International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 28, No. 4 (2014) 1450009 (29 pages)

DOI: 10.1142/S0218001414500098

World Scientific
www.worldscientific.com

SPARSITY SCORE: A NOVEL GRAPH-PRESERVING FEATURE SELECTION METHOD

MINGXIA LIU*,† and DAOQIANG ZHANG*,‡

*School of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing 210016, P. R. China

†School of Information Science and Technology,
Taishan University, Taian 271021, P. R. China
‡dqzhang@nuaa.edu.cn

Received 1 April 2013
Accepted 10 April 2014
Published 22 May 2014

As thousands of features are available in many pattern recognition and machine learning applications, feature selection remains an important task to find the most compact representation of the original data. In the literature, although a number of feature selection methods have been developed, most of them focus on optimizing specific objective functions. In this paper, we first propose a general graph-preserving feature selection framework where graphs to be preserved vary in specific definitions, and show that a number of existing filter-type feature selection algorithms can be unified within this framework. Then, based on the proposed framework, a new filter-type feature selection method called sparsity score (SS) is proposed. This method aims to preserve the structure of a pre-defined l_1 graph that is proven robust to data noise. Here, the modified sparse representation based on an l_1-norm minimization problem is used to determine the graph adjacency structure and corresponding affinity weight matrix simultaneously. Furthermore, a variant of SS called supervised SS (SuSS) is also proposed, where the l_1 graph to be preserved is constructed by using only data points from the same class. Experimental results of clustering and classification tasks on a series of benchmark data sets show that the proposed methods can achieve better performance than conventional filter-type feature selection methods.

Keywords: Feature selection; sparse representation; l_1 graph; clustering; classification.

1. Introduction

In many pattern recognition and machine learning applications, the number of features (or variables) is becoming much higher, and is even higher than that of the observations.[15,22,48] For example, there are usually tens of thousands of features in neuroimaging data, while the number of subjects is very limited.[57] In this case, a

learning model will face several challenges, which are as follows[12,15,34]:

(i) Noisy features: It is common to obtain noisy features in the process of feature extraction, especially for high-dimensional data. The noisy components in features may affect the right representation of data and then lead to the over-fitting problem, especially when there is only small number of data points for each class.
(ii) Small sample size and high dimensionality. It is a well-known challenge to train a model for small sample sized and high-dimensional data in statistics and pattern recognition areas. Without feature selection, directly performing classification or clustering in original high-dimensional data space is both difficult and time-consuming. Moreover, irrelevant features may degrade the performance of learners.

Thus, to perform classification or clustering in original data space is both difficult and time-consuming.[32,34,48] In the literature, feature selection (or variable selection) has been shown effective in solving the small sample size problem by reducing feature dimension to eliminate noisy or redundant features, and thus help improve learning performances and facilitate data understanding.[20,24,25,28,46,49] Recently, several studies have shown that graphs constructed in original feature space reflect some intrinsic properties of data, and thus can be used for dimension reduction.[36,52] Intuitively, features that can best preserve such graph structures are informative, because the graph structures reveal inherent characteristics of original data. However, most of the current feature selection studies do not evaluate features through their graph-preserving abilities.

Accordingly, in this paper, we first propose a general graph-preserving feature selection framework, to preserve the structure of a pre-defined graph in original feature space. More specifically, the better a feature respect the predefined graph structure, the more important it would be. As we will show in the rest of this paper, many popular filter-type feature selection algorithms, such as variance (Var),[6] Fisher score (FS),[6] Laplacian score (LS),[21] and constraint score (CS)[56] can be reformulated within this framework. In other words, the proposed graph-preserving framework provides a unified view to reconsider many existing feature selection methods. In addition, based on the proposed framework, one can develop new feature selection algorithms efficiently.

Second, we propose two new filter-type feature selection methods named sparsity score (SS) and supervised sparsity score (SuSS), based on l_1 graph constructed by all training samples and only within-class ones, respectively. Here, the modified sparse representation (MSR) based on l_1-norm minimization problem is used to determine the graph adjacency structure and corresponding graph weights simultaneously. That is, the proposed feature selection methods aim to select features that can best preserve the l_1 graph structure that is proven robust to data noise.[13] Hence, the advantage of the proposed methods is that it is very likely to eliminate noisy features

and to find the most compact representation of data, comparing to those preserving other kinds of graph structures. To the best of our knowledge, no previous feature selection research has tried to devise a general graph-preserving feature selection framework and to propose l_1 graph-preserving feature selection methods.

The rest of this paper is organized as follows. Section 2 introduces the background by briefly reviewing several filter-type feature selection algorithms. In Sec. 3, we present the proposed general graph-preserving feature selection framework and indicate its relationship with existing feature selection methods. Section 4 introduces the proposed l_1 graph-preserving SS and SuSS methods in detail. In Sec. 5, we report the experimental results on a number of data sets, by comparing the proposed methods with several established feature selection methods. Conclusion is given in Sec. 6.

2. Backgrounds

Typically, there are two main categories for feature selection, i.e. filter-type methods and wrapper-type methods.[19] Wrapper-type methods require one pre-defined learning algorithm, and its performance is evaluated on each candidate feature subset to determine the optimal feature subset.[38,40,42,54] As they choose features that are better suited to the pre-defined learning algorithm, wrapper-type feature selection methods tend to give superior performance in terms of accuracy comparing to filter-type methods, but are usually computationally more expensive.[7] Unlike wrapper-type methods, filter-type methods select features according to mutual information, correlation, or other criteria,[12,27,29,55] and involve no learning algorithm. Hence, filter-type methods are usually adopted in practice due to their simplicity and computational efficiency, especially in the case with huge number of features.[44]

Within filter-type feature selection methods, different algorithms can be further categorized into two groups,[19] i.e. (i) feature ranking methods and (ii) subset search methods. The subset search methods evaluate the "goodness" of each candidate feature subset and select the optimal one according to specific evaluation measures, such as consistency, correlation and information measure, coupled with various search strategies.[38,54] However, subset selection methods are usually time-consuming because they consider feature selection as a combinatorial problem. In contrast, feature ranking methods consider features individually and achieve a ranked list of selected features ordered by their importance.[2,31,35,58] Thus, feature ranking methods are usually computationally more efficient and are very scalable to data sets with huge number of samples and high dimensionality.[15,55] In this study, we focus on feature ranking methods.

Among a huge literature on feature ranking methods, variance,[6] LS,[21] FS[6] and CS[56] are typical examples. Recently, several new methods are proposed based on these popular ones, such as constrained LS[5] and CS-4.[26] We now briefly introduce some of typical ones as follows.

Given a set of data samples $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N], \boldsymbol{x}_i \in R^d$, where N is the number of data points and d is the feature dimension. Let f_{ri} denote the rth feature of the ith sample $\boldsymbol{x}_i$. Denote the mean of the rth feature as $\mu_r = \frac{1}{N}\sum_{i=1}^{N} f_{ri}$. For supervised learning problems, class labels of the data points are given in $\{1, 2, \ldots, P\}$, where P is the number of classes. Let N_p denote the number of data points belonging to the pth class. Moreover, for semi-supervised feature selection methods, a part of prior knowledge such as class labels or pair-wise constraints is provided in specific ways.

As the simplest unsupervised evaluation of features, Var utilizes the variance along a feature dimension to reflect the feature's representative power for the original data. The variance of the rth feature denoted as Var_r, which should be maximized, is computed as follows[6]:

$$\mathrm{Var}_r = \frac{1}{N}\sum_{i=1}^{N}(f_{ri} - \mu_r)^2. \tag{1}$$

As another unsupervised method, LS prefers features with larger variances as well as stronger locality preserving ability. A key assumption in LS is that the data points from the same class should be close to each other. The LS of the rth feature denoted as LS_r, which should be minimized, is computed as follows[21]:

$$\mathrm{LS}_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 \mathbf{S}_{ij}}{\sum_i (f_{ri} - \mu_r)^2 \mathbf{D}_{ii}}. \tag{2}$$

Here, $\mathbf{D}$ is a diagonal matrix and $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$, where $\mathbf{S}_{ij}$ is defined by the neighborhood relationship between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as follows:

$$\mathbf{S}_{ij} = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}}, & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where t is a constant to be set, and the term "$\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are neighbors" means that either $\boldsymbol{x}_i$ is among k nearest neighbors of $\boldsymbol{x}_j$, or $\boldsymbol{x}_j$ is among k nearest neighbors of $\boldsymbol{x}_i$.

FS is a supervised method using full class labels. It seeks features that can maximize the distance of data points between different classes and minimize the distance of data points within the same class simultaneously. Let μ_r^p and f_r^p be the mean and the feature vector of class p corresponding to the rth feature, where $p \in \{1, \ldots, P\}$. Denote N_p as the sample number of the pth class. The FS of the rth feature denoted as FS_r, which should be maximized, is computed as follows[6]:

$$\mathrm{FS}_r = \frac{\sum_{p=1}^{P} N_p(\mu_r^p - \mu_r)^2}{\sum_{p=1}^{P}\sum_{i=1}^{N_p}(f_{ri}^p - \mu_r^p)^2}. \tag{4}$$

Finally, CS performs feature selection according to the constraint preserving ability of features. It utilizes $\mathbf{M} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | \, \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to the same class}\}$ containing pair-wise must-link constraints and $\mathbf{C} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | \, \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong}$

to different classes} containing pair-wise cannot-link constraints as the supervision information. Two constraint scores are developed including constraint score-1 (CS-1) and constraint score-2 (CS-2). The CSs of the rth feature denoted as CS_r^1 and CS_r^2, which should be minimized, are computed in the following forms[56]:

$$\mathrm{CS}_r^1 = \frac{\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathbf{M}} (f_{ri} - f_{rj})^2}{\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathbf{C}} (f_{ri} - f_{rj})^2}, \tag{5}$$

$$\mathrm{CS}_r^2 = \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathbf{M}} (f_{ri} - f_{rj})^2 - \lambda \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathbf{C}} (f_{ri} - f_{rj})^2, \tag{6}$$

where λ is a parameter to balance the two terms in Eq. (6).

3. General Graph-Preserving Feature Selection Framework

3.1. *Graph-preserving feature selection criterion*

Like many other graph-based methods, an essential step of our graph-preserving feature selection framework is graph construction, i.e. graph adjacency determination and graph weight assignment. For graph adjacency determination, there exits two popular ways, i.e. k-nearest neighbor method and ε-ball based method.[52] On the other hand, for graph weight assignment, a number of methods have been proposed, among which several most widely used methods are heat kernel,[52] inverse Euclidean distance[14] and local linear reconstruction distance.[43] In fact, various graph structures exhibit some intrinsic properties of the original data, which can be used to find the most useful features.

Accordingly, to preserve a specific graph structure constructed from original data, we define the graph-preserving feature selection criterion as follows:

$$\mathrm{Score}_r^1 = \frac{\mathbf{f}_r^T \mathbf{A} \mathbf{f}_\mathrm{r}}{\mathbf{f}_r^T \mathbf{B} \mathbf{f}_\mathrm{r}}, \tag{7}$$

$$\mathrm{Score}_r^2 = \mathbf{f}_r^T \mathbf{A} \mathbf{f}_\mathrm{r} - \lambda \mathbf{f}_r^T \mathbf{B} \mathbf{f}_\mathrm{r}, \tag{8}$$

where $\mathbf{f}_r$ is the rth feature, $\mathbf{A}$ and $\mathbf{B}$ are matrices that respect the graph structure in specific forms, and λ is a parameter to balance the two terms in Eq. (8).

In Eqs. (7) and (8), we define the importance of a feature by measuring its ability of respecting some graph structure that exhibits some properties of original data. To be specific, features that have stronger abilities to preserve the pre-defined graph structure are considered very important.

It is worth noting that the proposed graph-preserving feature selection criterion is quite general, bringing some additional advantages. First, as will be shown in Sec. 3.2, it brings us a unified framework from which we can reconsider existing feature selection methods through graphs. Second, one can easily develop new

feature selection methods based on the proposed graph-preserving feature selection criterion, by defining appropriate graphs and corresponding weight matrices in Eq. (7) or Eq. (8).

3.2. *Relationship with existing feature ranking methods*

According to different graph structures, several popular feature ranking methods can be classified as the following three categories, i.e. (i) global graph-preserving methods, (ii) neighborhood graph-preserving methods, and (iii) constraint graph-preserving methods.

3.2.1. *Global graph-preserving feature ranking*

Recall that variance seeks features with maximum variation. With simple algebraic formulation, Eq. (1) can be rewritten as follows:

$$\mathrm{Var}_r = \frac{1}{N}\mathbf{f}_r^T\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\mathbf{f}_r, \tag{9}$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{1} \in \mathrm{R}^N$ is a vector of all ones. Now we discuss the graph that variance preserves. Just like the graph constructed by PCA,[52] all the data samples in the intrinsic graph are connected with equal weight $1/N$. Let $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = \frac{1}{N}\mathbf{1}\mathbf{1}^T$, and $\lambda = 1$. And we find that variance follows the proposed graph-preserving feature selection criterion defined in Eq. (8).

In contrast to the variance, FS is supervised seeking features with best discriminative ability. It can be seen that Eq. (4) can be rewritten as follows:

$$\mathbf{FS}_r = \frac{\mathbf{f}_r^T\left(\sum_{p=1}^{P}\frac{1}{N_p}\mathbf{e}^p\mathbf{e}^{p^T} - \frac{1}{N}\mathbf{e}\mathbf{e}^T\right)\mathbf{f}_r}{\mathbf{f}_r^T\left(\mathbf{I} - \sum_{p=1}^{P}\frac{1}{N_p}\mathbf{e}^p\mathbf{e}^{p^T}\right)\mathbf{f}_r} = \frac{\mathbf{f}_r^T(\mathbf{S}_w - \mathbf{S}_b)\mathbf{f}_r}{\mathbf{f}_r^T(\mathbf{I} - \mathbf{S}_w)\mathbf{f}_r}, \tag{10}$$

where N_p is the instance number of class p, and $\mathbf{e}^p$ is an d-dimensional vector with $\mathbf{e}^p(i) = 1$, if x_i belongs to this class and 0 otherwise. Note that, $\mathbf{S}_w$ is actually the sum of weight matrices of P within-class graphs. In each within-class graph, all data points in a same class are connected with equal weight $1/N_p$. And $\mathbf{S}_b$ is the weight matrix for between-class graphs where edges connecting different classes have equal weight $1/N$. The graphs that FS preserve are P within-class graphs and one between-class graph, which are constructed in globally ways. Let $\mathbf{A} = \mathbf{S}_w - \mathbf{S}_b$ and $\mathbf{B} = \mathbf{I} - \mathbf{S}_w$. Thus, FS method follows the proposed graph-preserving feature selection criterion given in Eq. (7).

In summary, both variance and FS seek to preserve global graph structures. Naturally, we can incorporate them within the global graph-preserving methods in our proposed framework.

3.2.2. *Neighborhood graph-preserving feature ranking*

The mean of rth feature μ_r can be rewritten as

$$\mu_r = \sum_i \left(\mathbf{f}_r \frac{\mathbf{D}_{ii}}{\sum_i \mathbf{D}_{ii}} \right) = \frac{1}{\sum_i \mathbf{D}_{ii}} \left(\sum_i f_{ri} \mathbf{D}_{ii} \right) = \frac{\mathbf{f}_r^T \mathbf{D1}}{\mathbf{1}^T \mathbf{D1}}. \tag{11}$$

To remove the mean from the samples, we define

$$\tilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathrm{f}_r^T \mathbf{D1}}{\mathbf{1}^T \mathbf{D1}} \mathbf{1}, \tag{12}$$

where $\mathbf{D} = \mathrm{diag}(\mathbf{S1})$ and $\mathbf{1} = [1, \ldots, 1]^T$. After simple algebraic steps, we get the following:

$$\sum_i (f_{ri} - \mu_r)^2 \mathbf{D}_{ii} = \sum_i \left(f_{ri} - \frac{\mathbf{f}_r^T \mathbf{D1}}{\mathbf{1}^T \mathbf{D1}} \right)^2 \mathbf{D}_{ii} = \tilde{\mathbf{f}}_r^T \mathbf{D}\, \tilde{\mathbf{f}}_r, \tag{13}$$

$$\sum_{i,j} (f_{ri} - f_{rj})^2 \mathbf{S}_{ij} = \sum_{i,j} (f_{ri}^2 + f_{rj}^2 - 2 f_{ri} f_{rj}) \mathbf{S}_{ij} = 2 \mathbf{f}_r^T (\mathbf{D} - \mathbf{S}) \mathbf{f}_r = 2 \mathbf{f}_r^T \mathbf{L} \mathbf{f}_r, \tag{14}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is called Laplacian matrix.

It is easy to show that $\tilde{\mathbf{f}}_r^T \mathbf{L}\, \tilde{\mathbf{f}}_r = \mathbf{f}_r^T \mathbf{L} \mathbf{f}_r$ (with more details in Ref. 17). Thus, the objective function of LS can be rewritten as follows:

$$\mathbf{LS}_r = \frac{\tilde{\mathbf{f}}_r^T \mathbf{L}\, \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T \mathbf{D}\, \tilde{\mathbf{f}}_r}. \tag{15}$$

By minimizing $\tilde{\mathbf{f}}_r^T \mathbf{L}\, \tilde{\mathbf{f}}_r$ and maximizing $\tilde{\mathbf{f}}_r^T \mathbf{D}\, \tilde{\mathbf{f}}_r$ simultaneously, LS prefers features which respect the pre-defined graph and those with large variance. Let $\mathbf{A} = \mathbf{L}$ and $\mathbf{B} = \mathbf{D}$. Then, it can be seen that the LS follows the proposed graph-preserving feature selection criterion defined in Eq. (7). Note that, the graph that LS preserves is constructed by connecting data samples in a pre-defined neighborhood. So LS can be categorized as a neighborhood graph-preserving method.

3.2.3. *Constraint graph-preserving feature ranking*

The CS can also be explained from the proposed graph-preserving feature selection criterion. First, using the pair-wise constraints in $\mathbf{M}$ and $\mathbf{C}$, we construct two graphs $\mathbf{G}^M$ and $\mathbf{G}^C$ respectively, both of which have N nodes and ith node refers to the sample x_i. It is worth noting that, an edge will be set if there is a must-link or a cannot-link constraint in these two graphs. Then, the CS seeks features on which two data points are close in $\mathbf{G}^M$ and far away in $\mathbf{G}^C$. Once the graphs are constructed, their weight matrices denoted by $\mathbf{S}^M$ and $\mathbf{S}^C$ respectively, are defined as

Table 1. Graph-preserving view for several filter-type feature selection methods.

Algorithm	A and B Definition	Characteristics
Variance	$\mathbf{A} = \mathbf{I}$; $\mathbf{B} = \frac{1}{N}\mathbf{1}\mathbf{1}^T$	Unsupervised; global
Fisher score	$\mathbf{A} = \mathbf{S}_w - \mathbf{S}_b$; $\mathbf{B} = \mathbf{I} - \mathbf{S}_w$	Supervised; global
Laplacian score	$\mathbf{A} = \mathbf{L}$; $\mathbf{B} = \mathbf{D}$	Unsupervised; neighborhood
Constraint score	$\mathbf{A} = \mathbf{L}^M$; $\mathbf{B} = \mathbf{L}^C$	Semi-supervised; Constraint

$$\mathbf{S}_{ij}^M = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}}, & \text{if } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbf{M} \text{ or } (\boldsymbol{x}_j, \boldsymbol{x}_i) \in \mathbf{M} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$\mathbf{S}_{ij}^C = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}}, & \text{if } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbf{C} \text{ or } (\boldsymbol{x}_j, \boldsymbol{x}_i) \in \mathbf{C} \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where t is a constant to be set. Then, Eqs. (5) and (6) can be written as follows:

$$\mathrm{CS}_r^1 = \frac{\mathbf{f}_r^T \mathbf{L}^M \mathbf{f}_r}{\mathbf{f}_r^T \mathbf{L}^C \mathbf{f}_r}, \tag{18}$$

$$\mathrm{CS}_r^2 = \mathbf{f}_r^T \mathbf{L}^M \mathbf{f}_r - \lambda \mathbf{f}_r^T \mathbf{L}^C \mathbf{f}_r, \tag{19}$$

where $\mathbf{L}^M$ and $\mathbf{L}^C$ are the Laplacian matrices for the must-link graph and the cannot-link graph, respectively. Let $\mathbf{A} = \mathbf{L}^M$ and $\mathbf{B} = \mathbf{L}^C$, and one can see that the CS-1 and CS-2 follow the proposed graph-preserving criterion defined in Eqs. (7) and (8), respectively. Hence, the CS can be categorized as a constraint graph-preserving method.

So far, we find that the above-mentioned feature selection methods can be unified in the proposed graph-preserving feature selection framework, despite of different proposing motivations. Table 1 lists the graph-preserving matrices for different methods, with corresponding characteristics of different graphs. It is worth noting that, different graph construction rules and weight assignment methods will lead to different feature ranking methods, which motivates us to develop new feature selection methods based on the proposed graph-preserving feature selection framework.

4. Proposed L1 Graph-preserving Feature Selection Methods

In recent years, much attention has been focused on sparse linear representation with respect to an over-complete dictionary of base elements,[23,30,41,50,53] where an l_1 graph and its affinity weight matrix can be constructed automatically. Although, there is no clear evidence that any of graph structure and its affinity weight matrix are always superior to others based on the celebrated "No Free Lunch" theorem,[17] the l_1 graph owns a special characteristic that is sparsity.[9] Note that, sparsity provides us an important way to improve the robustness of a model to data noise. Inspired by this, we present two novel filter-type feature selection methods that preserve l_1 graph.[13,50]

For completeness, we will first briefly review sparse representation theory, and then go into the details of our proposed feature selection methods.

4.1. *Sparse representation*

As an extension to traditional signal representation such as Wavelet and Fourier representation, sparse representation has been applied extensively in pattern recognition and signal processing recently.[8,22,45,51] Given a signal $\boldsymbol{x} \in R^d$, and a matrix $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \boldsymbol{x}_N] \in R^{d \times N}$ which contains the elements of an over-complete dictionary in its columns, sparse representation aims to represent each $\boldsymbol{x}$ using as fewer entries of $\mathbf{X}$ as possible. It can be expressed formally as follows[11,50]:

$$\min_{\mathbf{s}} \|\mathbf{s}\|_0, \quad s.t.\ \boldsymbol{x} = \mathbf{X}\mathbf{s}, \tag{20}$$

where $\mathbf{s} \in R^N$ is the coefficient vector, and $\|\mathbf{s}\|_0$ is the pseudo-l_0 norm denoting the number of nonzero components in $\mathbf{s}$. However, to find the sparsest solution of Eq. (20) is NP-hard, and it can be approximately solved by the following[50]:

$$\min_{\mathbf{s}} \|\mathbf{s}\|_1, \quad s.t.\ \boldsymbol{x} = \mathbf{X}\mathbf{s}, \tag{21}$$

where $\|\mathbf{s}\|_1$ is the l_1 norm of $\mathbf{s}$. It has been proven that the solution of l_1 norm minimization problem is equal to that of l_0 norm minimization problem, provided that the solution $\mathbf{s}$ is sparse enough.[4,16] The problem defined in Eq. (21) can be solved by standard linear programming.[11]

In practice, the constraint $\boldsymbol{x} = \mathbf{X}\mathbf{s}$ in Eq. (21) does not always hold because there are often some noises existing in $\boldsymbol{x}$ and the sample size is generally less than that of features. In Ref. 39, two robust extensions are proposed to handle these problems: (i) Relaxing the constraint to be $\|\boldsymbol{x} - \mathbf{X}\mathbf{s}\| < \delta$, where δ can be regarded as an error tolerance. (ii) Replacing $\mathbf{X}$ with $[\mathbf{X}\ \mathbf{I}]$, where $\mathbf{I}$ is a d-order identity matrix.

4.2. *Sparse reconstructive weight*

Based on a MSR framework, researchers in Ref. 42 construct a sparse reconstructive weight matrix, and show such matrix helps to find the most compact representation of original data. For a classification problem, we assume that the training data are given as $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \boldsymbol{x}_N] \in R^{d \times N}$ where $\boldsymbol{x}_i \in R^d$. A sparse reconstructive weight vector $\mathbf{s}_i$ for each $\boldsymbol{x}_i$ can be obtained by solving the following modified l_1 minimization problem[41]:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_1, \quad s.t.\ \boldsymbol{x}_i = \mathbf{X}\mathbf{s}_i, 1 = \mathbf{1}^T\mathbf{s}_i, \tag{22}$$

where $\mathbf{s}_i = [s_{i,1}, \ldots, s_{i,i-1}, 0, s_{i,i+1}, \ldots s_{i,N}]^T$ is an N-dimensional vector in which the ith element is equal to zero implying that $\boldsymbol{x}_i$ is removed from $\mathbf{X}$. The element $s_{i,j}(j \neq i)$ denotes the contribution of each $\boldsymbol{x}_j$ to reconstruct $\boldsymbol{x}_i$, and $\mathbf{1} \in R^N$ is a vector of all ones.

For each sample $\boldsymbol{x}_i$, we can compute the reconstructive weight vector $\hat{\mathbf{s}}_i$, and then get the sparse reconstructive weight matrix $\mathbf{S} = (\hat{s}_{i,j})_{N\times N}$:

$$\mathbf{S} = [\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \ldots, \hat{\mathbf{s}}_N]^T, \tag{23}$$

where $\hat{\mathbf{s}}_i$ is the optimal solution of Eq. (22). Note that, the discriminative information can be naturally preserved in the matrix $\mathbf{S}$, even if no class label information is used. The reason is that the nonzero entries in $\hat{\mathbf{s}}_i$ usually correspond to the samples from the same class, which implies that $\hat{\mathbf{s}}_i$ may help to distinguish that class from the others. After obtaining the reconstruction weight matrix $\mathbf{S}$ through Eq. (23), the l_1 graph including both graph adjacency structure and affinity weights matrix can be simultaneously determined by $\mathbf{S}$.

In many real-world problems, the constraint $\mathbf{x}_i = \mathbf{X}\mathbf{s}_i$ does not always hold. To overcome this problem, two modified objective functions are proposed.[41] The first one is as follows:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_1, \quad s.t.\ \|\boldsymbol{x}_i - \mathbf{X}\mathbf{s}_i\| < \delta, 1 = \mathbf{1}^T\mathbf{s}_i, \tag{24}$$

where δ is the error tolerance. It can be seen that the optimal solution of Eq. (24) reflect some intrinsic geometric properties, e.g. invariant to translation and rotation. The second extension is expressed as follows:

$$\min_{[\mathbf{s}_i^T \mathbf{t}_i^T]^T} \left\| [\mathbf{s}_i^T \mathbf{t}_i^T]^T \right\|_1, \quad s.t. \begin{bmatrix} \boldsymbol{x}_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{X} & \mathbf{I} \\ \mathbf{1}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{s}_i \\ \mathbf{t}_i \end{bmatrix}, \tag{25}$$

where $\mathbf{t}_i$ is a d-dimensional vector incorporated as a reconstructive compensation term. The optimal solution of Eq. (25) is also invariant to translations, but the invariance to rotation and rescaling does not rigorously hold.

4.3. *Proposed sparsity score*

We are now in the position to derive our l_1 graph-preserving feature selection method, called SS, by using our proposed general framework in Sec. 3 as a platform. The first step is to compute the sparse reconstruction weight matrix defined in Eq. (23), through which we can obtain the l_1 graph adjacency structure.

Following the notations in previous sections, we define the SS (denoted as SS-1) of the rth feature (SS_r^1), which should be minimized, as follows:

$$\mathrm{SS}_r^1 = \sum_{i=1}^{m} \left(f_{ri} - \sum_{j=1}^{m} \hat{s}_{i,j} f_{rj} \right)^2 = \mathbf{f}_r^T (\mathbf{I} - \mathbf{S} - \mathbf{S}^T + \mathbf{S}\mathbf{S}^T)\mathbf{f}_r, \tag{26}$$

where $\hat{s}_{i,j}$ is the entry of the sparse reconstruction weight matrix $\mathbf{S}$ constructed using all data points. By minimizing SS_r^1, we prefer features that can best respect the predefined l_1 graph structure.

In order to further improve the proposed SS, we take the variance into consideration. Accordingly, another SS (denoted as SS-2) of the rth feature (SS_r^2) is

Algorithm 1. Sparsity Score-1 (SS-1) and Sparsity Score-2 (SS-2)

Input:

Data matrix $\mathrm{X} = [x_1, x_2 \ldots, x_N] \in R^{d \times N}$, where $x_i \in R^d$.

Output: The ranked feature list

Procedure:

Step 1. Calculate the sparse reconstruction weight vector $\hat{s}_i$ by utilizing all data points through Eq. (24) or Eq. (25). And then we obtain the sparse reconstructive weight matrix S using Eq. (23), where the l_1 graph structure can be constructed automatically;

Step 2. For each of all the d features, compute its sparsity score using Eqs. (26) and (27), respectively;

Step 3. Rank all features according to their sparsity scores in ascending order.

defined as

$$\mathrm{SS}_r^2 = \frac{\sum_{i=1}^{m} (f_{ri} - \sum_{j=1}^{m} \hat{\mathrm{s}}_{i,j} f_{rj})^2}{\frac{1}{m} \sum_{i=1}^{m} (f_{ri} - \mu_r)^2} = \frac{\mathbf{f}_r^T (\mathbf{I} - \mathbf{S} - \mathbf{S}^T + \mathbf{S}\mathbf{S}^T) \mathbf{f}_r}{\mathbf{f}_r^T \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T\right) \mathbf{f}_r}. \tag{27}$$

In Eqs. (26) and (27), we prefer features that can best preserve the l_1 graph structure and those with large variance that have stronger representative ability. That is, with smaller reconstruction error (i.e. to preserve the l_1 graph structure), as well as larger variance for rth feature, SSs tend to be small that means the feature would be more important. The detailed procedures of our proposed two SS methods are shown in Algorithm 1.

With $\mathbf{A} = \mathbf{I} - \mathbf{S} - \mathbf{S}^T + \mathbf{S}\mathbf{S}^T$ and $\lambda = 0$. Then the proposed SS-1 method can be unified into the graph-preserving feature selection framework through Eq. (8). Let $\mathbf{A} = \mathbf{I} - \mathbf{S} - \mathbf{S}^T + \mathbf{S}\mathbf{S}^T, \mathbf{B} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, and we find that the proposed SS-2 method can be unified into the graph-preserving feature selection framework through Eq. (7).

4.4. *Proposed supervised sparsity score*

The SS developed in the previous section are unsupervised, i.e. no class label information is used. In this section, we extend them to supervised versions, i.e. SuSS, to make full use of the valuable class label information. Accordingly, we define two SuSS functions (i.e. SuSS-1 and SuSS-2) as follows:

$$\mathrm{SuSS}_r^1 = \sum_{p=1}^{P} \sum_{i=1}^{N_p} \left(f_{ri}^p - \sum_{j=1}^{N_p} \hat{\mathrm{s}}_{ij}^p f_{rij}^p \right)^2 = \mathbf{f}_r^T \left(\mathbf{I} - \sum_{p=1}^{P} \frac{1}{N_p} \mathbf{e}^p \mathbf{e}^{p^T} \right) \mathbf{f}_r, \tag{28}$$

Algorithm 2. Supervised Sparsity Score-1 (SuSS-1) and Supervised Sparsity Score-2 (SuSS-2)

Input:

Data matrix $\mathrm{X} = [x_1, x_2 \ldots, x_N] \in R^{d \times N}$, where $x_i \in R^d$;

Full class label vector $lb = [lb_1, lb_2 \ldots, lb_N]$, where $lb_i \in \{1, \ldots, P\}$ and P is the class number;

Output: The ranked feature list

Procedure:

Step 1. For each class, we construct the l_1 graph by computing sparse reconstruction weight vector utilizing samples within the same class through Eq. (24) or Eq. (25). Then we construct the pth sparse reconstructive weight matrix S_p using Eq. (23);

Step 2. For each of all d features, compute the supervised sparsity scores using Eqs. (28) and (29), respectively;

Step 3. Rank all features according to their supervised sparsity scores in ascending order.

$$\begin{aligned} \mathrm{SuSS}_r^2 &= \frac{\sum_{p=1}^{P} \sum_{i=1}^{N_p} \left(f_{ri}^p - \sum_{j=1}^{N_p} \hat{\mathrm{s}}_{ij}^p f_{rij}^p \right)^2}{\sum_{p=1}^{P} \sum_{i=1}^{N_p} (f_{ri}^p - \mu_r^p)^2} \\ &= \frac{\sum_{p=1}^{P} \mathbf{f}_r^{p^T} (\mathbf{I} - \mathbf{S}_p - \mathbf{S}_p^T + \mathbf{S}_p \mathbf{S}_p^T) \mathbf{f}_r^p}{\mathbf{f}_r^T \left(\mathbf{I} - \sum_{p=1}^{P} \frac{1}{N_p} \mathbf{e}^p \mathbf{e}^{p^T} \right) \mathbf{f}_r}, \end{aligned} \tag{29}$$

where N_c is the number of samples of the pth class, $\hat{\mathrm{s}}_{ij}^p$ is the entry of pth sparse reconstruction weight matrix $\mathbf{S}_p$ constructed using only the data points of the pth class.

Similar to SS, we prefer features that can best respect a pre-defined l_1 graph structure in SuSS. Note that, l_1 graphs to be preserved in SuSS are P within-class graphs constructed using only within-class data points. In principle, features that can best respect such within-class graphs are more important. The detailed procedures of SuSS are summarized in Algorithm 2.

Let $\mathbf{A} = \mathbf{I} - \mathbf{S}_p - \mathbf{S}_p^T + \mathbf{S}_p \mathbf{S}_p^T$ and $\lambda = 0$. Then the proposed SuSS-1 method can be unified into the graph-preserving feature selection framework through Eq. (8). Denote $\mathbf{A} = \mathbf{I} - \mathbf{S}_p - \mathbf{S}_p^T + \mathbf{S}_p \mathbf{S}_p^T, \mathbf{B} = \mathbf{I} - \sum_{p=1}^{P} \frac{1}{N_p} \mathbf{e}^p \mathbf{e}^{p^T}$, and the proposed SuSS-2 method can be unified into the proposed graph-preserving feature selection framework through Eq. (7). Hence, the proposed supervised Sparisity Score can be derived from the proposed graph-preserving feature selection criterion.

Note that, both the proposed SS (including SS-1 and SS-2) and SuSS (including SuSS-1 and SuSS-2) feature ranking methods are l_1 graph-preserving methods. The sparse reconstruction weight matrix of l_1 graph is constructed globally. Hence, the proposed SS method is global and unsupervised, while the proposed SuSS method is

global and supervised within our proposed graph-preserving feature selection framework.

4.5. *Computational complexity analysis*

Now we analyze the computational complexity of Algorithm 1 and Algorithm 2. There are three main steps in Algorithm 1: (i) Step 1 constructs the l_1 graph by computing the sparse reconstruction weight matrix using Eq. (24) or Eq. (25), requiring $O(N^2)$ operation given N data points. (ii) Step 2 evaluates d features based on the l_1 graph requiring $O(dN^2)$ operations. (iii) Step 3 ranks d features which needs $O(d \log d)$ operations. Hence, the overall time complexity of Algorithm 1 is $O(d \max(N^2, \log d))$.

Similarly, the Algorithm 2 contains three parts: (i) Step 1 constructs the within-class l_1 graphs for P classes, requiring $O(N_{\max}^2)$ operations given $N_{\max}$ data points for the pth class, and $N_{\max} = \max\{N_1, N_2, \ldots, N_P\}$. (ii) Step 2 evaluates d features based on the l_1 graph requiring $O(dN_{\max}^2)$ operations. (iii) Step 3 ranks d features which needs $O(d \log d)$ operations. Hence, the overall time complexity of Algorithm 2 is $O(d \max(N_{\max}^2, \log d))$.

5. Experiments

To evaluate efficiency of our proposed methods, we perform both clustering and classification experiments on a number of data sets, by comparing our proposed methods with several popular feature selection methods.

5.1. *Clustering experiments*

In this subsection, we apply two proposed SS methods (i.e. SS-1 and SS-2) for clustering, comparing to Var and LS methods. Note that, we do not compare supervised methods because class labels are not available in clustering tasks.

5.1.1. *Data sets*

The clustering experiments are performed on several data sets from UCI machine learning repository[3] including *wine, ionosphere, sonar, spectf heart disease, digits*, and *steel plate faults.* These data sets have small or middle size of feature numbers, with class numbers ranging from two to seven. In addition, we also use two gene expression data sets, which are *colon cancer*[1] and *prostate cancer*[12] with small sample size and high-dimensional features. Characteristics of these data sets are summarized in Table 2.

5.1.2. *Experimental design for clustering*

For clustering experiments, we first obtain a feature ranking list by performing a specific feature selection method on a data set. Second, we choose the first m features

Table 2. UCI and gene expression data sets used in our experiments.

Data Set	#Dimension	#Class	#Sample	Description
Wine	13	3	178	classical , small feature size, multi-class
Ionosphere	33	2	351	classical, middle feature size
Sonar	60	2	208	classical, middle feature size
Spectf heart disease	44	2	267	classical, middle feature size
Digits 246	60	3	539	classical, middle feature size, multi-class
Steel plate faults	27	7	1941	classical, middle feature size, multi-class
Colon cancer	2000	2	62	high-dimensional, small sample size
Prostate cancer	12600	2	136	high-dimensional, small sample size

from the ranking list to form a feature subset, where $m = \{1, 2, \ldots, d\}$ and d is the feature dimension of original data. Then, a clustering process is performed based on data with such feature subset. By varying m from 1 to d, we obtain d different clustering results. Finally, we report the best clustering result, as well as corresponding feature size that is the optimal number of selected features. In our experiments, we use K-means algorithm to perform clustering. Specifically, the clustering process is repeated for 10 times with different initializations and the best result is recorded. Note that, the initialization is the same for different algorithms for fair comparison. Finally, we report the best clustering results as well the optimal number of selected features. In addition, we also report the results of baseline (i.e. results without any feature selection procedure).

By comparing the obtained label of each data points of K-means algorithm with that provided by the data corpus, the clustering result can be evaluated. We use F-Score metric[39] to measure the clustering performance. Given a clustering result, F-Score is defined as follows:

$$\text{F-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{30}$$

where Precision and Recall are two measure criteria, which are defined as follows:

$$\text{Precision} = \frac{N_1}{N_1 + N_2}, \tag{31}$$

$$\text{Recall} = \frac{N_1}{N_1 + N_3}, \tag{32}$$

where N_1 is the number of sample pairs which are clustered correctly, N_2 is the number of sample pairs that belong to the different classes but are clustered into the same class, and N_3 is the number of sample pairs that belong to the same class but are clustered into different classes.

5.1.3. *Clustering results*

Experimental results of clustering are reported in Table 3 and Fig. 1. In Table 3, the values in the brackets are optimal numbers of selected features, the underlined terms

Table 3. Clustering performance of different feature selection methods (%).

Data Set	Baseline	Var	LS	SS-1	SS-2
Wine	59.42 (13)	60.84 (1)	61.09 (1)	61.89 (12)	81.71 (2)
Ionosphere	63.58 (33)	63.82 (31)	65.24 (33)	70.20 (2)	69.38 (4)
Sonar	50.59 (60)	56.85 (2)	52.91 (9)	52.15 (1)	59.22 (2)
Spectf heart disease	66.18 (44)	66.18 (37)	66.34 (3)	69.81 (23)	70.14 (16)
Digits 246	97.79 (60)	97.79 (33)	97.79 (30)	97.79 (55)	97.79 (50)
Steel plate faults	27.41 (24)	27.92 (19)	27.81 (20)	28.85 (9)	38.78 (2)
Colon cancer	61.33 (2000)	69.69 (78)	69.69 (368)	77.13 (283)	69.8167 (167)
Prostate cancer	56.39 (12600)	56.7652 (1)	61.93 (174)	62.57 (1)	62.58 (1)

are the best results among different methods, and the baseline is achieved by using all features. Note that, the cluster number in this set of experiments is set as the true class number of a specific data set, if without extra explanations.

From Table 3, one can see that the clustering performances of SS-1 and SS-2 are usually better than that of the other two methods, especially on the *wine, ionosphere* and *steel plate faults* data sets. On the other hand, it is obvious to see that, in most cases, the numbers of optimal features selected by the proposed SS-1 and SS-2 are less than those of Var and LS.

Figure 1 plots the clustering results versus different numbers of selected features on several data sets, from which one can see that most methods achieve better performance than baselines when less than half features are selected. Meanwhile, one can see from Fig. 1 that, the proposed SS-1 and SS-2 methods usually achieve better performances than the other two methods. For example, on the *prostate cancer* data set that is high-dimensional with small sample size, the proposed SS-1 and SS-2 methods using only one feature can achieve better performances than the other methods. It illustrates that the proposed l_1 graph-preserving feature selection methods can solve the small sample size problem efficiently.

Furthermore, we investigate the influence of cluster numbers on the clustering performance. Table 4 and Fig. 2 report the results on the *steel plate faults* data set with different cluster numbers. From Table 4 and Fig. 2, we can find that the performances of the proposed SS-1 and SS-2 are quite stable with the increase of cluster numbers, and are always better than those of Var, LS and baseline. The reason may be that feature selection methods preserving l_1-graph can find more discriminative features than methods preserving other kinds of graphs, due to the fact that the l_1 graph is robust to data noise.[13,50]

5.2. *Classification experiments*

5.2.1. *Data sets*

Besides UCI and gene expression data sets used in Sec. 5.1, we also use a broad collection of texture images corresponding to two well-known real-world texture data

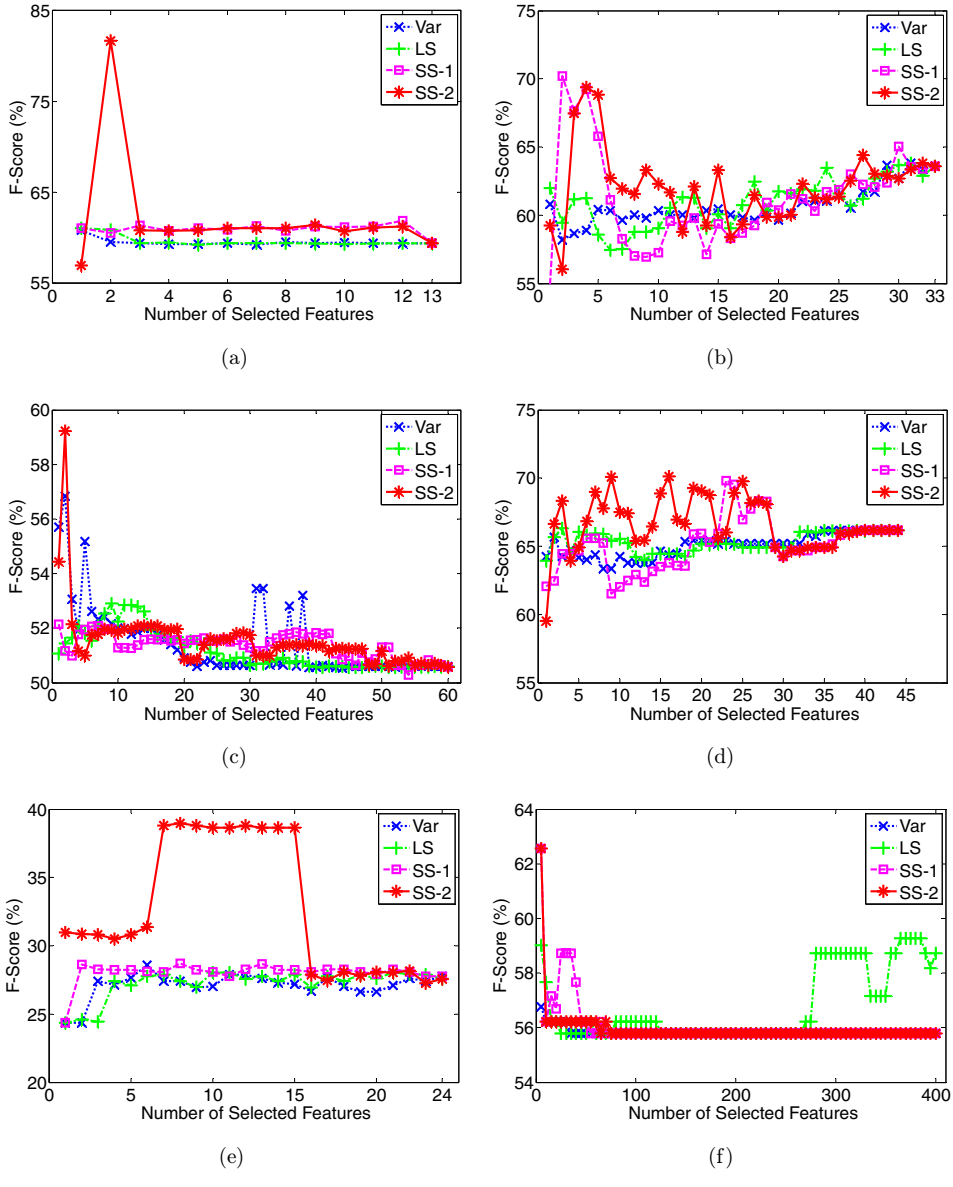


Fig. 1. Clustering results versus different numbers of selected features. (a) *Wine*, (b) *ionosphere*, (c) *sonar*, (d) *spectf heart disease*, (e) *steel plate faults* and (f) *prostate cancer*.

sets: *MeasTex*,54 and *VisTex*,[37] which are typical multi-class classification problems. Figure 3 shows example textures chosen from these data sets.

To obtain texture features, the wavelet package transform (WPT)[33] is used, which is a classical method for texture feature extraction. Specifically, we first perform a 4-level decomposition structure for WPT, and then compute the mean and

Table 4. Clustering performance on *steel plate faults* with different cluster numbers (%).

Cluster Number	Baseline	Var	LS	SS-1	SS-2
2	34.85 (24)	35.13 (21)	34.84 (1)	35.98 (23)	36.83 (7)
3	32.48 (24)	32.96 (13)	32.95 (12)	36.49 (3)	41.06 (15)
4	31.24 (24)	31.48 (3)	31.48 (9)	34.67 (2)	40.76 (7)
5	32.48 (24)	31.60 (11)	31.60 (15)	32.27 (4)	40.47 (9)
6	32.48 (24)	31.60 (11)	31.60 (15)	32.27 (4)	40.47 (9)
7	27.41 (24)	27.92 (19)	27.81 (20)	28.85 (9)	38.78 (2)

the variance of each coefficients matrix in the 4th level to construct the feature vector. Hence, features of texture images here is 128-dimensional. Characteristics of two texture data sets are shown in Table 5.

5.2.2. *Experimental design for classification*

We compare our proposed SS and SuSS methods with the following feature selection methods: (i) Var, (ii) LS, (iii) FS and (iv) Fisher–Markov selector with polynomial kernel (LFS)[12] that aims to find an global optimum of an objective function; (v) SVM-RFE (RFE)[18] that is a wrapper-type feature selection method. The code of LFS can be obtained from the authors (http://www2.cs.siu.edu/~qcheng/feature-selection/mrf.html), and the standard RFE algorithm is provided by the Spider package (http://people.kyb.tuebingen.mpg.de/spider/main.html). Because the CS method needs constraint information provided by hand, we do not compare our proposed methods with CS in the experiments. It is worth noting that, both variance and LS learn the feature scores without any class labels of training data, while Fisher Score, RFE and LFS need full class labels of the training data.

For fair comparison, we compare the proposed methods with other methods in the following way: (i) Sparsity Score-1 (SS-1) and Sparsity Score-2 (SS-2) are compared with variance and LS, which are in unsupervised manner. (ii) Supervised Sparsity Score-1 (SuSS-1) and Supervised Sparsity Score-2 (SuSS-2) are compared with Fisher Score, RFE and LFS, which are in supervised manner. In this set of experiments, the LIBSVM[10] using RBF kernel with default parameters is employed to perform classification.

In general, a 10-fold cross-validation strategy is adopted to compute the classification accuracy on the test set. To be specific, we first equally partition the whole data set into 10 subsets, and each time one of these subsets is utilized as the test set while the other nine subsets are combined together to be the training set. To confirm the optimal number of selected features in each fold, we further select 10% data from the training set to be the validate set, and use the remaining training data to perform feature ranking according to different feature selection methods. Second, we choose the first $m(m = \{1, 2, \ldots, d\})$ features from the ranking list generated by a specific feature selection method on the training data. Based on the training set with such feature subset, we construct a classification model. By varying m from 1 to d, we

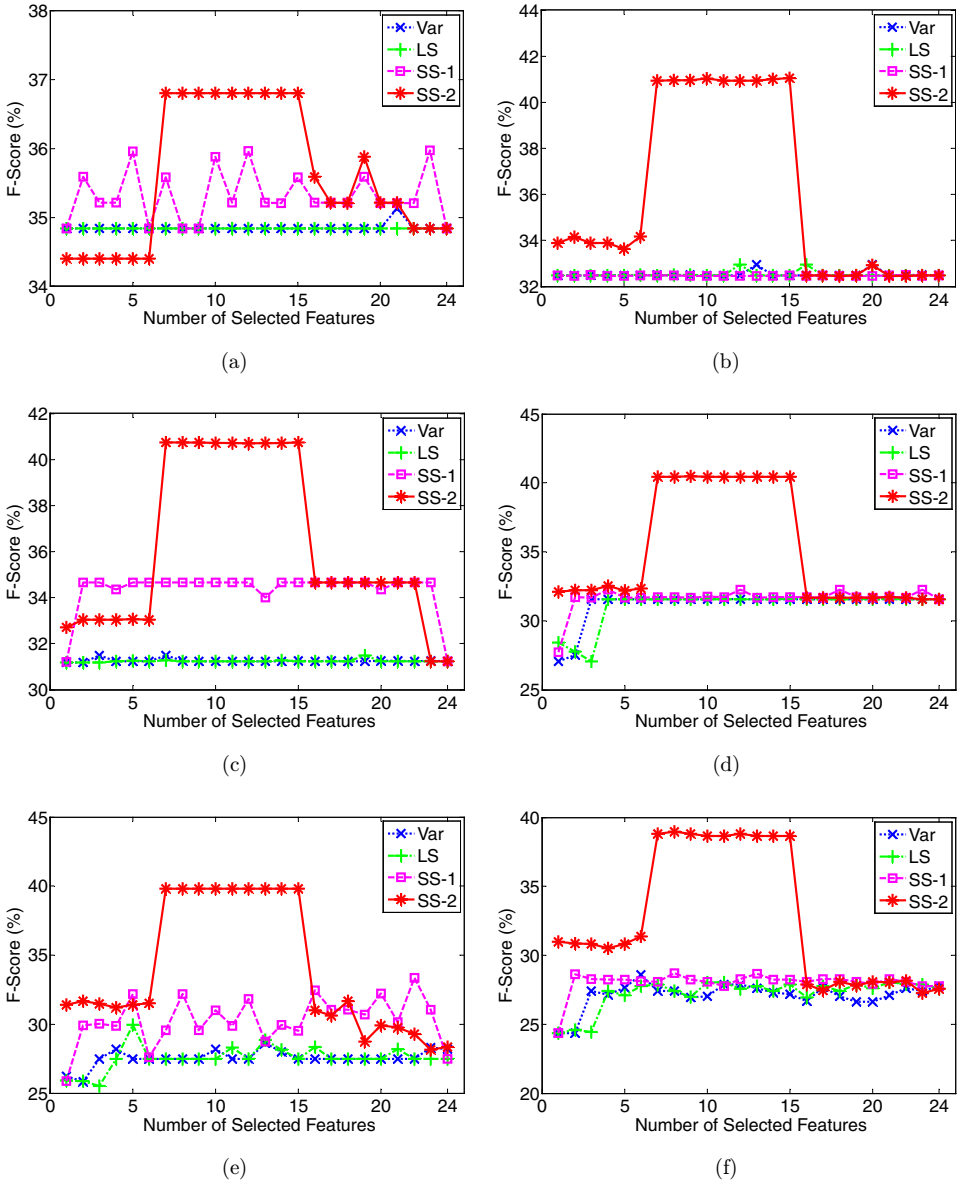
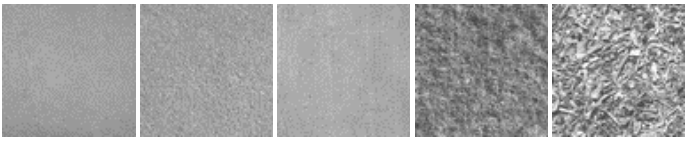
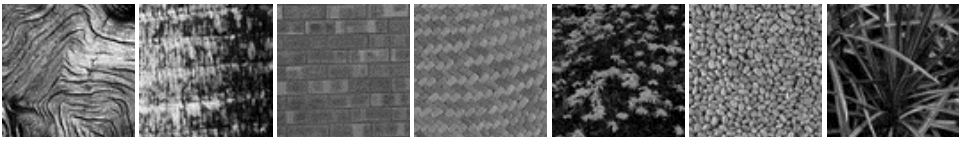


Fig. 2. Clustering results with different cluster numbers on *steel plate faults*. (a) Cluster number = 2, (b) cluster number = 3, (c) cluster number = 4, (d) cluster number = 5, (e) cluster number = 6 and (f) cluster number = 7.

obtain d different classification models. Then, we apply these models to classify samples in the validate set, and get multiple classification results. The number corresponding to the model that achieves the best classification result is set to be the optimal number of selected features. Note that, such validate set is only used to



(a)



(b)

Fig. 3. Example texture images of (a) *MeasTex* and (b) *VisTex* data sets used in our experiments.

Table 5. Texture data sets used in the experiments.

Data Set	#Dimension	#Class	#Sample	Resolution
MeasTex	128	5	640	128×128
VisTex	128	14	1792	64×64

determine the optimal feature subset, and is not used for the training of classification model. Finally, the classification model, which achieves the best result on validate set, is used to classify samples in the test set, and the mean and the standard deviation of classification accuracies are recorded. Meanwhile, classification accuracy on the test set without any feature selection (i.e. using original data) is used as baseline. Moreover, as the optimal number of selected features in 10 folds may be different, we also report the mean and the standard deviation of such numbers. Note that, if there are a relatively small number of samples for specific classes (e.g. the *colon cancer* data set), a five-fold cross-validation strategy is adopted.

5.2.3. *Classification results of unsupervised methods*

First, we perform a series of experiments using four unsupervised filter-type feature selection methods (including Var, LS, SS-1 and SS-2) on UCI and gene expression data sets. Figure 4 plots the curves of classification accuracy versus different numbers of selected features on validate sets. The mean and the standard deviation of the optimal number of selected features are reported in Table 6, where "a" in the term "a ($\pm$b)" is the mean result and "b" is the standard deviation of results.

From Fig. 4, one can see that the proposed SS-1 and SS-2 usually outperform Var and LS, especially when less than half of features are selected. On the high-dimensional gene expression data set (i.e. *prostate cancer*), our proposed SS-1 and SS-2 methods are nearly always superior to the LS and Var. It validates the efficacy of the proposed methods in dealing with small sample size problem.

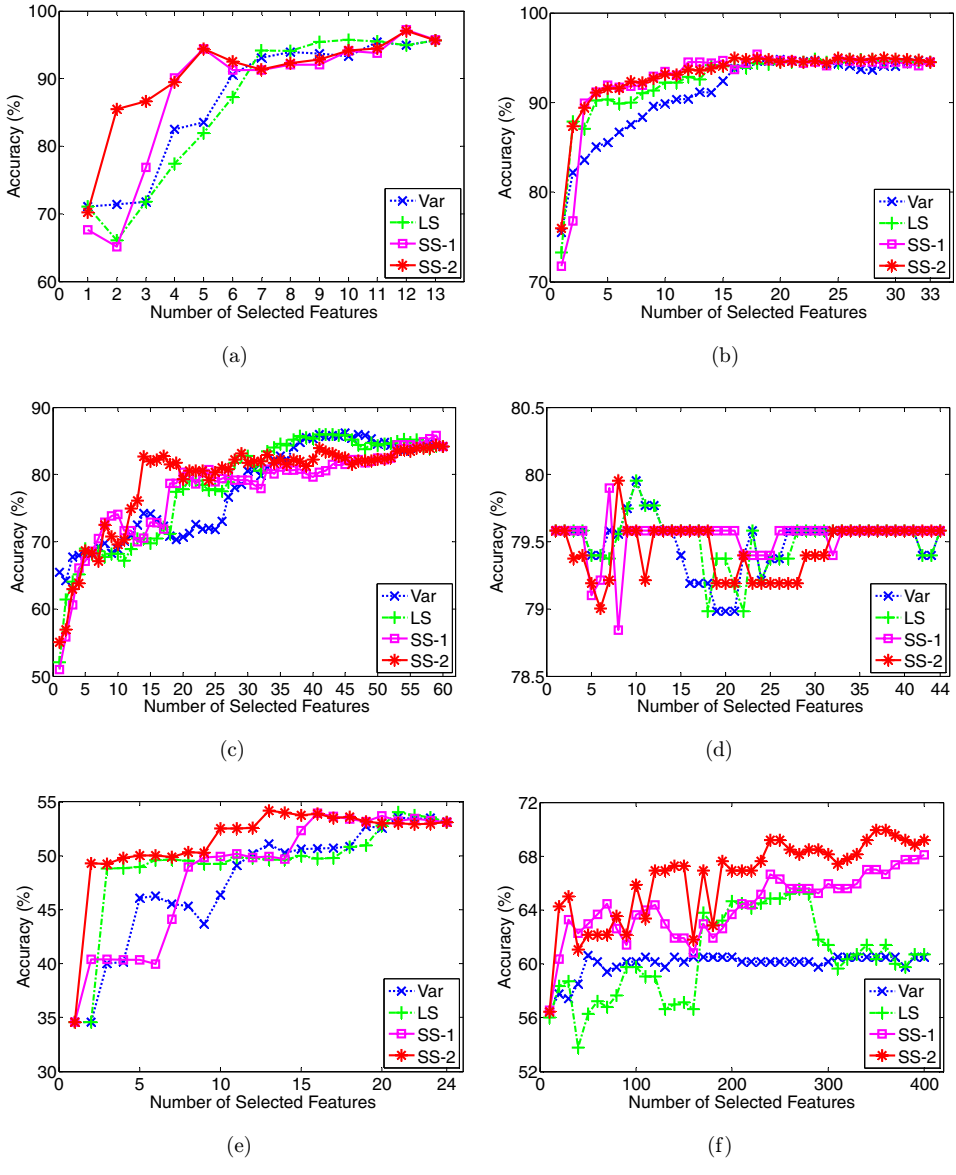


Fig. 4. Classification accuracy versus number of selected features achieved by unsupervised feature selection methods on validate sets. (a) *Wine*, (b) *ionosphere*, (c) *sonar*, (d) *spectf heart disease*, (e) *steel plate faults* and (f) *prostate cancer*.

From Table 6, we can see that these four algorithms do not need all features in order to achieve better performance. Especially on the *colon cancer*, less than 100 features are required to achieve the highest accuracy. One can also find that the proposed SS-1 and SS-2 methods usually use much less features to achieve best results, comparing to Var and LS.

Table 6. Optimal numbers of selected features determined by unsupervised feature selection methods.

Data Set	Var	LS	SS-1	SS-2
Wine	12 (±0.71)	10 (±2.96)	11 (±2.70)	8 (±2.51)
Ionosphere	20 (±8.73)	11 (±9.74)	12 (±7.56)	7 (±3.71)
Sonar	44 (±8.38)	51 (±10.69)	46 (±10.01)	43 (±8.88)
Spectf heart disease	3 (±3.93)	1 (±0)	1 (±0)	2 (±2.23)
Digits 246	13 (±3.83)	12 (±2.07)	15 (±5.35)	12 (±2.96)
Steel plate faults	23 (±1.30)	22 (±1.48)	21 (±2.64)	21 (±0.83)
Colon cancer	60 (±110.10)	100 (±122.40)	6 (±4.08)	28 (±30.80)
Prostate cancer	122 (±59.74)	66 (±65.73)	46 (±63.91)	156 (±56.75)

Table 7. Test classification accuracy of unsupervised feature selection methods (%).

Data Set	Baseline	Var	LS	SS-1	SS-2
Wine	94.96 (±3.01)	95.00 (±3.62)	94.71 (±3.63)	92.71 (±3.12)	95.52 (±2.45)
Ionosphere	76.34 (±3.92)	75.63 (±5.62)	78.63 (±5.62)	80.06(±3.61)	79.77 (±3.78)
Sonar	75.00 (±3.57)	80.78 (±4.01)	80.35 (±4.94)	77.90 (±7.13)	78.35 (±6.53)
Spectf heart disease	79.39 (±0.53)	79.39 (±0.53)	79.39 (±0.53)	79.39 (±0.53)	79.41 (±0.86)
Digits 246	98.88 (±0.77)	99.07 (±1.13)	98.32 (±0.41)	98.39 (±2.82)	99.21 (±1.84)
Steel plate faults	62.71 (±3.26)	64.06 (±1.05)	63.12 (±2.05)	64.16 (±1.39)	63.03 (±1.65)
Colon cancer	68.30 (±7.58)	68.92 (±11.16)	68.92 (±8.09)	70.92 (±6.39)	72.46 (±6.80)
Prostate cancer	66.07 (±10.71)	63.21 (±5.29)	60.23 (±3.72)	69.04 (±3.64)	67.35 (±3.19)

Meanwhile, we report the classification results on the test set using optimal feature subsets in Table 7. From Table 7, one can see that the performance of SS-1 and SS-2 are superior to Var, LS and baseline in most data sets. It is worth noting that on the *colon cancer* and the *prostate cancer* data sets, the proposed SS-1 and SS-2 methods achieve much higher accuracies than those of Var, LS and baseline.

In addition, from Fig. 4 and Table 7, one can see that the performances of SS-1 and SS-2 are inferior to Var and LS on the *sonar* data set. To uncover the underlying reason, we re-investigate the *sonar* data carefully. As shown in Table 2, the *sonar* data have 60-dimensional features. A further observation is that each sample has much larger values on the 34–37th features than those on other features, and thus the 34–37th features may dominate the feature ranking result. Thus, it is important for these features to appear in the ranking list of features. In the experiment, we find that Var and LS always rank the 34–37th features in head positions, while SS-1 and SS-2 rank them in rear positions of ranking lists. That is, Var and LS nearly always select the 34–37th features no matter what the desired number of selected features is, while SS-1 and SS- select them only when the desired number of selected features is near to the number of feature dimension of the data. This partially explains the results shown in Fig. 4 and Table 7.

5.2.4. *Classification results of supervised methods*

In this subsection, we report classification results achieved by different supervised feature selection methods including SuSS (including SuSS-1 and SuSS-2), FS and LFS. The curves of the classification accuracy versus different numbers of selected features on validate sets are plotted in Fig. 5. The mean and the standard deviation of the optimal feature dimension for different algorithms are shown in Table 8.

From Fig. 5 and Table 8, the analogous trend for the proposed SuSS and the other methods can be observed as in Fig. 4 and Table 6. To be specific, the proposed SuSS-1 and SuSS-2 are superior to FS, LFS and RFE in most cases, especially on two high-dimensional data sets. Meanwhile, relatively smaller feature size is required for the proposed SuSS methods to achieve the highest accuracy comparing to other methods. These results further validate the efficacy of the proposed l_1 graph preserving feature selection methods.

Table 9 records the mean as well as the standard deviation of classification accuracies on test sets. From Table 9, we can see that our proposed SS-1 and SS-2 usually have better performances than the other methods on all data sets except for *colon cancer*. We re-investigate this data set and find there are only 62 samples with 2000-dimensional features. In the five-fold cross-validation process, fewer samples are used to construct the l_1 graph where the noise will reduce its quality. Meanwhile, RFE is a wrapper-type feature selection method that is often superior to filter-type ones in terms of accuracy. This partially explains the results shown in Table 9.

On the other hand, from Table 9, we can see that the proposed SuSS-1 and SuSS-2 methods usually perform similarly. It indicates that considering the variance will not necessarily boost the l_1 graph-preserving feature selection method. We re-investigate the situation and find that, similar to variance, the l_1 graph constructed by sparse representation has involved natural discriminate information. The reason is that the nonzero entries in the reconstructive weight vector usually correspond to samples from the same class and therefore may help to distinguish that class from the others. In addition, from Table 9, one can see that our proposed SS and SuSS methods nearly always outperform the other ones on multi-class data sets, i.e. *wine, digits* and *steel plate faults.* This encourages us to apply the proposed methods to texture classification, which is a typical multi-class classification problem.

5.2.5. *Classification results on texture*

Now we apply our proposed SS and SuSS methods to reduce dimension in multi-class texture classification tasks. The first group of experiments is to compare the proposed SS (including SS-1 and SS-2) methods with Var and LS in an unsupervised way. Table 10 reports the mean and the standard deviation of classification accuracies on the test set using the optimal feature subset which is confirmed by the validate set. Note that, the term "a" in "a ($\pm$ b)" is the average accuracy and the term "b" is the standard deviation in the 10-fold cross-validation.

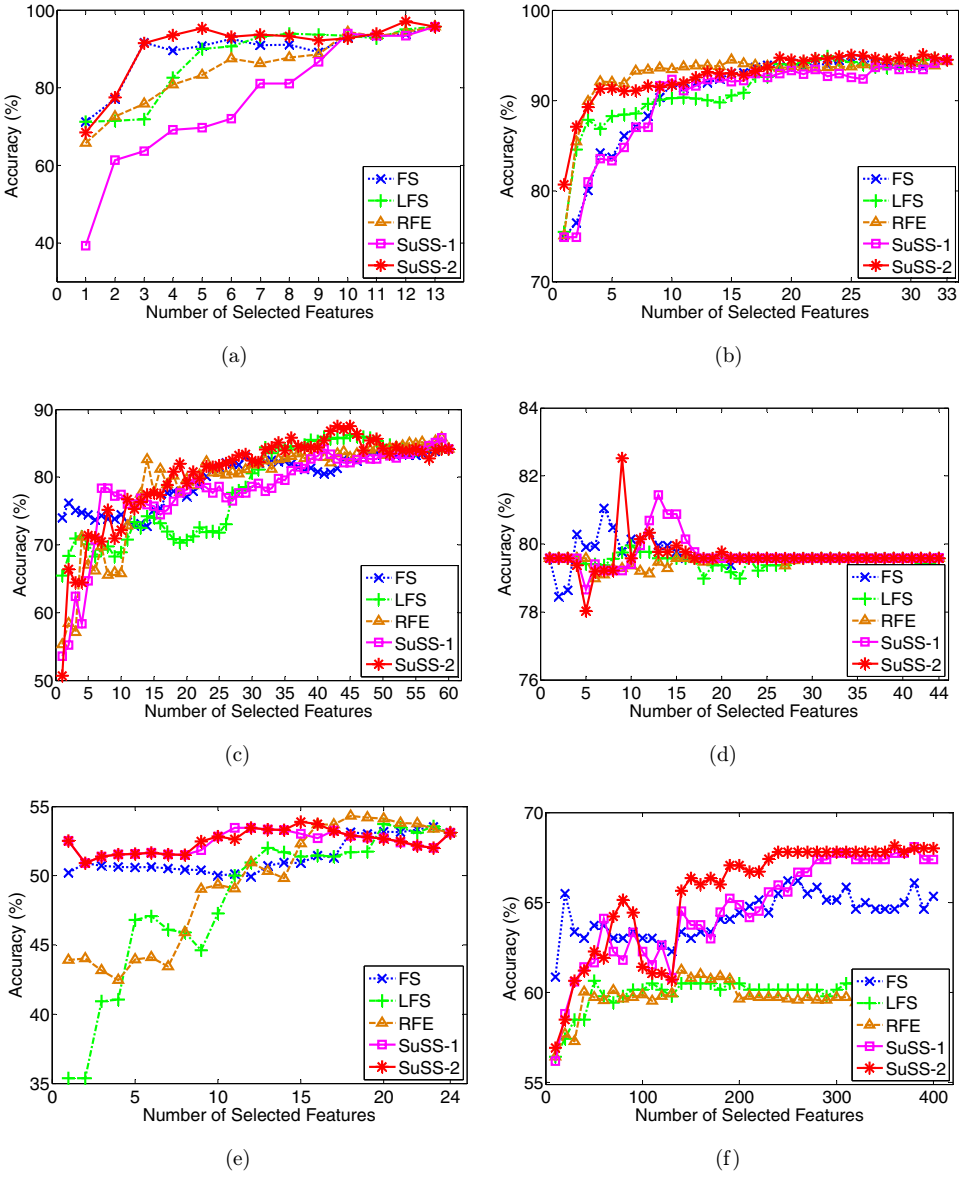


Fig. 5. Classification accuracy versus number of selected features achieved by supervised feature selection methods on validate sets. (a) *Wine*, (b) *ionosphere*, (c) *sonar*, (d) *spectf heart disease*, (e) *steel plate faults* and (f) *prostate cancer*.

From Table 10, one can see that SS-2 outperforms SS-1, Var and LS on two multi-class texture data sets. On the other hand, we find that on the *VisTex* data set, all algorithms achieve much better results than baseline. However, for the *MeasTex* data set, the trend is not so distinct. The underlying reason may be that the

Table 8. Optimal numbers of selected features determined by supervised feature selection methods.

Data Set	FS	LFS	RFE	SuSS-1	SuSS-2
Wine	10 (±4.00)	9 (±3.20)	9 (±3.93)	9 (±1.51)	11 (±2.28)
Ionosphere	6 (±2.30)	8 (±5.70)	8 (±3.84)	9 (±4.72)	5 (±1.30)
Sonar	30 (±13.10)	43 (±11.46)	40 (±13.61)	28 (±15.93)	24 (±12.42)
Spectf heart disease	5 (±2.49)	5 (±4.18)	1 (±0)	1 (±0)	1 (±0)
Digits 246	12 (±0.0)	7 (±2.94)	17 (±5.40)	11 (±7.30)	11 (±2.51)
Steel plate faults	21 (±0.83)	21 (±1.94)	23 (±1.92)	19 (±3.08)	16 (±5.70)
Colon cancer	25 (±32.95)	186 (±163.46)	93 (±145.61)	49 (±9.67)	22 (±13.93)
Prostate cancer	124 (±111.46)	289 (±36.68)	197 (±89.86)	174 (±92.68)	32 (±18.67)

Table 9. Test classification accuracy of supervised feature selection methods (%).

Data Set	Baseline	FS	LFS	RFE	SuSS-1	SuSS-2
Wine	94.96 (±3.01)	92.74 (±5.72)	91.07 (±6.29)	90.45 (±6.06)	96.63 (±1.22)	94.96 (±3.01)
Ionosphere	76.34 (±3.92)	83.19 (±1.85)	79.49 (±2.67)	74.06 (±3.30)	76.34 (±3.92)	85.19 (±1.54)
Sonar	75.00 (±3.57)	74.54 (±7.33)	79.35 (±5.07)	72.69 (±9.08)	80.78 (±2.76)	82.26 (±6.07)
Spectf heart disease	79.39 (±0.53)	79.39 (±0.53)	78.25 (±1.88)	79.39 (±0.53)	79.39 (±0.53)	79.41 (±0.97)
Digits 246	98.88 (±0.77)	98.58 (±0.55)	95.36 (±7.31)	98.21 (±2.28)	99.62 (±0.50)	98.88 (±0.77)
Steel plate faults	62.71 (±3.26)	64.21 (±1.26)	63.03 (±2.09)	63.42 (±1.41)	64.04 (±3.24)	63.71 (±3.26)
Colon cancer	68.30 (±7.58)	65.84 (±7.16)	70.46 (±8.86)	75.53 (±6.25)	69.84 (±8.53)	74.30 (±11.78)
Prostate cancer	66.07 (±10.71)	63.09 (±5.12)	63.92 (±8.41)	65.95 (±7.92)	65.64 (±4.07)	66.07 (±10.71)

contrast of images in *MeasTex* data set is not strong enough as shown in Fig. 3(a). If edges are not clear enough, features extracted by WPT which aiming to capture the edge and orientation information will be not so discriminative. Obviously, it will bring much challenge for feature selection methods to determine the optimal feature subset.

Then, we perform the second group of experiments to compare the proposed SuSS (including SuSS-1 and SuSS-2) method with FS, LFS and RFE in a supervised manner. Experimental results are reported in Table 11. From Table 11, one can find similar trend as in Table 10, i.e. the proposed SuSS-2 method is always superior to SuSS-1, FS, LFS and RFE.

On the other hand, from Tables 10 and 11, one can find that the proposed SS and SuSS methods usually outperform the other methods. It indicates that the feature subset preserving the l_1 graphs (SS and SuSS to preserve) are more compact representation of original data than that of global graphs with equal weights (Var and FS to preserve) and neighborhood graph (LS to preserve). In fact, all existing graph-based feature selection methods have to be faced the problem that the quality of the graph is essential to their performance.

Table 10. Classification accuracy of unsupervised feature selection methods on texture (%).

Data Set	Baseline	Var	LS	SS-1	SS-2
MeasTex	74.41 (±4.12)	75.33 (±3.74)	68.54 (±4.81)	68.62 (±8.04)	75.74 (±4.94)
VisTex	70.65 (±2.08)	84.52 (±2.34)	75.73 (±1.41)	76.19 (±5.28)	85.26 (±3.81)

Table 11. Classification accuracy of supervised feature selection methods on texture (%).

Data Set	Baseline	FS	LFS	RFE	SuSS-1	SuSS-2
MeasTex	74.41 (±4.12)	67.23 (±3.83)	74.41 (±4.12)	68.56 (±6.59)	69.48 (±5.61)	78.10 (±2.04)
VisTex	70.65 (±2.07)	71.61 (±1.51)	82.87 (±2.55)	72.52 (±1.52)	72.52 (±1.52)	84.34 (±1.80)

6. Conclusion

In this paper, we propose a general graph-preserving feature selection framework, and show a number of existing filter-type feature selection methods can be unified into this framework, with different graphs definitions. Moreover, two novel filter-type feature selection methods are proposed based on l_1 graph. Results of both clustering and classification experiments on a number of data sets have validated the efficacy of the proposed methods. Specifically, in clustering experiments, our proposed methods always achieve best performance. In the classification experiments, the proposed methods outperform other algorithms in most cases, especially for multi-class problems.

In the current work, based on the general graph-preserving feature selection framework, we construct l_1 graphs using sparse representation, which may be time-consuming especially for data with large sample size. To design fast algorithms for the construction of l_1 graph can promote the computational efficiency, which is one of our future works. In addition, in this paper, we only adopt l_1 graph for feature selection. In fact, besides l_1 graph, there are other kinds of graphs (e.g. l_2 graph) that can also be used under our general graph-based feature selection framework. It is interesting to investigate whether using other kinds of graphs can also lead to performance improvement, which is also our future work.

Acknowledgments

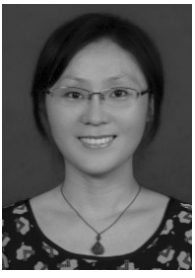
This work is supported in part by the Jiangsu Natural Science Foundation for Distinguished Young Scholar (No. BK20130034), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20123218110009), the NUAA Fundamental Research Funds (No. NE2013105), the Fundamental Research Funds for the Central Universities of China (No. NZ2013306), the Funding of Jiangsu Innovation Program for Graduate Education (No. CXZZ13_0173), and the National Natural Science Foundation of China (No. 61379015).

References

1. U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack and A. J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, *Proc. Natl. Acad. Sci. USA* **96** (1999) 6745–6750.
2. A. Anand, G. Pugalenthi and P. Suganthan, Predicting protein structural class by SVM with class-wise optimized features and decision probabilities, *Pattern Recogn.* **253** (2008) 375–380.
3. K. Bache and M. Lichman, UCI repository of machine learning databases, 2013. Available online at http://archive.ics.uci.edu/ml.
4. R. Baraniuk, A lecture on compressive sensing, *IEEE Signal Process. Mag.* **24** (2007) 118–121.
5. K. Benabdeslem and M. Hindawi, Constrained Laplacian score for semi-supervised feature selection, in *Machine Learning and Knowledge Discovery in Databases*, Vol. 6911 (Springer, Berlin; Heidelberg, 2011), pp. 204–218.
6. C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, Oxford, 1995).
7. A. L. Blum and P. Langley, Selection of relevant features in machine learning, *Artif. Intell.* **97** (1994) 245–271.
8. T. T. Cai and A. R. Zhang, Sparse representation of a polytope and recovery of sparse signals and low-rank matrices, *IEEE Trans. Inf. Theory* **60** (2014) 122–132.
9. D. Cai, X. He and J. Han, Spectral regression for dimensionality reduction, UIUCDCS-R-2007-2856, 2009.
10. C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, *ACM Trans. Intell. Syst. Technol.* **2** (2011) 1–27.
11. S. S. B. Chen, D. L. Donoho and M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM Rev.* **43** (2001) 129–159.
12. Q. A. Cheng, H. B. Zhou and J. Cheng, The Fisher–Markov selector: Fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data, *IEEE Trans. Pattern Anal. Mach. Intell.* **33** (2011) 1217–1233.
13. B. Cheng, J. Yang, S. Yan, Y. Fu and T. S. Huang, Learning with L1-graph for image analysis, *IEEE Trans. Image Process.* **19** (2010) 858–866.
14. C. Cortes and M. Mohri, On transductive regression, in *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2006.
15. D. Dernoncourt, B. Hanczar and J. D. Zucker, Analysis of feature selection stability on high dimension and small sample data, *Comput. Stat. Data Anal.* **71** (2014) 681–693.
16. D. L. Donoho, Compressed sensing, *IEEE Trans. Inf. Theory* **52** (2006) 1289–1306.
17. R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classication*, 2nd edn. (Wiley, New York, 2001).
18. I. Guyon, J. Weston, S. Barnhill and V. Vapnik, Gene selection for cancer classification using support vector machines, *Mach. Learn.* **46** (2002) 389–422.
19. I. Guyon, S. Gunn, M. Nikravesh and L. Zadeh, *Feature Extraction, Foundations and Applications* (Springer, 2006).
20. E. R. Hancock and Z. Zhang, Kernel entropy-based unsupervised spectral feature selection, *Int. J. Pattern Recogn. Artif. Intell.* **26** (2012) 1260002.
21. X. He, D. Cai and P. Niyogi, Laplacian score for feature selection, in *Advances in Neural Information Processing Systems*, Whistler, British Columbia, Canada, 2005.

22. R. He, W. S. Zheng, T. N. Tan and Z. A. Sun, Half-quadratic-based iterative minimization for robust sparse representation, *IEEE Trans. Pattern Anal. Mach. Intell.* **36** (2014) 261–275.
23. K. Huang and S. Aviyente, Sparse representation for signal classification, in *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2006.
24. J. Jaeger, R. Sengupta and W. L. Ruzzo, Improved gene selection for classification of microarrays, in *Pacific Symp. Biocomputing*, Lihue, Hawaii, 2003, pp. 53–64.
25. A. K. Jain, R. P. W. Duin and J. Mao, Statistical pattern recognition: A review, *IEEE Trans. Pattern Anal. Mach. Intell.* **22** (2000) 4–37.
26. M. Kalakech, P. Biela, L. Macaire and D. Hamad, Constraint scores for semi-supervised feature selection: A comparative study, *Pattern Recogn. Lett.* **32** (2011) 656–665.
27. K. Kira and L. Rendell, The feature selection problem: Traditional methods and a new algorithm, in *Proc. 10th National Conf. Artificial Intelligence*, Menlo Park, AAAI Press/ The MIT Press, 1992, pp. 129–134.
28. R. Kohavi and G. H. John, Wrappers for feature subset selection, *Artif. Intell.* **97** (1997) 273–324.
29. N. Kwak and C. H. Choi, Input feature selection by mutual information based on parzen window, *IEEE Trans. Pattern Anal. Mach. Intell.* **24** (2002) 1667–1671.
30. K. Labusch, E. Barth and T. Martinetz, Sparse coding neural gas: Learning of overcomplete data representations, *Neurocomputing* **72** (2009) 1547–1555.
31. Y. Li and B. Lu, Feature selection based on loss-margin of nearest neighbor classification, *Pattern Recogn.* **42** (2009) 1914–1921.
32. J. Y. Liang, F. Wang, C. Y. Dang and Y. H. Qian, A group incremental approach to feature selection applying rough set technique, *IEEE Trans. Knowl. Data Eng.* **26** (2014) 294–308.
33. S. Liapis, N. Alvertos and G. Tziritas, Maximum likelihood texture classification and Bayesian texture segmentation using discrete wavelet frames, in *Int. Conf. Digital Signal Processing Proceedings*, Santorini, 1997, pp. 1107–1110.
34. F. Y. Lin, D. R. Liang, C. C. Yeh and J. C. Huang, Novel feature selection methods to financial distress prediction, *Expert Syst. Appl.* **41** (2014) 2472–2483.
35. H. Liu, J. Sun, L. Liu and H. Zhang, Feature selection with dynamic mutual information, *Pattern Recogn.* **42** (2009) 1330–1339.
36. Z. Lu and Y. Peng, Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications, *Int. J. Comput. Vis.* **103** (2013) 1–20.
37. MIT Media Lab, Vision Texture, available online at http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.
38. S. Nakariyakul and D. P. Casasent, An improvement on floating search algorithms for feature subset selection, *Pattern Recogn.* **42** (2009) 932–940.
39. K. Nigam, A. Mccallum, S. Thrun and T. Mitchell, Text classification from labeled and unlabeled documents using EM, *Mach. Learn.* **39** (2001) 103–134.
40. H. Peng, F. Long and C. Ding, Feature selection based on mutual information: Criteria of max-dependency, max-relevance and min-redundancy, *IEEE Trans. Pattern Anal. Mach. Intell.* **27** (2005) 1226–1238.
41. L. S. Qiao, S. C. Chen and X. Y. Tan, Sparsity preserving projections with applications to face recognition, *Pattern Recogn.* **43** (2010) 331–341.
42. D. Rodrigues, L. A. M. Pereira, R. Y. M. Nakamura, K. A. P. Costa, X. S. Yang, A. N. Souza and J. P. Papa, A wrapper approach for feature selection and optimum-path forest based on bat algorithm, *Expert Syst. Appl.* **41** (2014) 2250–2258.
43. S. T. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* **290** (2000) 2323–2326.

44. C. X. Shang, M. Li, S. Z. Feng, Q. S. Jiang and J. P. Fan, Feature selection via maximizing global information gain for text classification, *Knowl.-Based Syst.* **54** (2013) 298–309.
45. G. Q. Shao, Y. P. Wu, A. Yong, X. Liu and T. D. Guo, Fingerprint compression based on sparse representation, *IEEE Trans. Image Process.* **23** (2014) 489–501.
46. P. Singh, V. Laxmi and M. S. Gaur, Near-optimal geometric feature selection for visual speech recognition, *Int. J. Pattern Recogn. Artif. Intell.* **27** (2013) 1350026.
47. G. Smith and I. Burns, Measuring texture classification algorithms, *Pattern Recogn. Lett.* **39** (1997) 1495–1501.
48. G. Wang, J. Ma and S. L. Yang, An Improved boosting based on feature selection for corporate bankruptcy prediction, *Expert Syst. Appl.* **41** (2014) 2353–2361.
49. A. R. Webb, *Statistical Pattern Recognition* (Cambridge University Press, New York, 1999).
50. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, Robust face recognition via sparse representation, *IEEE Trans. Pattern Anal. Mach. Intell.* **31** (2009) 210–227.
51. Y. Xie, W. S. Zhang, Y. Y. Qu and Y. H. Zhang, Discriminative subspace learning with sparse representation view-based model for robust visual tracking, *Pattern Recogn.* **47** (2014) 1383–1394.
52. S. Yan, D. Xu, B. Zhang, H. J. Zhang, Q. Yang and S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, *IEEE Trans. Pattern Anal. Mach. Intell.* **29** (2007) 40–51.
53. J. Yang, J. Wright, Y. Ma and T. Huang, Image super-resolution as sparse representation of raw image patches, in *IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1–8.
54. H. Yoon, K. Yang and C. Shahabi, Feature subset selection and feature ranking for mutivariate time series, *IEEE Trans. Knowl. Data Eng.* **17** (2005) 1186–1198.
55. L. Yu and H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in *Proc. 20th Int. Conf. Machine Learning*, Washington DC, 2003, pp. 601–608.
56. D. Q. Zhang, S. C. Chen and Z. H. Zhou, Constraint score: A new filter method for feature selection with pairwise constraints, *Pattern Recogn.* **41** (2008) 1440–1451.
57. D. Zhang, Y. Wang, L. Zhou, H. Yuan and D. Shen, Multimodal classification of alzheimer's disease and mild cognitive impairment, *NeuroImage* **33** (2011) 856–867.
58. D. Ziou, T. Hamri and S. Boutemedjet, A hybrid probabilistic framework for content-based image retrieval with feature weighting, *Pattern Recogn.* **42** (2009) 1511–1519.



Mingxia Liu received her B.S. degree and M.S. degree from Shandong Normal University, Shandong, China, in 2003 and 2006, respectively. In 2006, she joined the School of Information Science and Technology at Taishan University as a Lecturer. She is currently a Ph.D. candidate in Computer Science at Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her research interests include machine learning, pattern recognition, computer vision, and image analysis.

Daoqiang Zhang received his B.S. degree and Ph.D. in Computer Science from Nanjing University of Aeronautics and Astronautics (NUAA), China, in 1999, and 2004, respectively. He joined the Department of Computer Science and Engineering of NUAA as a Lecturer in 2004, and is currently a professor. His research interests include machine learning, pattern recognition, data mining, and medical image analysis. In these areas, he has published over 100 scientific articles in refereed international journals such as Neuroimage, Pattern Recognition, Artificial Intelligence in Medicine, IEEE Trans. Neural Networks; and conference proceedings such as IJCAI, AAAI, SDM and ICDM. He was nominated for the National Excellent Doctoral Dissertation Award of China in 2006, won the best paper award at the 9th Pacific Rim International Conference on Artificial Intelligence (PRICAI'06), and was the winner of the best paper award honorable mention of Pattern Recognition Journal 2007. He has served as a program committee member for several international and native conferences such as IJCAI, SDM, CIKM, PAKDD, PRICAI, and ACML, etc. He is a member of the Machine Learning Society of the Chinese Association of Artificial Intelligence (CAAI), and the Artificial Intelligence & Pattern Recognition Society of the China Computer Federation (CCF).