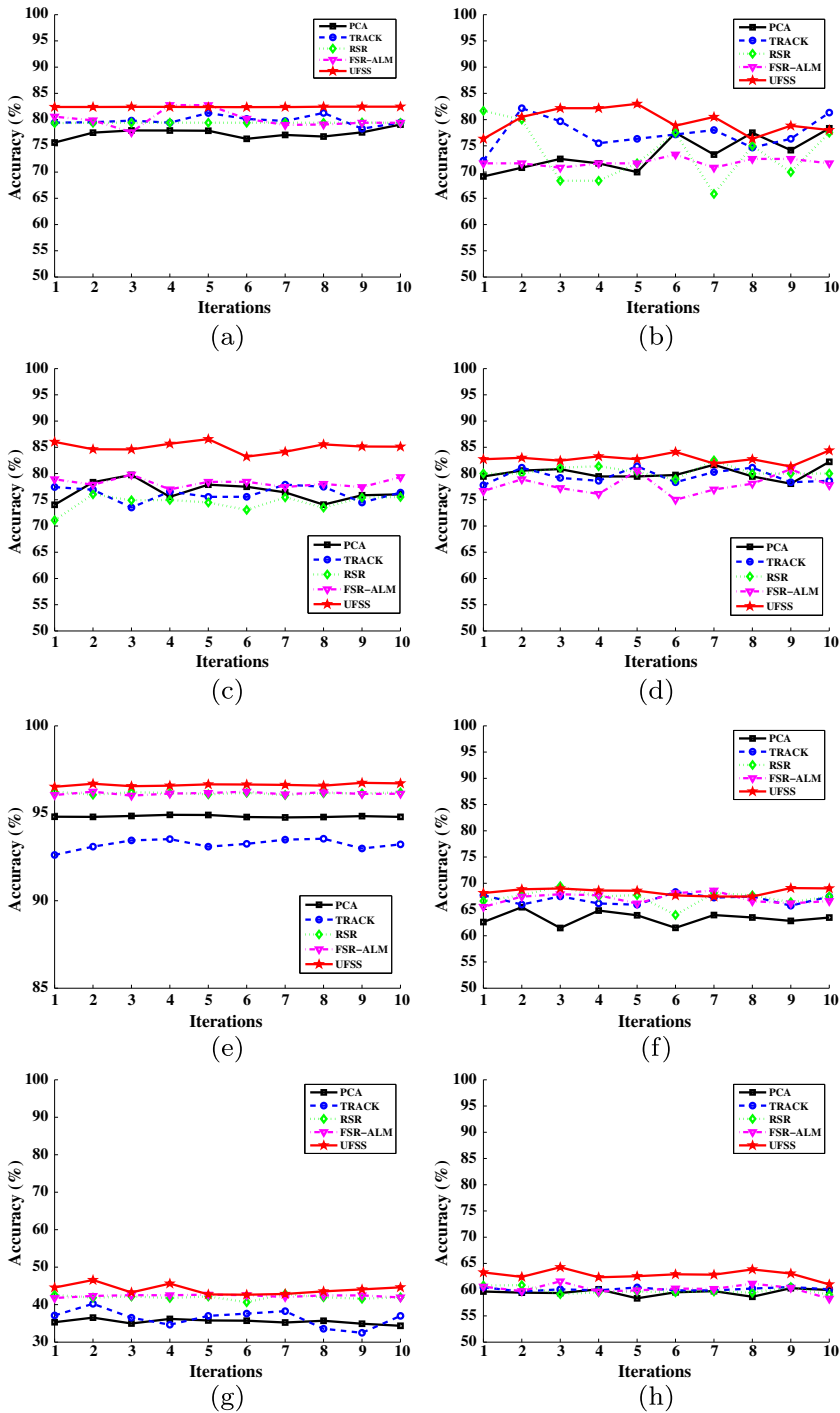


Figure 1 Classification accuracy on eight datasets in ten iterations. Note that **a** SPECTF_Heart; **b** LungCancer; **c** Sonar; **d** Movements; **e** USPS; **f** Arrhythmia; **g** Yeast; and **h** FERET

obtain two kinds of correlation inherent in data. At the same time, the proposed UFSS introduces a low-rank constraint to find the effective low-dimensional structures of the data, which can guarantee to reduce the redundancy. Therefore, the UFSS can get better performances than other comparison algorithms.

- The proposed method UFSS outperforms PCA, TRACK, RSR and FSR-ALM much better on LungCancer dataset, they are about 8.5%, 8.9%, 10.6% and 6.8%, respectively. The UFSS method absorbs the merits of both self-representation and LPP and integrates them into a unified framework. Thus it can select the most discriminative features and achieve the best classification accuracy.
- The proposed method UFSS has the least values of Standard Deviation (STD) compared with the comparison methods, it is less than others on average over eight datasets about by 0.28 (vs. PCA), 0.34 (vs. TRACK), 0.36 (vs. RSR), 0.05 (vs. FSR-ALM). It shows that our proposed method is more stable than other comparison methods.

In regard to Coefficient of Variation (CV) in Table 3, we can observe that: the UFSS method cannot always get the best performance on all datasets. For example, SPECTF_Heart and FERET UPSS is not the least one. But in general, the proposed method UFSS achieved the least coefficient of variation, i.e., 0.97, while other comparison methods are 1.39, 1.62, 1.47 and 0.99 corresponding to PCA, TRACK, RSR and FSR-ALM, respectively. This shows that the proposed algorithm UFSS is more robust than other comparison algorithms on the whole.

5 Conclusion

In this paper, we have proposed an unsupervised feature selection method based on self-representation and subspace learning. We use all features to represent each feature. In other words, each feature is reconstructed by all the features. A Frobenius norm regularizer term to constrain the reconstruction coefficient matrix is used to overcome the over-fitting problem and to select the most discriminative features. Also, we introduce Locality Preserving Projection (LPP) as a regularization term to maintain the local adjacent relation of the data constant when performing feature space transformation. Further, we consider applying a low-rank constraint to find the effective low-dimensional structures of the data. Experiments on real datasets have been conducted to compare the performances of the proposed method and the other state-of-the-art methods. The experimental results showed that the proposed method UFSS outperformed other methods in terms of classification accuracy, standard deviation and coefficient of variation.

In future, we consider improving the UFSS method for supervised feature selection.

Acknowledgments This work was in part supported by the Marsden Fund of New Zealand and the China Scholarship Council.

References

1. Bermejo, P., Gámez, J.A., Puerta, J.M.: A grasp algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. *Pattern Recogn. Lett.* **32**(5), 701–711 (2011)
2. Cai, D., He, X., Han, J.: Spectral regression: a unified approach for sparse subspace learning. In: *IEEE ICDM*, pp. 73–82 (2007)

3. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: ACM SIGKDD, pp. 333–342 (2010)
4. Cai, X., Ding, C., Nie, F., Huang, H.: On the equivalent of low-rank linear regressions and linear discriminant analysis based regressions. In: ACM SIGKDD, pp. 1124–1132 (2013)
5. Cai, X., Nie, F., Huang, H.: Exact top-k feature selection via $l_2, 0$ -norm constraint. In: IJCAI, vol. 13, pp. 1240–1246 (2013)
6. Chang, X., Nie, F., Yang, Y., Huang, H.: A convex formulation for semi-supervised multi-label feature selection. In: AAAI, pp. 1171–1177 (2014)
7. Chen, X.-W., Zeng, X., van Alphen, D.: Multi-class feature selection for texture classification. *Pattern Recognit. Lett.* **27**(14), 1685–1691 (2006)
8. Gottumukkal, R., Asari, V.K.: An improved face recognition technique based on modular pca approach. *Pattern Recognit. Lett.* **25**(4), 429–436 (2004)
9. Gu, Q., Li, Z., Han, J.: Joint feature selection and subspace learning. In: IJCAI, vol. 22(1), p. 1294
10. Hall, M.A., Smith, L.A.: Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In: FLAIRS, vol. 1999, pp. 235–239 (1999)
11. He, X., Niyogi, P.: Locality preserving projections. In: NIPS, pp. 153–160 (2004)
12. Hu, R., Zhu, X., Cheng, D., He, W., Yan, Y., Song, J., Zhang, S.: Graph self-representation method for unsupervised feature selection. *Neurocomputing* **220**, 130–137 (2017)
13. Liu, R., Yang, N., Ding, X., Ma, L.: An unsupervised feature selection algorithm: Laplacian score combined with distance-based entropy measure. In: IEEE IITA, vol. 3, pp. 65–68 (2009)
14. Lu, C., Lin, Z., Yan, S.: Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization. *IEEE Trans. Image Process.* **24**(2), 646–654 (2015)
15. Nikitidis, S., Tefas, A., Pitas, I.: Maximum margin projection subspace learning for visual data analysis. *IEEE Trans. Image Process.* **23**(10), 4413–4425 (2014)
16. Qian, M., Zhai, C.: Robust unsupervised feature selection. In: IJCAI, pp. 1621–1627 (2013)
17. Sebban, M., Nock, R.: A hybrid filter/wrapper approach of feature selection using information theory. *Pattern Recognit.* **35**(4), 835–846 (2002)
18. Swiniarski, R.W., Skowron, A.: Rough set methods in feature selection and recognition. *Pattern Recognit. Lett.* **24**(6), 833–849 (2003)
19. Tabakhi, S., Moradi, P., Akhlaghian, F.: An unsupervised feature selection algorithm based on ant colony optimization. *Eng. Appl. Artif. Intell.* **32**, 112–123 (2014)
20. Velu, R., Reinsel, G.C.: *Multivariate reduced-rank regression: theory and applications*, vol. 136. Springer Science Business Media, New York (2013)
21. Wang, T., Qin, Z., Zhang, S., Zhang, C.: Cost-sensitive classification with inadequate labeled data. *Inf. Syst.* **37**(5), 508–516 (2012)
22. Wang, H., Gao, Y., Shi, Y., Wang, R.: Group-based alternating direction method of multipliers for distributed linear classification. In: *IEEE transactions on cybernetics*. <https://doi.org/10.1109/TCYB.2016.2570808>, pp. 1–15 (2016)
23. Wu, J., Long, J., Liu, M.: Evolving rbf neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. *Neurocomputing* **148**, 136–142 (2015)
24. Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(1), 40–51 (2007)
25. Yi, P., Song, A., Guo, J., Wang, R.: Regularization feature selection projection twin support vector machine via exterior penalty. *Neural Comput. Appl.* 1–15 (2016)
26. Zhang, S.: Shell-neighbor method and its application in missing data imputation. *Appl. Intell.* **35**(1), 123–133 (2011)
27. Zhang, S., Jin, Z., Zhu, X.: Missing data imputation by utilizing information within incomplete instances. *J. Syst. Softw.* **84**(3), 452–459 (2011)
28. Zhang, S., Cheng, D., Zong, M., Gao, L.: Self-representation nearest neighbor search for classification. *Neurocomputing* **195**, 137–142 (2016)
29. Zhang, S., Li, X., Zong, M., Zhu, X., Cheng, D.: Learning k for knn classification. *ACM Trans. Intell. Syst. Technol.* **8**(3), 43 (2017)
30. Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R.: Efficient knn classification with different numbers of nearest neighbors. *IEEE Trans. Neural Netw. Learn. Syst.* 1–12 <https://doi.org/10.1109/TNNLS.2017.2673241> (2017)
31. Zhu, X., Zhang, S., Jin, Z., Zhang, Z., Xu, Z.: Missing value estimation for mixed-attribute data sets. *IEEE Trans. Knowl. Data Eng.* **23**(1), 110–121 (2011)
32. Zhu, X., Zhang, L., Huang, Z.: A sparse embedding and least variance encoding approach to hashing. *IEEE Trans. Image Process.* **23**(9), 3737–3750 (2014)

33. Zhu, P., Zuo, W., Zhang, L., Hu, Q., Shiu, S.C.: Unsupervised feature selection by regularized self-representation. *Pattern Recognit.* **48**(2), 438–446 (2015)
34. Zhu, X., Li, X., Zhang, S.: Block-row sparse multiview multilabel learning for image classification. *IEEE Trans. Cybern.* **46**(2), 450–461 (2016)
35. Zhu, X., Suk, H.-I., Lee, S.-W., Shen, D.: Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification. *IEEE Trans. Biomed. Eng.* **63**(3), 607–618 (2016)
36. Zhu, X., Li, X., Zhang, S., Ju, C., Wu, X.: Robust joint graph sparse coding for unsupervised spectral feature selection. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(6), 1263–1275 (2017)
37. Zhu, X., Li, X., Zhang, S., Xu, Z., Yu, L., Wang, C.: Graph pca hashing for similarity search. *IEEE Trans. Multimed.* **19**(9), 2033–2044 (2017)
38. Zhu, X., Suk, H., Wang, L., Lee, S., Shen, D.: A novel relational regularization feature selection method for joint regression and classification in AD diagnosis. *Med. Image Anal.* **38**, 205–214 (2017)
39. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. *J. Comput. Graph. Stat.* **15**(2), 265–286 (2006)