Neurocomputing 93 (2012) 115–125

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Graph embedding based feature selection

Dan Wei [a], Shutao Li [a,*], Mingkui Tan [b]

[a] *College of Electrical and Information Engineering, Hunan University, Changsha 410082, China*
[b] *School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore*

ARTICLE INFO

Article history:
Received 8 September 2011
Received in revised form
31 December 2011
Accepted 29 March 2012
Communicated by J. Kwok
Available online 17 May 2012

Keywords:
Feature selection
Graph embedding
Recursive feature elimination
Manifold learning
Gene selection

ABSTRACT

Usually many real datasets in pattern recognition applications contain a large quantity of noisy and redundant features that are irrelevant to the intrinsic characteristics of the dataset. The irrelevant features may seriously deteriorate the learning performance. Hence feature selection which aims to select the most informative features from the original dataset plays an important role in data mining, image recognition and microarray data analysis. In this paper, we developed a new feature selection technique based on the recently developed graph embedding framework for manifold learning. We first show that the recently developed feature scores such as Linear Discriminant Analysis score and Marginal Fisher Analysis score can be seen as a direct application of the graph preserving criterion. And then, we investigate the negative influence brought by the large noise features and propose two recursive feature elimination (RFE) methods based on feature score and subset level score, respectively, for identifying the optimal feature subset. The experimental results both on toy dataset and real-world dataset verify the effectiveness and efficiency of the proposed methods.

1. Introduction

Many real datasets such as images and microarray data are represented as very high dimensional vectors which bring great challenge in data mining and further processing [1–3]. High dimensionality not only increases the learning cost, but also deteriorates the learning performance, known as the problem of “Curse of dimensionality” [4]. Hence dimensionality reduction has attracted great attentions in pattern recognition and machine learning applications such as computer vision and microarray data analysis. Generally speaking, there are mainly two kinds of dimension reduction techniques, i.e. feature extraction [5,6] and feature selection [7,8], to tackle with the “Curse of dimensionality”. Feature extraction refers to the techniques that map the high dimension data (linearly or nonlinearly) to the lower dimensional subspace under some constraints. And feature selection refers to selecting the most informative features from the original dataset. Feature selection has received great attentions and is being widely used in recent years. One typical application of feature selection is the gene selection in the microarray data analysis. In general, the original microarray data contains thousands of genes (most of them are proved to be redundant) with a small number of samples, which causes the small sample size problem [6] and raises the difficulties in diagnosis. Hence, selecting high discriminative genes (or features) from the rude gene expression data can improve the performance of cancer classification and cut down the cost of medical diagnosis.

Many feature selection methods have been proposed in recent years. These methods can typically be categorized into two groups: wrapper method [9,10] and filter method [11–14]. The wrapper method selects the discriminative features dependently on the classifier used. The wrapper method can be expected to be of high performance, but it is difficult to scale to large datasets owing to the expensive computation cost. The wrapper methods, such as SVM-RFE can be expected of good performance in identifying optimal feature subset [9]. However, they are computationally more expensive compared with filter methods and lack of good generalization capability over classifiers [14]. What's more, if the classifier is not well trained, the performance of the wrapper methods may decline.

The filter method refers to selecting informative features according to their discriminative power without considering any knowledge of the classifier. The filter method possesses the advantages of high speed and capability of dealing with large datasets, but lack of abilities to find the optimal feature subset. Typical filter methods includes T-statistics [12], signal-to-noise ratio method [2] and Fisher score [13]. These methods have shown good performance on linear feature selection but poor performance on nonlinear feature identification owing to that they cannot reveal the mutual information among features. To solve this problem, some new feature scores have been proposed

* Corresponding author.
E-mail addresses: weiweidandan@163.com (D. Wei), shutao_li@yahoo.com.cn (S. Li), tanm0097@ntu.edu.sg (M. Tan).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.03.016

recently based on the graph constructed on the samples, such as Locality Sensitive Discriminant Feature (LSDF) score [1] and Laplacian score [15]. Recently, Nie et al. proposed a subset level (SL) score based method identifying the optimal feature. The SL method can be viewed as a special filter method but shows much better performance than traditional filter methods [14]. By exploring the intrinsic structure of the dataset, we can possibly find more informative features [1,14,15]. Particularly, via the intrinsic graph, some features with complex nonlinear structures can be identified, which is a hard problem for linear feature selection methods such as SVM-RFE. However, their performance may be declined as the noise features increase. Note that, in the traditional graph based feature selection methods, the graph is pre-computed with all features, including both informative and noninformative features. When doing feature selection, one assumes that only a small part of features are informative. Under this scenario, one can hardly build a stable graph when there are relatively large number of noise features. Correspondingly, the performance of the feature selection can no longer be guaranteed. An empirical study of this issue will be presented in Section 3.1.

Regarding the above ambiguity in graph based feature selection, in this paper, we assume that we can obtain a reasonable graph which can relatively describe the relationship among patterns with given features. Considering that with large number of features, the graph can be contaminated by the noise features, we start from all features and recursively build the graph with the remaining features and then remove the non-informative features with respect to the current graph. With this recursive strategy, we proposed two new feature selection methods, namely the feature score based recursive feature elimination method (FS-RFE) and the subset level score based recursive feature elimination method (SL-RFE). Although they are still local, the proposed methods can be expected to have better performance. In summary, the contributions of this paper are: (I) We reveal that the traditional graph based feature selection methods are sensitive to large noises. (II) To avoid the negative influence brought by the noise features to the graph, we proposed an FS-RFE method and an SL-RFE method for identifying the optimal feature subsets. The experimental results verified the performances.

The rest of this paper is organized as follows. A short introduction to the graph embedding framework is given in Section 2. In Section 3, we present a feature score recursive feature elimination method (FS-RFE) and a subset level score recursive feature elimination method (SL-RFE) for feature selection. The experimental results are presented in Section 4. The conclusions are finally discussed in Section 5.

2. Prior knowledge: graph embedding

For a general learning problem, let $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x_n}]$ denote the dataset and $\mathbf{x}_n \in \mathbf{R}^m$ is a sample with m dimensions. The dataset can also be written as $\mathbf{X}=[\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_m]^T$, where $\mathbf{f}_i \in \mathbf{R}^n$ $(i=1,2,\ldots,m)$ are the feature vectors. In supervised learning tasks, a sample $\mathbf{x}_n$ is labeled by class label $c_i \in \{1,2,\ldots,n_c\}$, where n_c is the number of classes. Generally, the dimension m is always very large which increases the difficulties of learning. Yan et al. present a novel unifying framework, named graph embedding, to formulate various feature extraction methods and provide new perspective in designing new methods [6]. In graph embedding framework, an intrinsic graph G and a penalty graph G^p are adopted. Graph $G=\{\mathbf{X},\mathbf{S}\}$ and $G^p=\{\mathbf{X},\mathbf{S}^p\}$ are two undirected weighted graphs with similarity matrix $\mathbf{S}$ and $\mathbf{S}^p$ that can be the adjacency matrix or similarity matrix, depending on different applications. Let $\mathbf{L}=\mathbf{D}-\mathbf{S}$ be the Laplacian matrix of graph G, where $\mathbf{D}$ is a diagonal matrix with entries $D_{ii}=\sum_{i \neq j} S_{ij}$. Similarly we can get the Laplacian matrix $\mathbf{L}^p$ of G^p. The intrinsic graph G denotes the similarity characteristics to be strengthened while the intrinsic graph G^p refers to the similarity characteristics to be suppressed. Simply suppose we project to a one dimensional line, then the graph-preserving criterion of the graph embedding framework is formulated as follows:

$$\mathbf{y}=\arg \min_{\mathbf{y}^T\mathbf{B}\mathbf{y}=\Delta} \sum_{i \neq j} \|\mathbf{y}_i-\mathbf{y}_j\|^2 S_{ij}, \tag{1}$$

where $\mathbf{y}$ is the lower dimensional representation of $\mathbf{X}$ with $\mathbf{y}=[y_1,y_2,\ldots,y_n]^T$. The above projection often appears in classification, where the data is projected to a direction that is perpendicular to the separating hyperplane [16]. By simple algebra calculation, we can get a simpler form with matrix formulations

$$\mathbf{y}=\arg \min_{\mathbf{y}^T\mathbf{B}\mathbf{y}=1} \mathbf{y}^T\mathbf{L}\mathbf{y}, \tag{2}$$

where matrix $\mathbf{B}$ can be the identity matrix $\mathbf{I}$ or the Laplacian matrix of the penalty graph G^p, that is $\mathbf{B}=\mathbf{L}^p$. The constrained minimization problem (1) and (2) can be interpreted as two aspects: on the one hand, for those vertices near to each other, we would like to make them be near in their lower representations, which can be realized by minimizing the objective function of (1) or (2); on the other hand, for those vertices far from each other, we would make them apart as far as possible, which can be realized by maximizing $\mathbf{y}^T\mathbf{B}\mathbf{y}=1$. By taking the two aspects together, it amounts to solve the constrained minimization problem (2) or the constrained maximization problem (3)

$$\mathbf{y}=\arg \max_{\mathbf{y}^T\mathbf{L}\mathbf{y}=1} \mathbf{y}^T\mathbf{B}\mathbf{y}. \tag{3}$$

There are three extensions of the above graph preserving criterion, i.e. linearization, kernelization and tensorization. In this paper, only the linear extension will be considered. In the linear extension, suppose that the high dimension data $\mathbf{X}$ will be linearly mapped to a lower dimensional subspace by linear projection $\mathbf{y}=\mathbf{w}^T\mathbf{X}$, where $\mathbf{y} \in \mathbf{R}^d$. Then the optimal projection direction $\mathbf{w}$ can be obtained by solving the following constrained maximization problem:

$$\mathbf{w}^*=\arg \max_{\mathbf{w}^T\mathbf{X}LX^T\mathbf{w}=1} \mathbf{w}^T\mathbf{X}BX^T\mathbf{w}. \tag{4}$$

The above constrained maximization problem can be reformulated as a general Rayleigh quotient problem [17]:

$$\mathbf{w}^*=\arg \max_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{X}BX^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}LX^T\mathbf{w}}. \tag{5}$$

Most of the linear feature extraction methods, such as Linear Discriminant Analysis (LDA) [12], MFA [6] can be formulated within the graph embedding framework. The only difference among them just lies in the different definitions of the intrinsic graph G and the corresponding penalty graph G^p. Here we only present the graph definitions of LDA and MFA. LDA searches for the projections that minimize the intra-class scatter and at the same time maximize the inter-class scatter, which is equivalent to the problem (5) by defining the intrinsic graph and the penalty graph as

$$\begin{cases} S_{ij}=\delta_{c_i,c_j}/n_{c_i}, & i \neq j, \\ S_{ij}^p=1/N-S_{ij}, & i \neq j, \end{cases} \tag{6}$$

where $\delta_{c_i,c_j}=1$, if $c_i=c_j$, otherwise $\delta_{c_i,c_j}=0$. Obviously in the intrinsic graph of LDA, all the data points in the same class are interconnected with weight S_{ij}, while in the penalty graph the data points from different classes are interconnected with weight S_{ij}^p. Therefore, LDA fails to discover the local geometrical structure of the data manifold [6] and therefore can not deal with nonlinear problems. To preserve the local structure of the original data in the

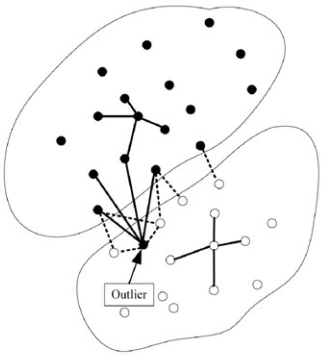

Fig. 1. Intrinsic graph and penalty graph of MFA.

subspace, MFA finds the projections which maximize the margin by discovering the local manifold structures. The intrinsic graph and the penalty graph of MFA are respectively defined as Eqs. (7) and (8).

$$S_{ij} = \begin{cases} 1 & \text{if } i \in N_{k_1}(j) \text{ or } j \in N_{k_1}(i), \\ 0 & \text{else,} \end{cases} \tag{7}$$

where $N_{k_1}(i)$ denotes the index set of the nearest neighbors of data i that are in the same class.

$$S_{ij}^{P} = \begin{cases} 1 & \text{if } (i,j) \in P_{k_2}(c_i,c_j)\ c_i \neq c_j, \\ 0 & \text{else,} \end{cases} \tag{8}$$

where $P_{k_2}(c_i,c_j)$ is a set of data pairs that are the k_2 nearest pairs among the class c_i and c_j. In two class problems, the data points indexed by $P_{k_2}(c_i,c_j)$ are to some extent like the support vectors in SVM classifiers, which are commonly believed to be more important for classification [18]. Compared with LDA and SVM, the major advantage of MFA is that it has a special strength for mining the nonlinear structures of the original dataset. The intrinsic graph and the penalty graph are illustrated in Fig. 1, where the dashed line denotes the edges of the penalty graph and the solid line denotes the edges of the intrinsic graph.

When calculating the graph, to avoid the attributes in greater numeric ranges dominate those in smaller numeric ranges, the feature vectors should be scaled in the data preparation. For example, the feature vector $\mathbf{f}$ can be firstly centered and then scaled so that $\|\mathbf{f}\| = 1$. Let $\mathbf{1} = [1,1,\ldots,1]^T$, the feature vector can be preprocessed according to

$$\mathbf{f} = \frac{\mathbf{f} - \mathbf{f}^{\mathbf{T}}\mathbf{1}}{\|\mathbf{f} - \mathbf{f}^{\mathbf{T}}\mathbf{1}\|}. \tag{9}$$

3. Graph embedding based feature selection

3.1. Feature score for measuring feature importance

Over the past years, many indices have been proposed to measure the importance of a feature, such as t-test score [12], Fisher score (also know as LDA score) [13], and LSDF score [1]. In the wrapper methods, the importance of a feature is usually measured according to its contribution to induced classifiers. For example, in SVM-RFE, the features are ranked according to their contribution to the margin, and the less important features are recursively removed from the feature list.

Intuitively, we say that a feature is important than another one if the former can better represent the intrinsic structure of the original data. The intrinsic structure here can be explained as the sub-manifold structures of the dataset in supervised cases or the manifold structure in unsupervised cases. To be specific, in graph embedding, the importance of a feature can be measured by the degree which respects the graph structure [15]. Recalling the graph preserving criterion in the linear extension, we should solve the maximization problem of (5), which can be reformulated as

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \frac{\sum\sum w_i w_j \mathbf{f}_i^T \mathbf{B} \mathbf{f}_j}{\sum\sum w_i w_j \mathbf{f}_i^T \mathbf{L} f_j}. \tag{10}$$

To compute the contribution of a feature $\mathbf{f}_j$ to the above problem, we can simply set the weights of other features $w_i (i \neq j)$ to 0, which is to some extent like the sensitivity analysis. Then a feature score can be calculated as follows [15]:

$$\theta_G(j) = \frac{\mathbf{f}_j^T \mathbf{B} \mathbf{f}_j}{\mathbf{f}_j^T \mathbf{L} f_j}. \tag{11}$$

Back to the definition of the graph preserving criterion, the feature score is equivalent to projecting the data to the axis and searching for the axis where the generalized Rayleigh quotient value is maximized as the most informative feature.

From Ref. [15], the θ_G-score is very effective for measuring the importance of the linear features. However, when dealing with the nonlinear feature selection problem, it is sensitive to the noise, which increases the difficulties of the nonlinear feature identification. The reasons are analytically illustrated as follows. In the manifold learning, the local structure which is usually characterized by the K-NN nearest neighborhood graph is very important to the learning performance. And the matrices **B** and **L** are usually the point-to-point distance based matrices. Then suppose a dataset with m_0 nonlinear informative features (indexed by an index set I_0) and m_N noise features. If $m_N \to \infty$, then $\mathbf{B} \to \mathbf{B}^{(-I_0)}$ and $\mathbf{L} \to \mathbf{L}^{(-I_0)}$, where $\mathbf{B}^{(-I_0)}$ and $\mathbf{L}^{(-I_0)}$ are the matrices calculated without the informative features. Then the θ_G-score reduces to the θ'_G-score.

$$\theta'_G(j) \approx \frac{\mathbf{f}_j^T \mathbf{B}^{(-I_0)} \mathbf{f}_j}{\mathbf{f}_j^T \mathbf{L}^{(-I_0)} \mathbf{f}_j}. \tag{12}$$

Obviously, the nonlinear structure of the original dataset will lose when the number of noise features is relatively far more than the informative features, i.e. $m_0 \ll m_N$, which is a common problem in pattern recognition applications especially in microarray datasets. Therefore, the nonlinear structure of the dataset may be hidden when the number of the noise features is relatively too large. However, the scores of linear features will be little influenced by the noise. The reason is that the local structure of the linear dataset is no longer as important as in the nonlinear data. In fact, the linear separability can be well described just using the label information. Obviously, the LDA score is noise independent. That is to say, although $\mathbf{B}^{(-I_0)}$ and $\mathbf{L}^{(-I_0)}$ may lose the nonlinear information of the dataset, they are good enough to describe the linear separability.

To further illustrate this issue, an experiment on two toy datasets is performed. Each dataset contains two informative features, with which the first dataset is linearly separable while the latter one is nonlinearly separable, as respectively shown in Fig. 2(a) and (b). We measure the difference between the scores of the informative features with the scores of the noises by score ratio value (SRV)

$$SRV = \log\left[\frac{m - |I_0| \sum_{i \in I_0} \theta_G(i)}{|I_0| \sum_{i \subseteqq / I_0} \theta_G(i)}\right], \tag{13}$$

where $|I_0|$ denotes the number of features in the feature subset index I_0. The larger the SRV value is, the larger the difference between the feature subset and the remained features will be. If the mean of the informative feature scores is equal to the mean of the noise feature scores, then the SRV=0. Fig. 3 shows the SRV

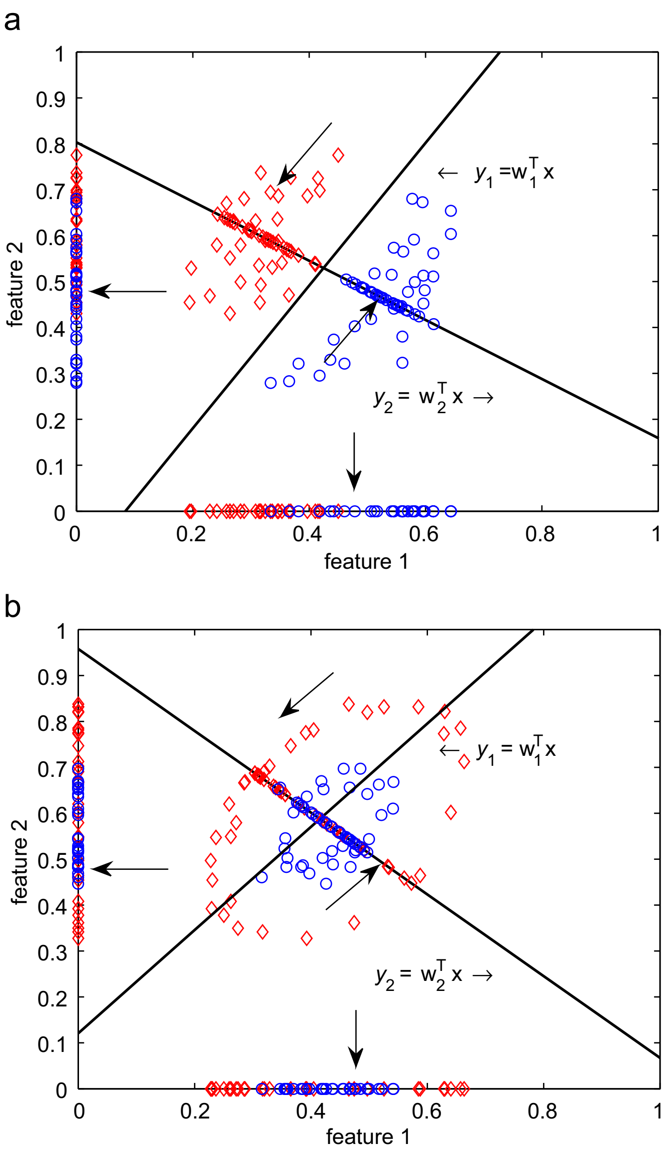

Fig. 2. Toy dataset for two class problem. (a) Linear case. (b) Nonlinear case.

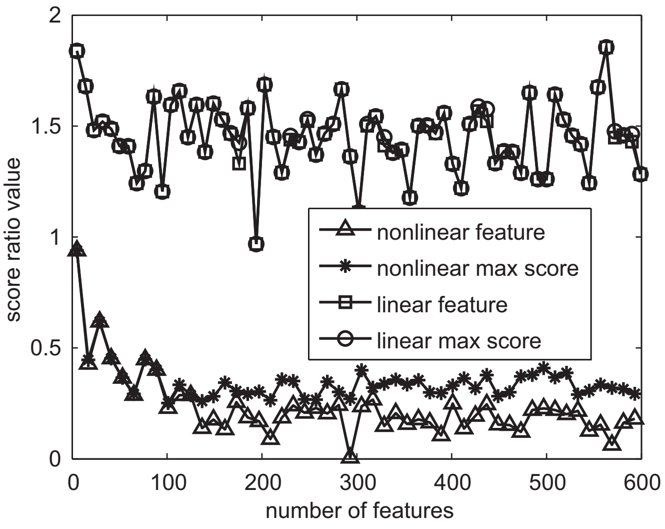

Fig. 3. Score ratio value with different numbers of features.

values of the two toy datasets with respect to the number of features with noises. The SRV values of the two features with the largest scores are also recorded. For the linear features, although SRV value varies with the increase of the features with noises, it is always larger than 1 even with large number of noise features. However, for the nonlinear dataset, the SRV will sharply decrease with the increase of the features with noises. Then we cannot select the right informative features just according to their scores.

Although a single feature cannot fully describe the manifold structure of the original dataset, to some extent it can represent some information of the whole dataset. To overcome the negative influence of the noise features, following the graph preserving criteria, a new single feature score, called θ_S-score, can be defined as follows:

$$\theta_S(j) = \frac{\mathbf{f}_j^T \mathbf{B}^{(j)} \mathbf{f}_j}{\mathbf{f}_j^T \mathbf{L}^{(j)} \mathbf{f}_j}, \tag{14}$$

where matrices $\mathbf{B}^{(j)}$ and $\mathbf{L}^{(j)}$ are constructed using $\mathbf{B}$ and $\mathbf{L}$ corresponding to the sole feature $\mathbf{f}_j$. Apparently, the θ_S-score is noise independent.

Fig. 4 compares the plot figures of the θ_G-MFA score and θ_S-MFA score on a nonlinear toy problem. In the toy dataset, there are 600 features in total and only two of them are nonlinear informative features described in Fig. 2(b). To facilitate display, we place the two informative features on position 200 and position 400. The 598 noises are generated as follows: the first 199 and last 200 noise features are uniform distributed noises generated by $\mathbf{f} = 4+6 \times rand(n,1)$. The middle 199 noise features are norm distributed noises generated by $\mathbf{f} = 4+6 \times randn(n,1)$. Obviously, the θ_G-score cannot correctly identify the two informative features. However, with the θ_S-score, the two informative features can be accurately identified.

3.2. Optimal feature subset selection

Identifying the most informative feature subset is an important but challenging problem. The optimal feature subset identification can be seen as the following combinatorial optimization problem:

Given a feature set $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$, find a gene subset $\mathcal{F}^* \subseteq \mathcal{F}$ that maximize an subset level score objective function $F : \Pi \rightarrow R$ that

$$\mathcal{F}^* = \arg\max_{\mathcal{F}_1 \in \Pi} F(\mathcal{F}_1), \tag{15}$$

where Π is the space of all possible feature subsets of $\mathcal{F}$, and $\mathcal{F}_1$ is a subset of Π. An optimal subset of feature is not necessarily unique and the feature selection problem is an NP-hard problem [19]. And it is very hard to define a reasonable subset score function, which, apparently, has a profound influence on the final feature subset. In the filter methods, the scores for each feature are calculated, and the leading features are selected to construct the optimal feature subset. Different from evaluating a single feature in the filter method, Nie et al. proposed a subset level score to measure the importance of a subset [14],

$$\mathcal{F}^* = \arg \max_{\mathcal{F}' \subset \mathbf{F}, |\mathcal{F}'| = m_0} \frac{\sum_{\mathbf{f}_i \in \mathcal{F}'} \mathbf{f}_i^T \mathbf{B} \mathbf{f}_i}{\sum_{\mathbf{f}_i \in \mathcal{F}'} \mathbf{f}_i^T \mathbf{L} f_i} = \arg \max_{\mathbf{F}' = \mathbf{W}^T \mathbf{x}} \frac{\operatorname{tr}(\mathbf{F}' \mathbf{B} \mathbf{F}'^T)}{\operatorname{tr}(\mathbf{F}' \mathbf{L} \mathbf{F}'^T)}, \tag{16}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{m_0}]$ is an $m \times m_0$ zero matrix except for the positions $(I(j), j)$ with 1, $I(j)$ is the index of the j th selected feature with $j = 1, 2, \ldots, m_0$. The optimal feature subset can be identified such that the subset level score is maximized. Problem (16) is solved by an iterative algorithm which is implemented as Algorithm 1 [14].

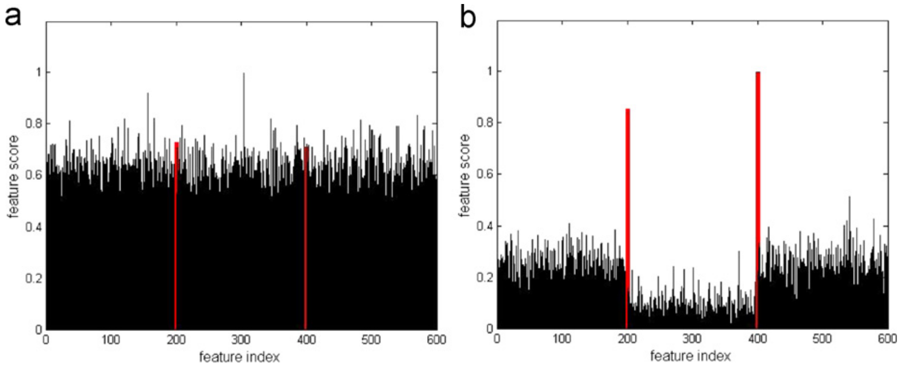

Fig. 4. Toy problem with noises. (a) θ_G-MFA score. (b) θ_S-MFA score.

Algorithm 1. Subset level score method for feature selection

```
0: Input: The training dataset $\mathbf{X}$, the corresponding label $\mathbf{Y}$ (if applicable), convergence tolerance $\varepsilon$ and the feature numbers $m_0$ of the final subset.
1: Construct the matrix $\mathbf{B}$ and $\mathbf{L}$, and randomly initialize a feature index set $\mathcal{F}' \subset \mathcal{F}$, such that $|\mathcal{F}'| = m_0$. Set iteration index $k=0$.
2: Calculate the trace ratio value $\lambda_k = \frac{\sum_{\mathbf{f}_i \in \mathcal{F}'} \mathbf{f}_i^T \mathbf{B} \mathbf{f}_i}{\sum_{\mathbf{f}_i \in \mathcal{F}'} \mathbf{f}_i^T \mathbf{L} f_i}$.
3: For each feature, calculate its feature score according to $\theta_l(j) = \mathbf{f}_j^T \mathbf{B} \mathbf{f}_j - \lambda_k \mathbf{f}_j^T \mathbf{L} f_j$.
4: Rank the feature score in descending order. Update the index vector $\mathbf{I}$ with the indices of the top $m_0$ features.
5: Stop if $|\lambda_k - \lambda_{k-1}| < \varepsilon$, else go to Step 2.
   Output: The index of the selected features $\mathbf{I}$.
```

The subset level score method shows good performance when the number of noise features is small. However, it is also very sensitive to large number of noise features when identifying the nonlinear features. The reasons are similar to those discussed for the θ_G-score. In the subset level score method, the matrices **B** and **L** are computed using all the features. As mentioned previously, if the number of the noise features is too large, **B** and **L** will lose the nonlinear information, leading to the bias when measuring the importance of features by $\theta_l(j) = \mathbf{f}_j^T \mathbf{B} \mathbf{f}_j - \lambda_k \mathbf{f}_j^T \mathbf{L} f_j$ [14]. So the key to improve the subset level score method is also to reduce the negative influences brought by the noise or irrelevant features.

3.3. *Recursive feature elimination for optimal feature subset identification*

Reducing the neglect influences brought by the noisy or irrelevant features is very important to improve performance of graph based feature selection. Note that, in the case of large number of noise features, although the graph may not stable and accurate, it is a reasonable graph to the original dataset and the corresponding feature score, though loss of accuracy, it can relatively represents the importance of features. Based on this assumption, the features with the least scores can be considered as the least important. Then we can remove these features as noises. We can repeat this procedure until the number of features is less than we need. This strategy is known as the recursive feature elimination (RFE) method. A typical application of RFE is the SVM-RFE method which is proposed by Guyon et al. to identify the most informative genes from the gene microarray data [9]. The key idea of SVM-RFE is that the influence of the noise features or irreverent features can be eliminated step by step while the useful information can be gradually condensed. In SVM-RFE, the noise features with less contribution to the margin are eliminated recursively from the gene list and the features that contribute to the classifier most are kept. The SVM-RFE shows good performance on linear gene subset identification. However, it always suffers from the problem of the outliers and is hard to extend to nonlinear problems. As previously presented, the intrinsic matrixes **B** and **L** play a key role in the graph based filter methods and the subset level score method. However, the intrinsic structure presented in **B** and **L** will be gradually weakened with the increase in the noise features. Therefore, with an RFE scheme we can gradually refine the intrinsic graph and hence improve the performance. In this paper, two types of RFE methods based on graph embedding have been proposed. In the first method, we can recursively remove the features with the least feature scores and re-compute the **B** and **L** for each step. The feature score based recursive feature elimination (FS-RFE) method is described in Algorithm 2.

Algorithm 2. Feature score based recursive feature elimination

```
0: Input The training dataset $\mathbf{X}$, the corresponding label $\mathbf{Y}$ (if applicable), the feature numbers $m_0$ of the final subset and the removed feature number $m_r$ at one step.
1: Initialize the feature index list $\mathbf{s} = [1 : m]$, and the removed feature index list $\mathbf{r} = []$. Let $length(\mathbf{s})$ be the length of the vector $\mathbf{s}$.
2: Construct the matrix $\mathbf{B}$ and $\mathbf{L}$ with $\mathbf{X} = \mathbf{X}(: ,\mathbf{s})$.
3: For each feature, calculate its feature score according to the feature score defined in (11).
4: Rank the feature score in ascending order and identify the indices of the top $m_r$ features $\mathbf{s}_r$. Add the top $m_r$ features to the removed feature index list by $\mathbf{r} = [\mathbf{s}(1,\mathbf{s}_r),\mathbf{r}]$ and remove them from the top $m_r$ features from the feature list by $\mathbf{s}(1,\mathbf{s}_r) = []$.
5: Stop if $length(\mathbf{s}) \le m_0$, else go to Step 2.
   Output: The index of the selected features $\mathbf{s}$.
```

Taking the computation cost into consideration, in the FS-RFE method, although several feature scores can be adopted for measuring the importance of the features, the θ_G-score will be a better choice for its simplicity in computation. And θ_G-score can ensure good performance if we remove a small enough number of features in each step. With the RFE method, we obtain a series of nested feature subsets $F_1 \subset F_2 \subset \cdots \subset F$.

One problem of the FS-RFE method is that if the dataset contains linear features, the nonlinear structure will be totally suppressed. In general, the scores of the linear features are larger than the nonlinear features. Therefore, the selected m_0 features tend to be linear features. More importantly, the nonlinear features tend to be removed in the earlier steps. As discussed previously, the linear feature scores are little affected by the noises and usually much larger than the noises and the nonlinear features. Hence, we can remove those features with large enough SRV values (for example SRV > 1) in each step. Then the removed linear features as well as the features in $\mathbf{s}$ formulate the final feature subset. However, we should reorder the sub subset by a new RFE procedure or the subset level score method.

In the second RFE method, we extend the subset level score method for RFE feature selection. In the original subset level score method, an optimal feature subset with predefined size is confirmed with several iterations. Then in the RFE method, we iteratively redo the subset level score method with sequentially decreased number of features. Then the noise features will be recursively removed from the feature list. The subset level score based recursive feature elimination (SL-RFE) method is shown in Algorithm 3.

Algorithm 3. Subset level score based recursive feature elimination

```
0: **Input**: The training dataset $\mathbf{X}$, the corresponding label $\mathbf{Y}$
   (if applicable), the feature numbers $m_0$ of the final subset
   and the removed feature number $m_r$ at one step.
1: Initialize the feature index list $\mathbf{s} = [1 : m]$, and the removed
   feature index list $\mathbf{r} = []$. Let $length(\mathbf{s})$ be the length of the
   vector $\mathbf{s}$.
2: Construct the matrix $\mathbf{B}$ and $\mathbf{L}$ with $\mathbf{X} = X(: ,s)$.
3: Remove $m_r$ features by subset level score method and
   identify the indices of the removed features $s_r$. Add the top
   $m_r$ features to the removed feature index list by
   $\mathbf{r} = [\mathbf{s}(1,\mathbf{s}_r),\mathbf{r}]$ and remove them from the top $m_r$ features
   from the feature list by $\mathbf{s}(1,\mathbf{s}_r) = []$.
4: If $length(\mathbf{s}) \le m_r + m_0$, $m_r = length(\mathbf{s}) - m_0$.
5: Stop if $length(\mathbf{s}) \le m_0$, else go to Step 2.
   **Output** The index of the selected features $\mathbf{s}$.
```

Note that in the linear extension graph preserving criteria, we need to search for a projection direction $\mathbf{w}$ that maximizes the problem (5), obtaining a hyperplane $y = \mathbf{w}^T\mathbf{X}$ that best separates the datasets under the graph embedding criteria. At the same time, $\mathbf{w}$ can be seen as a weight vector to the attributes and the features with small absolute weights or apparently are not important to the hyperplane. We can remove the features with small w_i^2 recursively as the SVM-RFE does [9]. In RFE methods, the number of the removed features at each step is important to the performance. A small m_r always leads to good performance on identifying the useful features but needs more computation costs. In the original SVM-RFE method, the features are eliminated one by one. In our experiments, we use a constant m_r, which will be discussed in the experiments.

3.4. Complexity analysis

In this section, we provide an analysis of the computational complexity of the mentioned feature selection methods to the number of samples and the number of features. Generally speaking, the RFE method is more complex in computation than filter method. Although the intrinsic graph and the penalty graph are needed to be calculated in the graph embeddings, only one of them is taken into consideration since they are equivalent to each other.

In θ_G-score computation, for a given point $\mathbf{x}$, computing the distances needs $O(nm)$ calculations and sorting to get its k-NNs takes $O(kn)$. Thus, for all n points, finding k-NNs is $O(n^2(m+k))$. The computation of $\mathbf{f}^T\mathbf{M}\mathbf{f}$, where $\mathbf{M}$ is an $n \times n$ matrix, takes $O(n^2m)$ for all features. Therefore, it takes $O(n^2(2m+k))$ for computation of θ_G-score for all features. If $m > k$, it will take $O(n^2m)$.

In θ_S-score, the k-NN graph should be calculated for each feature. Thus, for all n points and m features, finding k-NNs needs $O(n^2m(1+k))$. Besides, it will take $O(n^2m)$ for the computation of $\mathbf{f}^T\mathbf{M}\mathbf{f}$, where $\mathbf{M}$ is an $n \times n$ matrix. Therefore, the time complexity of θ_S-score is $O(n^2m(2+k))$. If $m > k$, it will take $O(n^2m)$.

For the subset level score method, the most computational part lies in the computing of all $\mathbf{f}_i{}'\mathbf{B}\mathbf{f}_i$, which will take $O(n^2m)$. It will take $O(n^2(m+k))$ for finding k-NNs graph. Therefore, it will take $O(n^2m)$ in all.

For FS-RFE method, suppose in FS-RFE method, a fixed number of m_r features will be filtered out in each filter step and the whole algorithm will be terminated when the number of the remaining features is less than m_r. Then it will take at most k_2 steps to finish the whole procedure, where $k_2 = [m/m_r]$. Suppose $m > k$, then in the ith iteration, it will take $O(2n^2(m-i \times m_r))$. To sum it up, it will take $O(2k_2mn^2 - m_rk_2(k_2-1)n^2)$ in all for FS-RFE method. Let m_r=1, then $k_2 = m$ and it will take $O(m^2n^2+mn^2)$ in total. That is to say, the FS-RFE method at most takes $O(m^2n^2+mn^2)$. The same complexity analysis can be applied in the SL-RFE method. Typically, if we set m_r=1. it will take at most $O(n^2m^2)$. Although the graph based feature selection methods takes $O(n^2)$, the computational cost can be much reduced by constructing an approximate graph. Furthermore, because both $\mathbf{B}$ and $\mathbf{L}$ are sparse matrix, the cost of matrix vector multiplication can be much reduced. Finally, because the graph have relatively stable structures if we remove relatively large number of features, we can set a large m_r to speed up the feature selections.

For the SVM-RFE method, its computational complexity largely depends on the number of feature eliminated in each step. Assume that linear SVM training takes $O(nm)$ time. Therefore, if one feature is removed from the feature list in each elimination step, the SVM-RFE will take $O(nm^2)$ time.

4. Experiments

4.1. Data preparation and performance evaluations

To evaluate the performance of the proposed methods, several datasets including toy dataset and real-world dataset are adopted to evaluate various methods mentioned in this paper. The toy experiments will be described in next section. In the real-world experiments, eight datasets are used for verification. A brief description of these datasets is summarized in Table 1.

In the toy experiment, we just gradually increase the noises to the dataset and test whether the feature selection method can mine the predefined features. In the real-world experiments, the classification accuracies with selected features on testing data are measured. In this paper, we use two well known classifiers as baseline classification methods. Different from the Fisher score or SVM-RFE, the proposed method in this paper apparently can identify the nonlinear informative features. So the linear

Table 1
General information of the real-world dataset.

Dataset	Samples	Features	Classes
Australian [20]	690	14	2
German [20]	1000	20	2
Vehicle [20]	846	18	4
USPS [20]	7291	256	10
UMIST [21]	575	644	20
Yalefaces [22]	165	2500	15
Leukemia [20]	72	7129	2

classifiers are not appropriate for classification. As to the nonlinear classifiers, there are usually some parameters to be tuned. All things considered, in this paper, we adopt the SVM classifier with Gaussian kernel and K-NN for two class problems. As to the multi-class problem, only K-NN classifier is adopted for its simplicity in use. For the purpose of facilitating the comparison, for the former two datasets, we randomly select 60 samples per class for training and the remaining samples for testing, as suggested in [14]. The average accuracy rates versus selected feature number are recorded over 20 random splits. For the other datasets, except for specification, we use a leave-one-out cross-validation scheme to do the experiments. And the average prediction accuracy are recorded over each trial.

Model selection is a crucial problem in the pattern recognition algorithms. In some situations, the learning performance may drastically vary with different choices of the parameters [23]. In this paper, we have referred several learning algorithms, such as LDA, MFA, LSDF, SVM and K-NN classifiers. Among these methods, LDA is non-parametric method. In the MFA method, there are two parameters, i.e. the number of the inter-class nearest neighborhood k_1 and the number point pairs k_2 of between classes with the least distances. As suggest in [6], k_1 is usually set to 4. The setting of the parameter k_2 is complicated. Because the data points confirmed by the k_2 pairs in MFA is a bit like the support vectors in SVM, we can set k_2 according to the studies of SVM. As commonly believed that the less the support vectors, the better the generalization ability the SVM is [24]. In our experiments, we set k_2 according to the following equation:

$$k_2 = \frac{1}{10}\left[\frac{n(n-1)}{2} - \sum_{i}^{c}\frac{n_c(n_c-1)}{2}\right], \tag{17}$$

where c is the number of classes, n_c denotes the number of samples in class c. Eq. (17) indicates that only 1/10 of the between class pairs with least distances are connected in the penalty graph. In LSDF, the number of the inter-class nearest neighborhood k_1 is also set to 4. As to SVM, there are two parameters. The variance of Gaussian kernel is set to 0.5, and the cost of the constrain violation C is set to 100. In K-NN classifier, we set $K=1$ for all the databsets. In addition, the RFE method proposed in this paper contains an additional parameter m_r. We empirically set $m_r=10$ in all RFE methods. In SL method, SL-RFE, and FS-RFE, the intrinsic graph and penalty graph are constructed using the MFA graph definitions. The SVM classifier in SVM-RFE is implemented via the liblinear solver.[1] We build the k-NN graph via a public available package[2] and the Matlab implementation of our methods can be downloaded from http://c2inet.sce.ntu.edu.sg/Mingkui/spectral-feature.htm. All the experiments are performed on AMD Athlon (tm) 64 X2 Dual Core Processor 4400+ 2.31 GHz 2GB RAM PC.

4.2. Toy experiments

In the toy data experiments, we collect eight toy features as the informative features. Feature 1&2 are linear features, and features 3&4, 5&6, 7&8 are nonlinear features. In this experiment, we gradually increase the noise to the dataset and test whether the considered methods can identify the eight informative features or not. Note that the eight features are all pair features and a single feature can not completely represent the structure of the dataset. Therefore, the features are identified if both two features in the same feature pair are identified. When adding the noises to the dataset, we gradually add 20 noise features to the dataset each time, among which half of them are uniform noises and the other half are norm noises. The maximum number of noise features is set to 800 and the largest number of noise features at which the informative feature pairs are recorded for each algorithm, as shown in Table 2.

From Table 2, we can see that all the algorithms listed in the table can correctly identify the linear features. However, for the nonlinear features, the performance varies. Using the θ_G-LDA score, the nonlinear feature cannot be identified at all. The reason is that the θ_G-LDA score does not consider the nonlinear structure of the dataset. Different from the θ_G-LDA score, the θ_G-LSDF score and θ_G-MFA score show improved performance on the nonlinear feature identification. That is to say, with the sub-manifold structure considered, the nonlinear score can be identified. However, both methods show great sensitivity to the noises. For example, the second feature pair cannot be identified if number of noises is greater than 100 for the θ_G-LSDF and 120 for the θ_G-MFA score. The detailed reasons have been discussed in the previous section. As to the SVM-RFE method with linear kernel, it shows great performance on linear feature selection, but failed to identify the nonlinear features. Compared with the θ_G-score, the SL method shows better performance on selecting the nonlinear features. However, as previously mentioned, the performance of the SL method will decline when the noises increase. For the SL-RFE and FS-RFE method, they can identify the linear and nonlinear features when there are large number of noise features.

In this paper, we will further discuss the CPU time to the sample size and the feature size respectively. First, we fix the sample number to 1000 and increase the features. The time needed for each feature size is shown in Fig. 5. Note that except for SVM-RFE, all other methods are implemented in Matlab. Hence the training time is only for reference. From Fig. 5, with features increasing, the

Table 2
Experiments on toy datasets.

Method	Feature 1&2	Feature 3&4	Feature 5&6	Feature 7&8
θ_G-LDA score	**800**	0	0	0
θ_G-LSDF score	**800**	100	240	320
θ_G-MFA score	**800**	120	260	320
SVM-RFE	**800**	0	0	0
SL	**800**	200	300	420
SL-RFE	**800**	**800**	**800**	**800**
FS-RFE	**800**	**800**	**800**	**800**

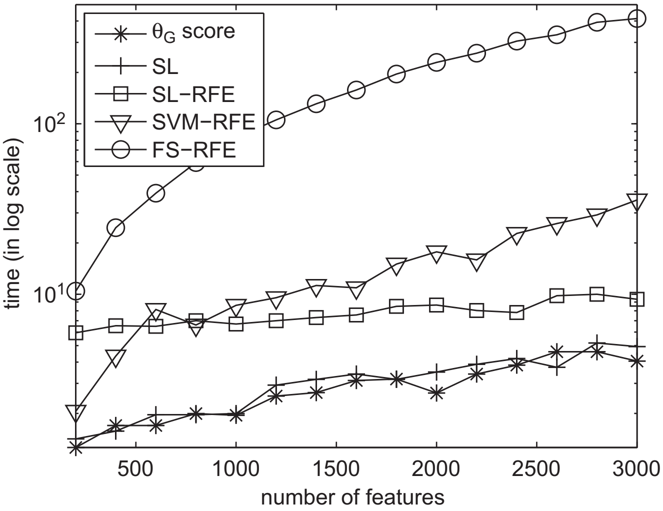

Fig. 5. The training time of various methods with different numbers of features (with number of samples $n=1000$).

[1] http://www.csie.ntu.edu.tw/~cjlin/liblinear/
[2] http://www.zjucadcg.cn/dengcai/Data/DimensionReduction.html

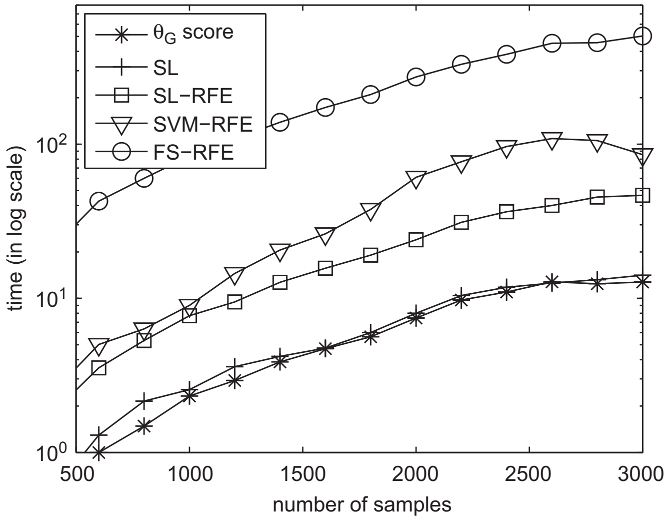

Fig. 6. The training time of various methods with different numbers of samples (with number of features $m = 1000$).

time needed by the FS-RFE method shows the greatest rise. The θ_G-score, SL method, SL-RFE and SVM-RFE methods show slower increase regarding to the dimensionality. Second, we just fix the feature number to 1000 and gradually increase the sample size. The time regarding to the number of sample size is recorded in Fig. 6. From Fig. 6, we can see that the FS-RFE method shows the fast growth rate in time regarding to the sample size. However, the θ_G-score, SL method, SL-RFE and SVM-RFE method show lower time growth rates regarding to the increase of the sample size. Hence generally speaking, the SL-RFE method shows better performance while maintaining a lower time complexity.

4.3. Two-class real-data experiments

To further verify the performance of the proposed methods, some real-world benchmark datasets are adopted to test their performance. The first experiment is conducted on three datasets, namely Australian, German and Leukemia. In each dataset, the results of testing accuracy versus selected feature number are shown in Fig. 7. The left figures in Fig. 7 are the results obtained by the SVM classifier and the right figures are obtained by the

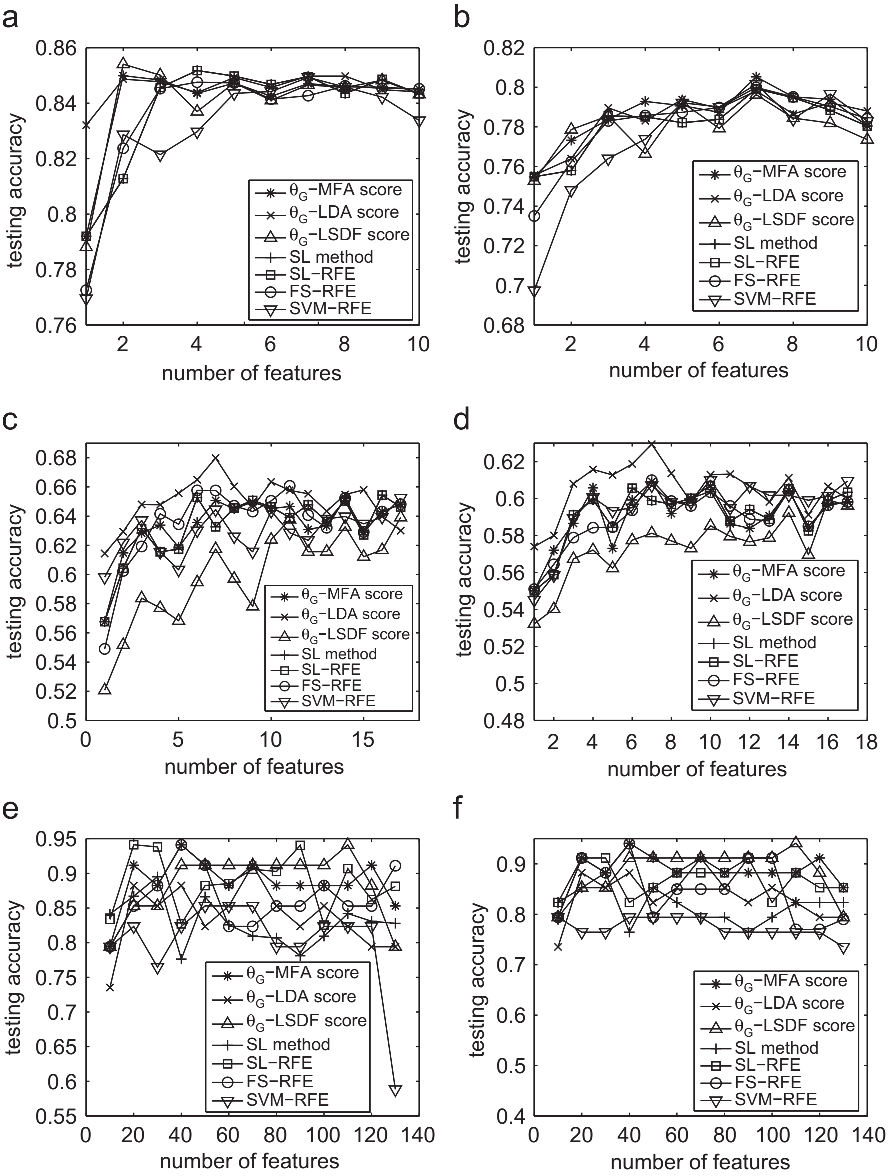

Fig. 7. The prediction accuracy versus dimension. (a) Australian (SVM). (b) Australian (K-NN). (c) German (SVM). (d) German (K-NN). (e) Leukemia (SVM). (f) Leukemia (K-NN).

K-NN classifier. Generally speaking, the predicting accuracies by the SVM are higher than the K-NN classifier. Obviously, our proposed FS-RFE method shows competitive performance compared with other methods. However, the LDA score, MFA score and LSDF score have also shown good enough performance. That is to say, in the problem with small features, the θ_G-score can be very effective. Two possible reasons account for this fact. On the one hand, the small features have little influence on the data structure. On the other hand, the informative features of these datasets are possibly with linear relationship. The Leukemia dataset contains 7129 features with 72 samples. In the experiment, we use the default 38 samples for training and the left 34 samples for testing [9]. The prediction accuracy obtained by SVM and K-NN with respect to the number of selected features are recorded in Fig. 7(e) and (f), respectively. From Fig. 7(e) and (f), the FS-RFE obtains the best performance among the methods.

4.4. Multi-class real-data experiments

In this experiment, several multi-class problems are adopted to test the various methods. Except for the vehicle dataset, there are three image datasets, i.e. UMIST, USPS and Yalefaces. All image datasets are with high dimensions. Furthermore, as commonly acknowledged, the pixels in nature images are nonlinear and highly correlated. The prediction accuracy regarding to the number of selected features are shown in Fig. 8. Considering that the SVM-RFE is hard to be applied for multi-class problems, we do not consider this method in multi-class problems. From the results presented in Fig. 8, the proposed FS-RFE method obtains the best results on the four benchmark datasets, which demonstrate the effectiveness and high performance of our method.

4.5. Multi-class real-data with noises

The major concentration of this paper is to avoid the negative influence brought by the noise features on the graph. In this experiment, we will test the performance of proposed methods over increasing number of noise features on UMIST, USPS and Yalefaces datasets, which have more complex structures. To implement the experiment, we gradually increase the number of noise features to those datasets and fix the number of selected features to 50. Other experimental settings are the same as the above experiment. The testing accuracy regarding the number of noise features are shown in Fig. 9. From the results presented in Fig. 9, we can observe the following facts. At first, on all the three datasets, when increasing the noise features, the proposed SL-RFE and FS-RFE can always obtain much better performance over the benchmark algorithms, which further verified the validity of these two methods. Secondly, on Yalefaces, the prediction accuracy of the benchmark shows apparent decline with the increasing noise features while SL-RFE and FS-RFE show much stable performance, which verify the importance of the proposed two methods. However, on USPS and UNIST, the performance of the benchmark algorithms shows very little decline in prediction accuracy. Two reasons account for this. At first, the structure of these two dataset may be simple and we can still obtain relatively stable graph on these two datasets even with a lot of noise features. This fact further indicates that even with large number of noises, we can possibly obtain a relatively stable graph and hence we can remove the most non-informative features regarding this graph. The second reason is due to the way of construction. Specifically in this experiment, both MFA graph and LSDF graph highlight the importance of labels, which makes the graph have stable performance over noises. While for LDA, the graph is defined by the labels and will not be affected by the noise features.

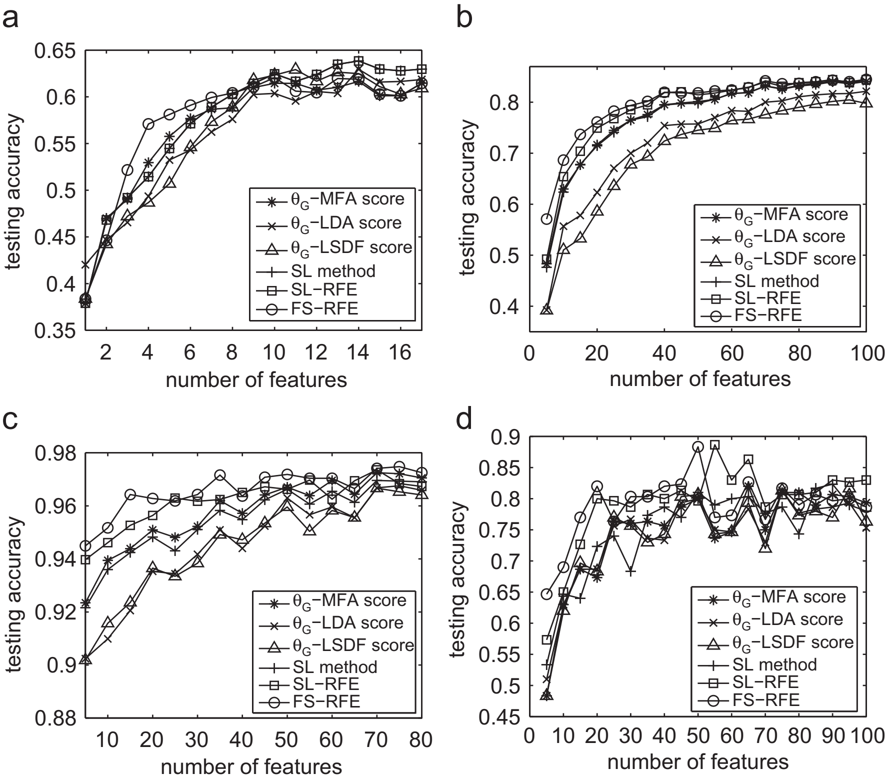

Fig. 8. Prediction accuracy with selected features on real dataset. (a) Vehicle. (b) USPS. (c) UMIST. (d) Yalefaces.

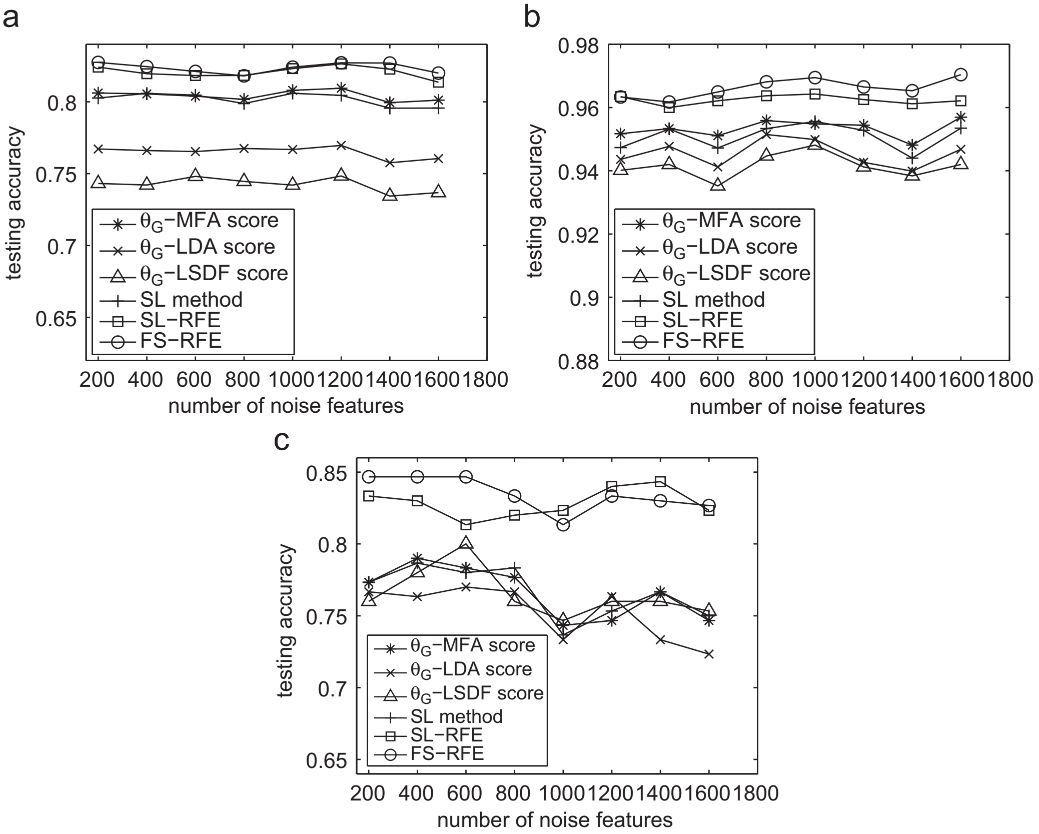

Fig. 9. Prediction accuracy over the number of noise feature with 50 selected features. (a) USPS. (b) UMIST. (c) Yalefaces.

5. Conclusion

In this paper, we developed two new feature selection methods based on the recently developed graph embedding idea. Firstly, we show that the recently developed feature scores can be seen as a direct extension of the graph preserving criterion. Further more, we apply the recursive feature elimination scheme to identify the optimal feature subset to reduce the negative influence brought by the noise features. The experimental results both on toy datasets and real-world datasets verify that the proposed RFE methods can achieve the state-of-art performance.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which have considerably improved this paper. This work was supported by the National Natural Science Foundation of China (No. 61172161).

References

[1] J. Zhao, K. Lu, X. He, Locality sensitive semi-supervised feature selection, Neurocomputing 71 (2008) 1842–1849.

[2] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.

[3] R. Cai, Z. Hao, W. Wen, H. Huang, Kernel based gene expression pattern discovery and its application on cancer classification, Neurocomputing 73 (2010) 2562–2570.

[4] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley-Interscience, Hoboken, NJ, 2000.

[5] A.R. Teixeira, A.M. Tomé, E.W. Lang, Unsupervised feature extraction via kernel subspace techniques, Neurocomputing 74 (2011) 820–830.

[6] S.C. Yan, D. Xu, B.Y. Zhang, H.J. Zhang, Graph embedding: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 40–51.

[7] M. Han, Z.P. Liang, D.C. Li, Sparse kernel density estimations and its application in variable selection based on quadratic Renyi entropy, Neurocomputing 74 (2011) 1664–1672.

[8] Q.H. Hu, X.J. Che, L. Zhang Lei, Feature evaluation and selection based on neighborhood soft margin, Neurocomputing 73 (2010) 2114–2124.

[9] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.

[10] F.P. Nie, H. Huang, X. Cai, Efficient and robust feature selection via joint l2,1-norms minimization, in: Neural Information Processing Systems, Canada, 2010.

[11] F.P. Nie, S.M. Xiang, C.S. Zhang, Neighborhood MinMax projections, in: Joint Conference on Artificial Intelligence, India, 2007.

[12] H. Liu, J. Li, L. Wong, A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns, Genome. Inform. 27 (2002) 51–60.

[13] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University, Oxford, 1995.

[14] F. Nie, S. Xiang, Y. Jia, C. Zhang, S. Yan, Trace ratio criterion for feature selection, in: 23rd AAAI Conference on Artificial Intelligence, Chicago, 2008.

[15] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Neural Information Processing Systems, Vancouver, 2005.

[16] N. Cristianini, T.J. Shawe, An Introduction to SVM, Cambridge University Press, Cambridge, 2000.

[17] D. Cai, X.F. He, J.W. Han, H.J. Zhang, Orthogonal laplacianfaces for face recognition, IEEE Trans. Image Process. 15 (2006) 3608–3614.

[18] D. Xu, S. Yan, D. Tao, S. Lin, H.J. Zhang, Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval, IEEE Trans. Image Process. 16 (2007) 2811–2821.

[19] E. Alba, N.J. Garcia, L. Jourdan, E.G. Talbi, Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms, in: IEEE Congress on Evolutionary Computation, Singapore, 2007.

[20] LIBSVM data sets. 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm/〉.

[21] UMIST database. 〈http://www.sheffield.ac.uk/eee/research/iel/research/face.html〉.

[22] Yale univ. face database. 〈http://cvc.yale.edu/projects/yalefaces/yalefaces.html〉.

[23] X. He, Learning a maximum margin subspace for image retrieval, IEEE Trans. Knowl. Data Eng. 20 (2008) 189–201.

[24] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

Dan Wei received the B.S. degree and M.S. degree from Hunan University in 2006 and 2008. Now she is a Ph.D. candidate at the College of Electrical and Information Engineering, Hunan University, China. Her research interests are focused on image processing and pattern recognition.

Shutao Li received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Hunan University, in 1995, 1997, and 2001, respectively. He joined the College of Electrical and Information Engineering, Hunan University, in 2001. He was Research Associate in the Department of Computer Science, Hong Kong University of Science and Technology, from May 2001 to October 2001. From November 2002 to November 2003, he was a postdoctoral fellow at the Royal Holloway College, University of London, working with Prof. John-Shawe-Taylor. During April 2005 to June 2005, he has visited the Department of Computer Science, Hong Kong University of Science and Technology as a visiting professor. Now, he is a full professor with the College of Electrical and Information Engineering, Hunan University. He has authored or coauthored more than 120 refereed papers. His professional interests are information fusion, image processing, and pattern recognition.

Mingkui Tan received the B.S. degree and M.S. degree from Hunan University in 2006 and 2009, and he is currently pursuing his Ph.D. degree at the Nanyang Technological University, Singapore. His technical interests include particle swarm optimization and large scale machine learning.