FAST FEATURE RANKING AND ITS APPLICATION TO FACE RECOGNITION

Pan Feng（潘锋）[1]，*Wang Jiandong*（王建东）[2]，*Song Guangwei*（宋广为）[1]
Niu Ben（牛奔）[1]，*Gu Qiwei*（顾其威）[1]

（1. College of Management，Shenzhen University，Shenzhen，518060，P. R. China；
2. College of Computer Science and Technology，Nanjing University of Aeronautics and Astronautics，Nanjing，210016，P. R. China）

Abstract：A fast feature ranking algorithm for classification in the presence of high dimensionality and small sample size is proposed. The basic idea is that the important features force the data points of the same class to maintain their intrinsic neighbor relations，whereas neighboring points of different classes are no longer to stick to one another. Applying this assumption，an optimization problem weighting each feature is derived. The algorithm does not involve the dense matrix eigen-decomposition which can be computationally expensive in time. Extensive experiments are conducted to validate the significance of selected features using the Yale，Extended YaleB and PIE datasets. The thorough evaluation shows that，using one-nearest neighbor classifier，the recognition rates using 100—500 leading features selected by the algorithm distinctively outperform those with features selected by the baseline feature selection algorithms，while using support vector machine features selected by the algorithm show less prominent improvement. Moreover，the experiments demonstrate that the proposed algorithm is particularly efficient for multi-class face recognition problem.

Key words：feature selection；feature ranking；manifold learning；Laplacian matrix

CLC number：TP301.6 **Document code**：A **Article ID**：1005-1120(2013)04-0389-08

INTRODUCTION

High throughput technologies now routinely produce large data sets characterized by enormous number of features in data mining and machine learning field. However，not all of them are really correlated with the class labels. Many irrelevant and redundant features may exist in noisy data，which poses serious time and cost challenges to the traditional statistical learning methods. For example，it is reported that the support vector machine（SVM）algorithm，one of the most advanced classifiers，has a worst-case sample complexity that grows at least linearly in the number of irrelevant features[1]. Researchers and practitioners have realized that data preprocessing plays an essential role for successful data mining tasks. Feature selection is one of the effective and frequently used data preprocessing method. It can reduce the computational cost for classifcation or regression，alleviate the risk of overfitting in situations with small training set size，and help to reveal unknown relationship among features. An interesting observation is if a good feature subset of the data is provided，even the naive classifier such as *k* nearest neighbor rule can achieve sufficiently high classification accuracy[2]. Many schemes have been suggested to solve the feature selection problem. They can be categorized into three groups：Embedded，filter and wrapper models. The filter model selects the optimal feature subset in term of general characteristics of the data，hence it is independent of any classifier. The embedded and wrapper models construct and

Website of on-line first：**http**：//www.cnki.net/kcms/detail/32.1389.V.20121219.1137.003.html(2012-12-19 11：37).

Foundation items：Supported by the National Nature Science Foundation of China（71001072）；the Natural Science Foundation of Guangdong Province（9451806001002294）.

Received date：2011-12-29；**revision received date**：2012-04-04

Corresponding author：Song Guangwei，Associate Professor，E-mail：broadfunc@hotmail.com.

select feature subsets that are useful to build a good predictor. For example, Guyon utilizes SVM as a subroutine (wrapper) in the feature selection process with the purpose of optimizing the SVM accuracy on the resulting subset of features[3]. For high dimensional data, the filter model is often preferable because of its usability with alternative classifiers, its computational speed and its simplicity[4], although it can not always achieve the comparative classification performance as wrapper and embedded models.

The remarkable filter method, RELIEF, is proposed to weight the features for two-class problems[5]. The algorithm iteratively calibrates the weight of each feature by updating the hypothesis margin, which is defined as the difference between the distance from the point to its nearest neighbor in the same class and the distance from the point to its nearest neighbor in the opposite class. The RELIEF-F algorithm[6] is a generalization of RELIEF capable of handling multi-class problems, using an average of the k nearest neighbors of a sample instead of only one nearest neighbor of the sample to compute the margins. It is shown that the RELIEF-F algorithm is reliable enough to guide the feature searching. Further, it is pointed out that in both RELIEF and RELIEF-F the nearest neighbors of a given sample are predefined in the original feature space, hence liable to yield erroneous nearest point either in the same class or in different classes with copious irrelevant features[7]. The bias can be reduced following the principle of the EM algorithm, which approximates the real nearest point iteratively in the weighted feature space. However, the pseudo EM algorithm will necessarily increase the computational complexity of the approach.

A provable and important feature selection criterion for filter methods is mutual information[8-13]. Fast correlation-based filter (FCBF)[9] applies the concept of approximate Markov blanket to eliminate the feature redundancy, and uses the concept of symmetric uncertainty to determine the feature relevance. An optimal subset can thus be determined by a group of features with no approximate Markov blanket. Minimum redundancy-maximum relevance (MRMR)[10-11] feature selection framework incrementally selects features minimizing their redundancy with features chosen in previous steps and maximizing their relevance to the class simultaneously. Two criteria, mutual information difference and mutual information quotient are used and the latter is experimentally demonstrated better. All the mutual information based algorithms nevertheless have the high computational complexity in calculating the entropy.

In this paper, a fast feature ranking algorithm is proposed based on local learning. Feature ranking is a filter method: It is a preprocessing step, without trying to optimize the performance of any specific predictor. The main idea comes from manifold learning. We assume that the important features can force the data points of the same class to maintain their intrinsic neighbor relations, and the data points of different classes are no longer to stick to one another. The similar notion can be found in Lalacian Score[14], Fisher Score[15], HSIC[16], SPEC[17] and Trace Ratio[18]. In recent study[19] it is further proved that a unified similarity preserving framework encompasses these criteria. We extend this notion and emphasize the feature that will "pull" the neighbors closely in the same class and "push" the neighbors away in different classes. The experiments on the face datasets show the proposed algorithm often outperforms the well-known methods when used as a preprocessing step for classification rules, and has the time complexity less than or equal to that of RELIEF-F.

1 ALGORITHM

Given a $N \times d$ data set $\boldsymbol{D}$ consisting of N samples over d-dimensional feature space R_d representing d features $\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_d$ over N samples, let the row vectors of $\boldsymbol{D}$ be denoted by $\boldsymbol{x}_1^{\mathrm{T}}, \cdots, \boldsymbol{x}_N^{\mathrm{T}}$ and $y_1, \cdots, y_N$ the corresponding class labels. Most manifold learning algorithms aim to reveal the intrinsic distribution in a lower-dimensional space in the case whose data are densely distributed on a manifold. One may also expect that the significant features are characterized by maintaining the original neighbor relations for neighboring data points of the same class, and differentiating and keeping away neighboring points of different classes. Let $\boldsymbol{\omega} \in \mathbf{R}^d$ denote the weight vector, each element of which represents the significance of a feature. We firstly construct the neighborhood graphs. Let G_w and G_b denote two (undirected) graphs both over all data points. To construct G_w, we consider each pair of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from the same class, i. e. $y_i = y_j$. An edge is added between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ if $\boldsymbol{x}_j$ is one of $\boldsymbol{x}_i{}'$ s k-nearest neighbors or is one of $\boldsymbol{x}_j{}'$ s k-nearest neighbors (the other possibility is to consider the ε-ball implementation). For G_b , we instead consider each pair of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with $y_i \neq y_j$, and likewise, connect $\boldsymbol{x}_i$ and $\boldsymbol{x}_i$ if $\boldsymbol{x}_j$ is one of $\boldsymbol{x}_i{}'$ s k-nearest neighbors or $\boldsymbol{x}_i$ is one of $\boldsymbol{x}_j{}'$ s k-nearest neighbors. Then we can naturally specify the adjacent matrix $\boldsymbol{W}$ of graph G_w , where each element $\boldsymbol{W}_{ij}$ refers to the weight of the edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and is given by

$$\boldsymbol{W}_{ij} = \begin{cases} 1 & \boldsymbol{x}_i \in knn(j) \text{ or } x_j \in knn(i), \text{ and } y_i = y_j \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

The defined $N \times N$ matrix $\boldsymbol{W}$ is clearly symmetric and sparse. Similarly the other adjacent matrix $\boldsymbol{B}$ can be computed from the graph G_b , where each element $\boldsymbol{B}_{ij}$ refers to the weight of the edge between two vertices from different classes

$$\boldsymbol{B}_{ij} = \begin{cases} 1 & \boldsymbol{x}_i \in knn(j) \text{ or } \boldsymbol{x}_j \in knn(i), \text{and } y_i \neq y_j \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

The important features can preserve the closeness of each neighborhood within which the samples have the same label in the weighted feature space. Thus the following criterion should be minimized

$$J_w = \sum_i \sum_j \| \boldsymbol{w} * \boldsymbol{x}_i - \boldsymbol{w} * \boldsymbol{x}_j \|_2^2 \boldsymbol{W}_{ij} \tag{3}$$

where * denotes the element-wise multiplication and $\| \cdot \|_2$ the Euclidean distance.

Considering the matrix $\boldsymbol{W}$ is symmetric, we can rewrite the criterion as

$$J_w = \sum_i \sum_j \Big(\sum_{k=1}^{d} (\boldsymbol{w}_k \boldsymbol{x}_i^k - \boldsymbol{w}_k \boldsymbol{x}_i^k)^2 \Big) \boldsymbol{W}_{ij} =$$

$$\sum_{i,j} \Big(\sum_{k=1}^{d} \boldsymbol{w}_k^2 (\boldsymbol{x}_i^k - \boldsymbol{x}_i^k)^2 \boldsymbol{W}_{ij} \Big) =$$

$$\sum_{k=1}^{d} \boldsymbol{w}_k^2 \Big(\sum_{i,j} \boldsymbol{x}_i^{k^2} \boldsymbol{W}_{ij} + \sum_{i,j} \boldsymbol{x}_j^{k^2} \boldsymbol{W}_{ij} - 2 \sum_{i,j} \boldsymbol{x}_i^k \boldsymbol{x}_j^k \boldsymbol{W}_{ij} \Big) =$$

$$2 \sum_{k=1}^{d} \boldsymbol{w}_k^2 (\boldsymbol{f}_k^{\mathrm{T}} \times \mathrm{diag}(\boldsymbol{We}) \times \boldsymbol{f}_k - \boldsymbol{f}_k^{\mathrm{T}} \times \boldsymbol{W} \times \boldsymbol{f}_k)$$

where $\boldsymbol{x}_i^k$ denotes the kth element of $\boldsymbol{x}_i$, $\boldsymbol{e}$ is the vector of all ones, and diag($\boldsymbol{x}$) generates a diagonal matrix from the elements of the vector $\boldsymbol{x}$. Let $\boldsymbol{D}_w = \mathrm{diag}(\boldsymbol{We})$, $\boldsymbol{L}_w = \boldsymbol{D}_w - \boldsymbol{W}$, the criterion can be expressed as

$$J_w = 2 \sum_{k=1}^{d} \boldsymbol{w}_w^2 \boldsymbol{f}_k^{\mathrm{T}} \boldsymbol{L}_w \boldsymbol{f}_k$$

Let $\boldsymbol{T}_w = \mathrm{diag}([\boldsymbol{f}_1^{\mathrm{T}} \boldsymbol{L}_w \boldsymbol{f}_1, \boldsymbol{f}_d^{\mathrm{T}} \boldsymbol{L}_w \boldsymbol{f}_d]^{\mathrm{T}})$, the criterion can be further simplified as

$$J_w = 2 \boldsymbol{w}^{\mathrm{T}} \boldsymbol{T}_w \boldsymbol{w} \tag{4}$$

Note $\boldsymbol{L}_w$ is a positive semi-definite matrix, hence $\boldsymbol{f}_k^{\mathrm{T}} \boldsymbol{L}_w \boldsymbol{f}_k \geqslant 0, k = 1, \cdots, d$. It is clear that the matrix $\boldsymbol{T}_w$ is a diagonal matrix with all non-negative diagonal elements.

For the data in different classes, the important features can differentiate and keep away the neighboring points in the weighted feature space. Therefore a reasonable criterion is to maximize

$$J_b = \sum_i \sum_j \| \boldsymbol{w} * \boldsymbol{x}_i - \boldsymbol{w} * \boldsymbol{x}_j \|^2 =$$

$$\sum_i \sum_j \Big(\sum_{k=1}^{d} (\boldsymbol{w}_k \boldsymbol{x}_i^k - \boldsymbol{w}_k \boldsymbol{x}_i^k)^2 \Big) \boldsymbol{B}_{ij} =$$

$$2 \sum_{k=1}^{d} \boldsymbol{w}_k^2 (\boldsymbol{f}_k^{\mathrm{T}} \times \mathrm{diag}(\boldsymbol{Be}) \times \boldsymbol{f}_k - \boldsymbol{f}_k^{\mathrm{T}} \times \boldsymbol{B} \times \boldsymbol{f}_k) \tag{5}$$

Let $\boldsymbol{D}_b = \mathrm{diag}(\boldsymbol{Be})$, $\boldsymbol{L}_b = \boldsymbol{D}_b - \boldsymbol{B}$. The objective function(Eq. (5)) can be reformulated as

$$J_b = 2 \boldsymbol{w}^{\mathrm{T}} - \boldsymbol{T}_b \boldsymbol{w} \tag{6}$$

where $\boldsymbol{T}_b = \mathrm{diag}([\boldsymbol{f}_1^{\mathrm{T}} \boldsymbol{L}_b \boldsymbol{f}_1, \cdots, \boldsymbol{f}_2^{\mathrm{T}} \boldsymbol{L}_b \boldsymbol{f}_2, \cdots, \boldsymbol{f}_d^{\mathrm{T}} \boldsymbol{L}_b \boldsymbol{f}_d]^{\mathrm{T}}$. $\boldsymbol{T}_b$ is also a diagonal matrix with all non-negative diagonal elements.

With these two aspects of consideration, we provide two ranking criteria, e. g. the quotient criterion and the difference criterion

$$\mathrm{Max} \quad (J_b / J_w)$$

$$\mathrm{Max} \quad (J_b - J_w)$$

The quotient criterion leads to the following optimization problem

$$\max \quad \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}_w\boldsymbol{w}} \qquad (7)$$
$$\text{s. t. } \boldsymbol{w}(i) \geqslant 0 \qquad i=1,\cdots,d$$

where the constraint $\boldsymbol{w} \geqslant 0$ ensures the weight vector is a distance metric.

Eq. (7) can be approached in an efficient way. Firstly we eliminate the last constraint and reduce the problem to

$$\max \quad \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}_w\boldsymbol{w}} \qquad (8)$$

By applying the Lagrangian technique, we switch to a generalized eigen-decomposition problem $\boldsymbol{T}_b\boldsymbol{w}=\lambda\boldsymbol{T}_w\boldsymbol{w}$. The elements in eigenvector are necessarily non-negative because the matrices $\boldsymbol{T}_w$ and $\boldsymbol{T}_b$ are both diagonal with non-negative diagonal elements. Thus Eq. (7) is actually equivalent to Eq. (8). We can also take advantage of the diagonalization of the matrix $\boldsymbol{T}_w^{-1}\boldsymbol{T}_b$ to derive eigen-values directly as $\boldsymbol{\lambda}_k=\boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{L}_b\boldsymbol{f}_k/\boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{L}_w\boldsymbol{f}_k, k=1,\cdots,d$, and use them as the weights for the features, e. g. $\boldsymbol{w}_k=\boldsymbol{\lambda}_k$. The elements in represent the significance of the features and should be sorted in descending order.

It is noted here that $\boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{L}_w\boldsymbol{f}_k$ almost never reach the value of zero on real-world data sets, therefore the small sample size problem is avoided. A simple proof is sketched as follows. Suppose $\boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{L}_w\boldsymbol{f}_k=0$, which means $\sum_i\sum_j(\boldsymbol{x}_i^k-\boldsymbol{x}_j^k)^2\boldsymbol{W}_{ij}=0$. This equation requires that all the points within each neighborhood have the identical value in feature k, which should be removed in practice because it has no discriminating capability. In the experiments this phenomenon never happens, hence the further processing step is ignored.

Using the difference criterion, the optimization problem is formulated as follows

$$\begin{aligned} \max \quad & \boldsymbol{w}^{\mathrm{T}}(\boldsymbol{T}_b-\boldsymbol{T}_w)\boldsymbol{w} \\ \text{s. t.} \quad & \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}=1 \qquad (9) \\ & \boldsymbol{w}(i) \geqslant 0 \qquad i=1,\cdots,d \end{aligned}$$

where the constraint $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}=1$ prevents the maximization from increasing without bound. The diagonal matrix $\boldsymbol{T}_b-\boldsymbol{T}_w$ is generally positive semi-definite since the distances among the points from different classes are usually larger than the distances among the points from the same class. Observing the similar technique, the solution with no generalized eigen-vector computation can be derived.

To summarize, the main procedure is the algorithm based on the quotient criterion FRLL-Q and the algorithm based on the difference criterion FRLL-D, shown in Algorithm 1 and Algorithm 2.

Algorithm 1 FRLL-Q

```
Input:
The data set $\boldsymbol{D}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]\in\mathbf{R}^{N\times d}$ with class labels $y_1,\cdots,y_N$ ;
Output:
The order of features and the corresponding weights;
Step:
1. Construct the graph $G_w$ and $G_b$ on the data $\boldsymbol{D}$ with class labels;
2. Construct the adjacent matrices $\boldsymbol{W}$ and $\boldsymbol{B}$;
3. Compute the Laplacian matrix $\boldsymbol{L}_w$ and $\boldsymbol{L}_b$ from $\boldsymbol{W}$, $\boldsymbol{B}$, respectively;
4. For each feature $\boldsymbol{f}_k$ , $k=1,\cdots,d$ , there is
$\boldsymbol{w}_k=\boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{L}_b\boldsymbol{f}_k/\boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{L}_w\boldsymbol{f}_k$ ;
5. Sort $\boldsymbol{w}$ in descending order;
6. Return $\boldsymbol{w}$.
```

Algorithm 2 FRLL-D

```
Input:
The data set $\boldsymbol{D}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]\in\mathbf{R}^{N\times d}$ with class labels $y_1,\cdots,y_N$ ;
Output:
The order of features and the corresponding weights;
Step:
1. Construct the graph $G_w$ and $G_b$ on the data $\boldsymbol{D}$ with class labels;
2. Construct the adjacent matrices $\boldsymbol{W}$ and $\boldsymbol{B}$;
3. Compute the Laplacian matrix $\boldsymbol{L}_w$ and $\boldsymbol{L}_b$ from $\boldsymbol{W}$,$\boldsymbol{B}$, respectively;
4. For each feature $\boldsymbol{f}_k$ , $k=1,\cdots,d$ , there is
$\boldsymbol{w}_k=\boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{L}_b\boldsymbol{f}_k-\boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{L}_w\boldsymbol{f}_k$ ;
5. Sort $\boldsymbol{w}$ in descending order;
6. Return $\boldsymbol{w}$.
```

2 COMPUTATIONAL ANALYSIS

A time complexity analysis of FRLL-Q and FRLL-D is analyzed and compared with that of other algorithms. It can be found in case of $d \gg N$, the two FRLLs are fast feature ranking algorithms comparable to RELIEF-F.

The computation of FRLLs contains three steps: Construct the graphs G_w , G_b and the adjacent matrices $\boldsymbol{B}$, $\boldsymbol{W}$, compute the weight vector $\boldsymbol{w}$, and sort the weights. The cost of the first step is mainly the costs of computing the pairwise Euclidean distances and sorting the distances in ascending order, which require around $\frac{1}{2}N^2d+2N_d$ and $2N^2\log_2 N$ computations respectively if k-nearest neighbor algorithm is adopted to construct the adjacent graph[16-17]. Obtaining the d weights of the features in the second step requires around $2N^2d$ computations, while sorting the weights in the last step requires around $d\log_2 d$ computations. We can conclude the computational complexity of FRLL-Q and FRLL-D algorithms is $\frac{3}{2}N^2d+2Nd+2N^2\log_2 N+d\log_2 d$.

The RELIEF-F algorithm consists of computing the pairwise Euclidean distances and sorting the distances in ascending order identical to the first step of FRLLs, iterating the weight vector for times, and sorting the weights in descending order. Let c denote the class number, the three steps require around $\frac{1}{2}N^2d+2Nd+cN^2\log_2 N$, cNd , and $d\log_2 d$ computations respectively, hence the overall computational complexity is $\frac{1}{2}N^2d+2Nd+cN^2\log_2 N+cNd+d\log_2 d$, which is not as high as FRLLs for two-class problems. With the class number increasing, the computation of FRLLs may be fewer than that of RELIEF-F.

To the best of our knowledge, FCBF is the fastest feature selection algorithm among all the mutual information based methods. To determine the relevant features in the first step requires complexity $O(d)$; to determine the predominant features in the second step requires a best-case complexity $O(d)$ when only one feature is selected and all of the rest of the features are removed, and a worst-case complexity $O(d^2)$ when all features are selected. In the experiments, it can be seen that the time cost of FCBF basically presents a polynomial dependence on the number of dimensionality, thus for very high dimensional data sets, FRLL-Q and FRLL-D often show the computational advantage over FCBF.

3 EXPERIMENTS

The aim of the experiments described here is twofold: (1) To compare classification accuracy achieved using FRLLs versus other algorithms; (2) to compare their computational cost. All the experiments are implemented in Matlab 7.0 and run on a Pentium (R) 4 CPU 3.60 GHz machine with 4 GB RAM.

3.1 Feature selection for face classification

The effectiveness of the feature ranking algorithm obtained on three face data sets is analyzed, including Yale, Extended YaleB and PIE. All images are scaled to 32 pixel × 32 pixel . For YaleB and PIE, only ten classes of data are selected respectively for feature selection and validation. Each class contains 15 randomly chosen images, hence $d \gg N$. The descriptions of the data sets are summarized in Table 1, where # Feature represents the number of the original features, # Samle the sample size, and # Class the number of classes.

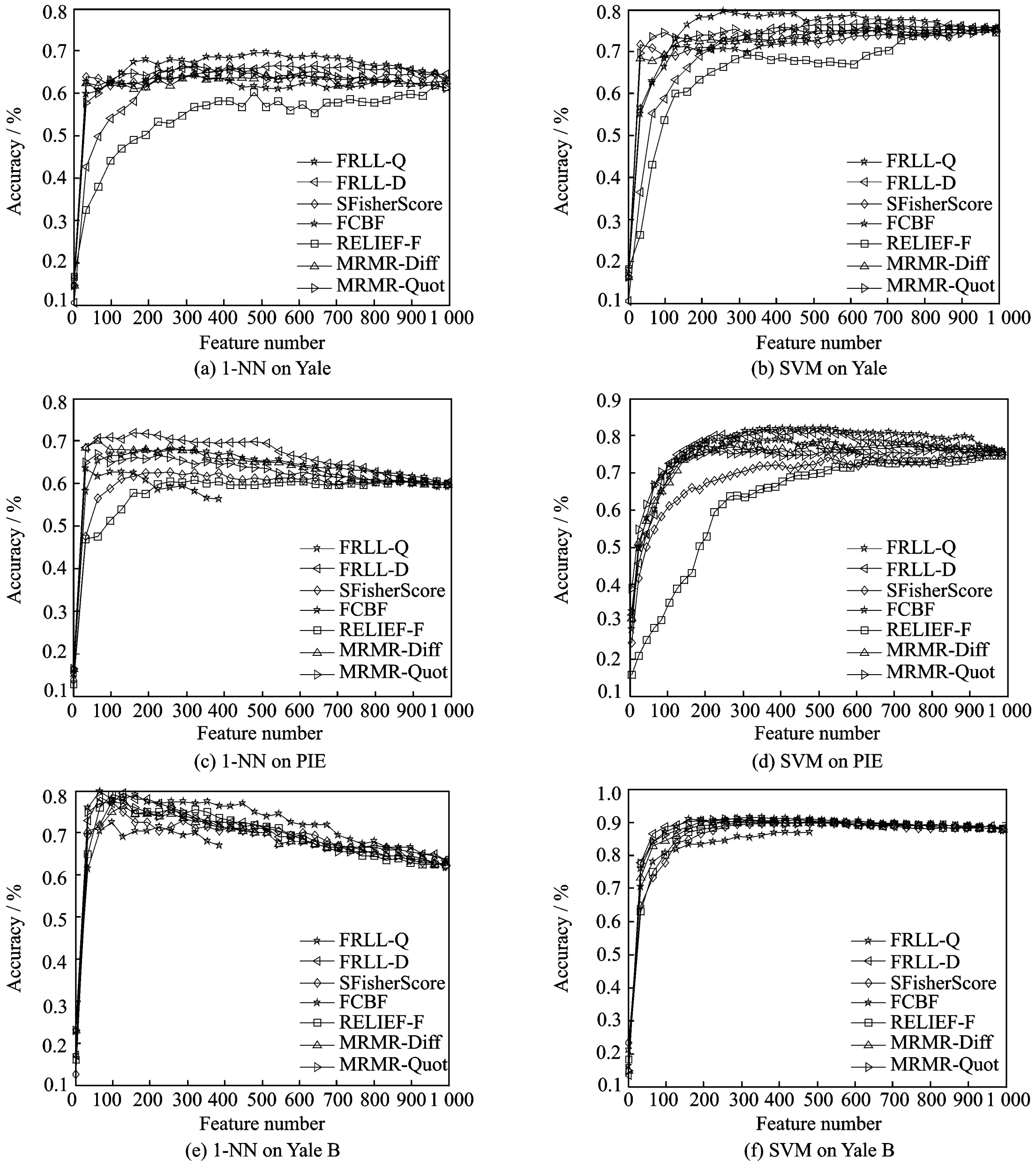
Table 1 Summary of face datasets

Data set	# Feature	# Sample	# Class
Yale	1 024	165	15
YaleB	1 024	150	10
PIE	1 024	150	10

The classification rates preprocessed by the proposed algorithms are compared with the aforementioned RELIEF-F, subset-level fisher score (SFisherScore), FCBF, MRMR-MID and MRMR-MIQ. In the proposed algorithms and RELIEF-F the number of the nearest neighbors is always set to be 5. For classes with too few samples (<5), all data points are used as neighbors. SFisherScore is an iterative algorithm finding the optimal feature subset such that the fisher score is maximized[18-19]. The optimal feature subset is always used, instead of the features selected by sequential forward selection. For FCBF, the threshold is fixed as default value 0 to obtain a descending order of all the relevant features in the first step. The MRMR-Diff and MRMR-Quot algorithms, corresponding to the MRMR framework with mutual information difference and mutual information quotient criteria, are superior in gene expression selection. The codes are downloaded from the author' s website[11]. Furthermore, it has been reported that the mutual information based feature selection algorithms, e. g. FCBF, MRMR-Diff and MRMR-Quot perform better if discretization methods have been applied on the continuous data[10-11]. For simplicity here each feature is discretized in three segments $(-\infty,\mu-\sigma]$, $[\mu-\sigma,\mu+\sigma]$, and $(\mu+\sigma,+\infty)$, where μ is the sample mean of training data and σ its standard deviation.

The performance of the algorithms is measured by the classification accuracy rate with selected features via five-fold cross-validation. The process is repeated for 10 times and the averaged accuracy rates versus selected feature number are recorded. To calculate the classification accuracy, 1-nearest neighbor classifier and linear SVM are used. The parameter in SVM (cost) is also tuned via cross-validation and the best accuracy is adopted. The testing results versus the increasing numbers of features are plotted in Fig. 1.

In Figs. 1(a,c,e), it can be seen on Yale and



(a) 1-NN on Yale

(b) SVM on Yale

(c) 1-NN on PIE

(d) SVM on PIE

(e) 1-NN on Yale B

(f) SVM on Yale B

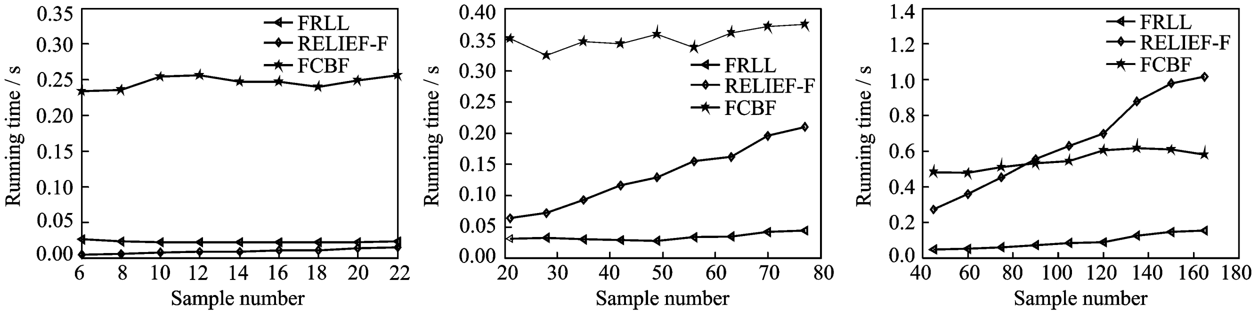
Fig. 1 Classification error as function of the number of features for data sets

YaleB the best classification performance of the 1-nearest neighbor classifier is obtained with features selected by FRLL-Q, while on PIE is obtained with features selected by FRLL-D. Noticeably, the three accuracy-rate curves have similar tendency. They increase rapidly and constantly decrease, and then converge at some points, indicating that more irrelevant features lead to worse classification. The overall accuracy rates achieved by FRLL-Q and FRLL-D are higher than that by the baseline algorithms. However, it is observed that MRMR-Diff and MRMR-Quot have the comparable or a little better performance when only very few features are selected. The reason may lie in the fact that FRLLs are based on local learning, capable of classifying the data in each local space, thus need more features to recognize a face image, whereas MRMR-Diff and mRMR-Quot directly estimate the correlation between individual feature and the label in the global hypothesis space. Figs. 1(b, d, f) show the respective comparison results with the SVM classifier. Features selected by the proposed algorithms still indicate a great discriminating strength on Yale, but show less prominent improvement than the baseline algorithms on the other two datasets. It may be due to the superiority of the SVM classifier.

3.2 Time complexity results

Since the previous subsection has established the effectiveness of the FRLL-Q and FRLL-D, it is useful now to compare the proposed algorithms and other feature selection approaches empirically with respect to time complexity. As stated in Section 1, the running time of FRLLs and RELIEF-F is of the same order of the multitude, both showing the polynomial dependence on the training data size when dimension is fixed. On the contrary, the time cost of FCBF has polynomial dependence on the number of the dimensionality. It is expected that the FRLLs and RELIEF-F algorithms will show less computation complexity than FCBF in high dimensional data sets with small sample size. Below to confirm experimentally this theoretical analysis, time consumption as a function of the number of training examples is measured on the 2-class Yale, 7-class Yale and 15-class Yale datasets. Number of data points in each class ranges from 3 to 7, and other settings are same as the aforementioned section.

Fig. 2 shows the time consumed for the three face datasets. Since FRLL-Q and FRLL-D almost have the same computational cost, we only plot the time curve of the FRLL-Q. It can be seen for the 2-class problem, time cost of the proposed algorithm is greater than that of RELIEF-F, but on the 7-class and 15-class datasets FRLL-Q is the most computationally efficient. Moreover, for FCBF the slop of the dependence on the number of the patterns is approximately linear whereas RELIEF-F and FRLL-Q have the polynomial dependence on the number of samples, which presents that the FRLL-Q algorithm is appropriate if small sample size datasets are provided.



(a) On 2-class Yale (b) On 7-class Yale (c) On 15-class Yale

Fig. 2 Time cost for three face datasets by different algorithms

4 CONCLUSION

The main contribution of the paper is to provide a principled way to perform feature weighting for classification problems with high data dimensionality. It avoids any heuristic combinatorial search, and hence can be implemented fast. The algorithm is based on the graph Laplacian notion, but has no eigenvector computation involved which will lead to a huge save of both time and memory. Extensive experimental results show that the proposed algorithms consistently outperform the state-of-the-art RELIEF-F extensions for face recognition problems considering both effectiveness and efficiency.

The FRLL framework forms the initial study for feature ranking. From the solution it can be seen that the feature selection procedure is greedy and unable to handle feature redundancy. If, for instance, there are many correlated features in the dataset, the leading features may not be optimal for classification. It has been known that redundant features can adversely affect the performance of classification, therefore should be removed by feature selection. That will be investigated in the future work.

References:

[1] Ng A. Feature selection. L1 vs. L2 regularization, and rotational invariance[C]// Proceedings of the Twenty-First International Conference on Machine Learning. New York, USA: ACM Press, 2004:78.

[2] Gilad-Bachrach R, Navot A, Tishby N. Margin based feature selection-theory and algorithms[C]// Proceedings of the Twenty-First International Conference on Machine Learning. New York, USA: ACM Press, 2004: 43.

[3] Weston J, Mukherjee S, Chapelle O, et al. Feature selection for SVMs[C]// Advances in Neural Information Processing Systems. Vancouver, Canada: MIT Press, 2001: 668-674.

[4] Guyon I, Elisseeff A. An introduction to variable and feature selection[J]. Journal of Machine Learning Research, 2003, 3:1157-1182.

[5] Kira K, Rendell L. A practical approach to feature selection[C]// Proceedings of the Ninth International Workshop on Machine Learning. San Francisco, USA: Morgan Kaufmann Publishers Inc., 1992:249-256.

[6] Sun Y, Li J. Iterative RELIEF for feature weighting [C]// Proceedings of the 23rd International Conference on Machine Learning. New York, USA: ACM Press, 2006: 913-920.

[7] Sun Y, Todorovic S, Goodison S. Local-learning-based feature selection for high-dimensional data analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9):1610-1626.

[8] Qiao Lishan, Zhang Limei, Sun Zhonggui. Self dependent locality preserving projection with transformed space oriented neighborhood graph [J]. Transactions of Nanjing University of Aeronautics and Astronautics,2010,27(3):261-268.

[9] Yu L, Liu H. Efficient feature selection via analysis of relevance and redundancy[J]. Journal of Machine Learning Research, 2004, 5:1205-1224.

[10] Ding Weiping. Minimum attribute co-reduction algorithm based on multilevel evolutionary tree with self-adaptive subpopulations[J]. Transactions of Nanjing University of Aeronautics and Astronautics,2013,30(2):175-184.

[11] Peng H, Long F, Ding C. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(8):1226-1238.

[12] Yan Xuemei. Dual-sparsity preserving projection[J]. Transactions of Nanjing University of Aeronautics and Astronautics,2012,29(3):284-288.

[13] Estévez P, Tesmer M, Perez C, et al. Normalized mutual information feature selection[J]. IEEE Transactions on Neural Networks, 2009, 20(2):189-201.

[14] He X, Cai D, Niyogi P. Laplacian score for feature selection[C]// Advances in Neural Information Processing Systems. Vancouver, Canada: MIT Press, 2006: 507.

[15] Wang L, Shen C, Hartley R. On the optimality of sequential forward feature selection using class separability measure[C]// 2011 International Conference on Digital Image Computing Techniques and Applications (DICTA). Noosa, Australia: IEEE, 2011:203-208.

[16] Song L, Smola A, Gretton A, et al. Feature selection via dependence maximization[J]. Journal of Machine Learning Research, 2007,13(1):1393-1434.

[17] Zhao Z, Liu H. Spectral feature selection for supervised and unsupervised learning[C]// Proceedings of the 24th International Conference on Machine Learning. New York, USA: ACM, 2007:1151-1157.

[18] Nie F, Xiang S, Jia Y, et al. Trace ratio criterion for feature selection[C]// Proceedings of the 23rd National Conference on Artificial Intelligence. Chicago, USA: AAAI Press, 2008:671-676.

[19] Zhao Z, Wang L, Liu H, et al. On similarity preserving feature selection[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 25(3):619-632.