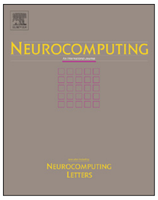
Contents lists available at ScienceDirect

Neurocomputing

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



Efficient sequential feature selection based on adaptive eigenspace model

Nannan Gu [a], Mingyu Fan [b,*], Liang Du [c], Dongchun Ren [d]

[a] School of Statistics, Capital University of Economics and Business, Beijing 100070, China
[b] Institute of Intelligent Systems and Decision, Wenzhou University, Wenzhou 325000, China
[c] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[d] Beijing 7Invensun Technology Co., Ltd., Beijing 100080, China

ARTICLE INFO

Article history:
Received 15 September 2014
Received in revised form
4 February 2015
Accepted 14 February 2015
Communicated by Zhi Yong Liu

Keywords:
Feature selection
Fisher score
Incremental learning
Eigenspace model

ABSTRACT

Though Fisher score is a representative and effective feature selection method, it has an unsolved drawback: it either evaluates the features individually and selects the top features, or selects features using the sequential search strategies. The individual-method ignores the mutual relationship among the selected features while the sequential-methods always suffer from heavy computation. In this work, we present an efficient sequential feature selection method. In the proposed method, the generalized Fisher score is used as a robust measurement of the discriminative ability of the features, which can naturally deal with the Small Size Sample problem. Besides, each feature is considered as a pattern vector and an adaptive eigenspace model is applied to update the generalized Fisher score. In the proposed adaptive eigenspace model, the size of the eigen-decomposition problems does not increase with the number of selected features, but is determined by the dimension of the adaptive eignespace. If the dimension of the adaptive eigenspace model is fixed, the proposed algorithm approximately consumes constant time to evaluate a candidate feature. Therefore, the proposed method is computationally more efficient than the traditional sequential methods. Experiments on six widely used face databases are conducted to demonstrate the efficacy of the proposed approach.

1. Introduction

With the development of information technology, researchers are often confronted with high-dimensional data. For example, in face recognition [1], an image of size $m \times n$ is often represented as a Gabor wavelet vector whose dimensionality is as high as $40 \times m \times n$. Similarly, biology databases such as microarray data can have tens of thousands of features [2,3]. Such a large number of features are always superfluous and can easily lead to the problem of *the curse of dimensionality* [4]. This significantly increases the time and space requirements for processing the data. Moreover, learning tasks, such as classification, clustering, retrieval and others [5], may become analytically or computationally intractable in very high-dimensional spaces [6].

Feature selection [7], which reduces the dimensionality of data by identifying the most informative features and removing the redundant and irrelevant ones, is an important preprocessing stage and is one of the two ways of avoiding the curse of dimensionality (the other is feature extraction). It brings the immediate effects for applications [2]: speeding up a data mining algorithm, and improving mining performance such as predictive accuracy and result comprehensibility. Many feature selection methods have been proposed based on different motivations. For example, there have been some methods that select features by keeping intrinsic manifold structure of data, including discriminative feature selection based on manifold regularization (FS-Manifold) [8], minimum–maximum local structure information Laplacian score (MMLS) [9], a variance minimization criterion for feature selection [10], and a global and local structure preservation framework for feature selection (GLSPFS) [11]. Recently, due to rapid development in optimization [12,13], there is a trend in using l_1 or $l_{2,1}$ norm minimization for feature selection. Typical works in this field include l_1-SVM [14], nonnegative discriminative feature selection (NDFS) [15], unsupervised discriminative feature selection (UDFS) [16], joint embedding learning and sparse regression (JELSR) [17], and clustering-guided sparse structural learning (CGSSL) [18].

Generally speaking, feature selection can be categorized into three families [7]: the embedded, wrapper and filter approaches. The embedded methods directly integrate feature selection into the training process of a given learning algorithm [16,19–22]. The wrapper methods require a predetermined learning algorithm and

[*] Corresponding author. Tel.: +86 13616641250.
E-mail addresses: nngu@cueb.edu.cn (N. Gu), fanmingyu@wzu.edu.cn (M. Fan), duliang@iso.ac.cn (L. Du), dongchun.ren@gmail.com (D. Ren).

http://dx.doi.org/10.1016/j.neucom.2015.02.043
0925-2312/© 2015 Elsevier B.V. All rights reserved.

use its performance as the evaluation criterion to select features, such as the correlation-based feature selection method (CFS) [23], support vector machine recursive feature elimination (SVM-RFE) [24] and others [25,26]. Different from the embedded and the wrapper approaches, the filter model assesses the importance or relevance of features by considering only the intrinsic properties of the data, without involving any learning algorithm. Typical filter-type feature selection methods include $Q-\alpha$ [27], Laplacian score [28], information gain (IG) [29], Fisher score [4], relief-F [30], and minimal-redundancy–maximal-relevance (mRMR) [31]. Up to date improvements of filter model include Fisher–Markov selector [32], quadratic programming feature selection [33] and the fast clustering-based feature selection algorithm (FAST) [34]. Comparably speaking, the embedded and wrapper methods include interaction with the learning algorithm, and thus tend to achieve better results than filter methods; while filter model has better generality, and is usually computationally less expensive than embedded and wrapper methods, which makes filter model a good choice when the number of feature is large. In this paper, we will focus on the filter methods for supervised feature selection.

In filter methods for feature selection, Fisher score is one of the most widely used supervised methods due to its general good performance. It evaluates each individual feature separately and then selects the top-d ranked features with high scores. However, it does not take account of the feature dependence, which leads to a suboptimal subset of features. Considering the correlation of features, the sequential search methods have been proposed and proved to be very effective [7], such as the sequential forward search strategy, sequential backward search strategy, and plus-L minus-R selection. Besides, in order to select a subset of features simultaneously rather than selecting each feature individually, Gu et al. [35] generalized the traditional Fisher score and proposed a feature selection method in the form of a quadratically constrained linear programming. However, the computational complexity of these methods is prohibitive for high dimensional data.

In this paper, we propose a filter-type sequential supervised feature selection method based on the generalized Fisher criterion and an adaptive eigenspace model. Specifically, in the proposed method, the adaptive eigenspace model is utilized to update the generalized Fisher score with new features, and informative features are selected incrementally, i.e., the sequential forward search manner. The contribution of proposed incremental supervised feature selection method covers the following aspects:

1. *It can naturally deal with the Small Size Sample problem*: In the proposed method, the generalized Fisher score is computed by the pseudoinverse of the total scatter matrix, thus it can still be got even if the total scatter matrix is singular.
2. *It finds the desired feature subset efficiently*: The proposed method utilizes an adaptive eigenspace model to approximate the generalized Fisher score. For each candidate feature, the time consumed for computing its generalized Fisher score depends on the dimension of the eigenspace model rather than the number of selected features, where the dimension is always lower than the number of selected features. If the dimension of the eigenspace is fixed, our method consumes nearly constant time to compute the Fisher score of a feature, which is much more efficient than traditional sequential methods.

The rest of this paper is structured as follows: In Section 2, we provide a brief review of the Fisher score and the related sequential search strategies for feature selection. In Section 3, the proposed sequential supervised feature selection method is presented. Experimental results on benchmark databases are shown in Section 4. Finally, we conclude the paper in Section 5.

2. A brief review of Fisher score and sequential search strategies

In this section, we first establish our notations for the rest of the paper. Then, we briefly review Fisher score and its related methods based on sequential search strategies.

2.1. Notations

In order to avoid confusion, we give a list of the main notations used in this paper, in Table 1. Throughout this paper, all data points are in the form of column vectors and all feature vectors are in the form of row vectors. Both the data points and feature vectors are denoted by lowercase. All sets are represented by capital curlicue letters. Matrices are denoted by normal capital letters.

2.2. Fisher score

The key idea of Fisher score is to find a subset of features which maximizes the inter-class dispersion and the intra-class compactness simultaneously. Specifically, given the selected d features, the input data matrix $Z\in\mathbb{R}^{D\times N}$ is reduced to be $X\in\mathbb{R}^{d\times N}$. Then the Fisher score of X is defined as

$$F(X)=\mathrm{tr}\left\{(S_t)^{-1}S_b\right\}. \tag{1}$$

Here, $\mathrm{tr}(\cdot)$ is the trace of a matrix, S_b is the between-class scatter matrix, and S_t is the total scatter matrix. S_b and S_t are defined as

$$S_b=\sum_{i=1}^{C}n_i(m_i-m)(m_i-m)^T,$$
$$S_t=\sum_{j=1}^{N}(x_j-m)(x_j-m)^T, \tag{2}$$

where $m_i=(1/n_i)\sum_{j\in\{i\}}x_j$ is the mean of ith class and $m=(1/N)\sum_{j=}$ $1^N x_j$ is the global mean of the input data.

Roughly speaking, feature selection based on Fisher score is a challenging combinatorial optimization problem as there are $\binom{D}{d}$ candidates X out of Z. For this problem, the widely used heuristic strategies include the single feature evaluation method and the methods based on sequential search strategies.

In the single feature evaluation method, there are only D candidates (each candidate is a single feature) to be scored, and then the top-d ranked features are selected. In detail, let m_i^k and σ_i^k be the mean and the standard deviation, respectively, of the ith class, corresponding to the kth feature f_k. Let m^k and σ^k denote the mean and the standard deviation, respectively, of the whole dataset, corresponding to the kth feature f_k. It is easy to verify that $(\sigma^k)^2=\sum_{i=1}^{C}n_i(\sigma_i^k)^2$. Then the Fisher score of the kth feature f_k is computed as

$$F(f_k)=\frac{\sum_{i=1}^{C}n_i(m_i^k-m^k)^2}{(\sigma^k)^2},\quad k=1,\dots,D. \tag{3}$$

After scoring all the features, the top-d ranked features with large scores are selected as the desired feature subset. Totally speaking, the computation complexity of single feature evaluation method is low. However, the combination of individually good features does not necessarily lead to a good feature set, so the selected feature set is always suboptimal.

2.3. Sequential search strategies

Different from the single feature evaluation method which ignores the mutual information among features, sequential feature selection methods evaluate unselected features by referring to the

Please cite this article as: N. Gu, et al., Efficient sequential feature selection based on adaptive eigenspace model, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.043

Table 1
Notations.

D	The dimension of input data,i.e. the number of all features of input data.
N	The number of data points.
Z	$Z = [z_1, \ldots, z_N] \in \mathbb{R}^{D \times N}$ is the input data matrix. Each column $z_j \in \mathbb{R}^D$ denotes a data point,for $j = 1, \ldots, N$.
C	The total number of classes that the input data belong to.
$\{i\}$	The index set of the ith class,where $i = 1, \ldots, C$.
n_i	The number of data points in the ith class,where $i = 1, \ldots, C$.
f_k	$f_k \in \mathbb{R}^N$ is the kth feature vector of data ($k = 1, \ldots, D$). It is also the kth row of the data matrix Z, i.e., $Z = [f_1^T, \ldots, f_D^T]^T$.
$\mathcal{S}$	The set of currently selected features.
$\mathcal{U}$	The set of currently unselected features.
d	The number of features in $\mathcal{S}$.
X	$X = [x_1, \ldots, x_N] \in \mathbb{R}^{d \times N}$ is the reduced data matrix which consists of certain rows (corresponding to features in $\mathcal{S}$) of Z.
$\tilde{X}$	$\tilde{X} = \begin{bmatrix} f \\ X \end{bmatrix} \in \mathbb{R}^{(d+1) \times N}$ is the concatenation of an unselected feature $f \in \mathcal{U}$ and X.
$\hat{X}$	$\hat{X} \in \mathbb{R}^{(d-1) \times N}$ represents the reduced data matrix, which consists of features in $\mathcal{S}$ ÄÈaÏ¿ $\{f\}$.

selected features and thus can handle redundant features more effectively. Typical sequential feature selection strategies include sequential forward selection (SFS), sequential backward selection (SBS), and plus-L minus-R selection (LRS).

SFS selects the features based on the sequential forward search strategy. It starts from the feature with the highest Fisher score. Then, the method sequentially adds a currently unselected feature $f \in \mathcal{U}$ which maximizes Fisher score $F(\tilde{X})$. Here, $\tilde{X} = \begin{bmatrix} f \\ X \end{bmatrix}$ is the concatenation of the currently unselected feature vector f and the currently selected feature matrix X. SFS method has been proved to be effective in selecting features with considering their mutual information. The main disadvantage of SFS is that it is unable to remove features that become obsolete after the addition of other features.

SBS works in the opposite direction of SFS. It starts from the input data Z, and sequentially removes the feature $f \in \mathcal{S}$ that least reduces the value of the Fisher score $F(\hat{X})$, where $\hat{X}$ is the reduced data matrix which consists of features in $\mathcal{S} \backslash \{f\}$. The main limitation of SBS is its inability to reevaluate the usefulness of a feature after it has been discarded.

LRS can be considered as an integration of the SFS and SBS methods, where L and R are both integers. If $L > R$, LRS starts from the empty set and repeatedly adds L features and then removes R features, otherwise when $L < R$, LRS starts from the input data Z and repeatedly removes R features followed by L additions. LRS attempts to compensate for the weaknesses of SFS and SBS with some backtracking capabilities.

SFS, SBS and LRS are computationally expensive in practice. For SFS method, evaluating a candidate feature in $\mathcal{U}$ needs $O(\min(d+1, N)^3)$ computational time, where d is the number of features that have been selected. Therefore, SFS consumes $O\Big((D-d)(\min(d+1, N)^3)\Big)$ computational time to select a feature from $\mathcal{U}$ and add it to $\mathcal{S}$. The total computational time becomes considerably huge as d increases. On the other hand, SBS method consumes $O(\min(d-1, N)^3)$ time to evaluate a candidate feature from $\mathcal{S}$. In total, SBS needs $O\Big(d(\min(d-1, N)^3)\Big)$ computational time to delete a redundant feature from $\mathcal{S}$. The computational complexity of SBS is also high, since the deletion process is started from the total feature set, where $d = D$. The computational complexity of LRS is the highest because it not only evaluates and selects candidate features from $\mathcal{U}$ as the SFS method, but also removes features from $\mathcal{S}$ as the SBS method.

3. The efficient sequential feature selection method

In this section, we first present the generalized Fisher score to measure the importance of features. Then, we discuss how to drop the trivial eigenvalues and eigenvectors of the total scatter matrix. Thirdly, we propose the adaptive eigenspace model for the incremented generalized Fisher score. At last, we present the efficient sequential feature selection method based on the generalized Fisher score and the adaptive eigenspace model.

3.1. The generalized Fisher score

For the reduced data matrix $X \in \mathbb{R}^{d \times N}$, we define the between-class matrix H_b and the total matrix H_t as

$$H_b = [\sqrt{n_1}(m_1 - m), \ldots, \sqrt{n_C}(m_C - m)],$$
$$H_t = X - m\mathbf{1}^T, \tag{4}$$

where n_i is the number of points in the ith class, C is the total number of classes, $\mathbf{1}$ is the column vector whose elements are all ones, m_i and m are defined as in Eq. (2). Then, the between-class scatter matrix S_b and the total scatter matrix S_t of X can be expressed as $S_b = H_b H_b^T$ and $S_t = H_t H_t^T$.

A serious disadvantage of Fisher score defined in Eq. (1) is that its objective function requires the total scatter matrix S_t to be nonsingular. However, such requirement is not always satisfied in modern data mining problems, especifically when the input data are very high dimensional. To address this matrix singularity problem, we make use of the generalized Fisher score [36] based on pseudoinverse.

Definition 1. The pseudoinverse of a matrix $A \in \mathbb{R}^{d \times N}$, which is denoted as $A^\dagger \in \mathbb{R}^{N \times d}$, refers to the unique matrix satisfying the following condition:

$$AA^\dagger A = A. \tag{5}$$

The commonly used technique to compute the pseudoinverse matrix $A^\dagger$ is singular value decomposition (SVD). Assume that the SVD of A is $A = U\Sigma V^T$, where $U \in \mathbb{R}^{d \times r}$ and $V \in \mathbb{R}^{N \times r}$ are unitary matrices, $\Sigma \in \mathbb{R}^{r \times r}$ is a diagonal matrix with diagonal elements $\Sigma(i,i) \geq 0 (i = 1, \ldots, r)$, and $r \leq \min\{d, N\}$ is the rank of matrix A. Then the pseudoinverse matrix $A^\dagger$ can be given as $A^\dagger = V\Sigma^{-1}U^T$ where Σ^{-1} is the diagonal matrix with diagonal elements:

$$\Sigma^{-1}(i,i) = \begin{cases} (\Sigma(i,i))^{-1} & \text{if } \Sigma(i,i) > 0 \\ 0 & \text{if } \Sigma(i,i) = 0 \end{cases}. \tag{6}$$

Considering the total scatter matrix S_t, there are two approaches to compute its pseudoinverse matrix $S_t^\dagger$:

(1) *Using the eigen-decomposition of S_t*: Since S_t is a positive semi-definite symmetric matrix, we can assume that its eigen-decomposition is $S_t = U\Xi U^T$, where U is an unitary matrix and Ξ is a diagonal matrix. The columns of U are eigenvectors

of S_t and the diagonal elements of Ξ are the corresponding nonnegative eigenvalues of S_t. Then, the pseudoinverse matrix of S_t can be given as $S_t^{\dagger} = U\Xi^{-1}U^T$.

(2) *Using the SVD of* H_t: Suppose that the SVD of H_t is $H_t = U\Sigma V^T$. From $S_t = H_t H_t^T$ we can get that the eigen-decomposition of S_t is $S_t = U\Sigma^2 U^T$ and its pseudoinverse matrix can be computed as $S_t^{\dagger} = U\Sigma^{-2}U^T$.

Replacing S_t^{-1} with $S_t^{\dagger}$ in Eq. (1), the generalized Fisher score is defined as

$$G(X) = \mathrm{tr}\left(S_t^{\dagger} S_b\right). \tag{7}$$

Then we can use this criterion to rank the importance of the feature subsets.

3.2. Dropping trivial eigenvalues and eigenvectors

Assume that the eigen-decomposition of S_t is

$$S_t = U\Sigma^2 U^T, \tag{8}$$

where the diagonal elements of Σ^2 are sorted in the decreasing order, $\Sigma^2(i,i)$ denotes the ith largest eigenvalue of S_t and the ith column of U is the corresponding eigenvector. Each eigenvalue reveals how much variance of the data can be explained by the associated eigenvector. For example, the largest eigenvalue indicates that the highest variance of the data can be observed in the direction of the corresponding eigenvector. Accordingly, if we take all eigenvectors together, we can explain all variance of the data.

Typically, only a fraction of the eigenvectors have significant eigenvalues and small eigenvalues are presumed negligible. Therefore, only a fraction of eigenvalues and eigenvectors need to be retained. This is common in practice as the features are correlated so that the covariance matrix can be well approximated by a lower rank matrix. Different methods for discarding trivial eigenvalues and eigenvectors exist and these suit different applications. Three methods are

1. Stipulate r as a fixed integer and simply keep the r largest eigenvalues.
2. Keep the eigenvalues whose sum takes a specified fraction of energy in the eigenspectrum (computed as the sum of eigenvalues).
3. Keep the eigenvalues which are larger than a absolute threshold.

As to the problem of which is the best method to discard trivial eigenvalues and eigenvectors (remove data noise), there should be no permanent winner of the three methods. Theoretically, if one wants to know which dropping method is the best for a dataset, he should know information about the quantity and the type of the noise.

Once chosen the significant eigenvalues and eigenvectors of S_t, we can change the column order of U and Σ in Eq. (8) such that Eq. (8) can be reformulated as

$$\begin{aligned} S_t = U\Sigma^2 U^T &= [U_r, U_{D-r}]\begin{pmatrix} \Sigma_r^2 & 0 \\ 0 & \Sigma_{D-r}^2 \end{pmatrix}[U_r, U_{D-r}]^T \\ &= U_r\Sigma_r^2 U_r^T + U_{D-r}\Sigma_{D-r}^2 U_{D-r}^T \approx U_r \Sigma_r^2 U_r^T. \end{aligned} \tag{9}$$

Here, r is the number of eigenvalues that are kept, the diagonal elements of Σ_r^2 are the kept eigenvalues, the columns of U_r are the corresponding eigenvectors, Σ_{D-r}^2 and U_{D-r} are those discarded. In Eq. (9), the error $U_{D-r}\Sigma_{D-r}^2 U_{D-r}^T$ is small as the diagonal elements of Σ_{D-r}^2 are very small. Without lose of generality, we write $U = U_r$ and $\Sigma^2 = \Sigma_r^2$ in the rest part of this paper.

3.3. The adaptive eigenspace model for incremented generalized fisher score

For SFS method, it sequentially adds a feature f^*, which is in the currently unselected feature subset $\mathcal{U}$, to the currently selected feature subset $\mathcal{S}$. Specifically, in each step, SFS evaluates each feature $f \in \mathcal{U}$ by computing the Fisher score $F(\tilde{X}_f)$, where $\tilde{X}_f = [f^T, X^T]^T$ and X is the reduced data matrix corresponding to $\mathcal{S}$. Then $f^* = \arg\max_{f \in \mathcal{U}} F(\tilde{X}_f)$ is added to $\mathcal{S}$ and removed from $\mathcal{U}$. Denote d as the number of features in $\mathcal{S}$, then computing the Fisher score $F(\tilde{X}_f)$ for a candidate feature f needs $O(\min(d+1,N)^3)$ computational time, and selecting a favorite feature from $\mathcal{U}$ and adding it to $\mathcal{S}$ consume $O\left((D-d)(\min(d+1,N)^3)\right)$ computational time. Therefore, the total computational time of SFS becomes considerably huge when the number d of selected features increases to a certain extent. To deal with this problem, in this subsection we propose an adaptive eigenspace model to efficiently update the generalized Fisher score when a new feature f is added.

Let X be the reduced data matrix associated with the currently selected feature set $\mathcal{S}$, and $f \in U$ be a candidate feature where U is the currently unselected feature set. Then, the incremented data can be expressed as

$$\tilde{X} = \begin{bmatrix} f \\ X \end{bmatrix}. \tag{10}$$

We define the between-class feature vector and total feature vector of f as

$$\begin{aligned} H_b^f &= [\sqrt{n_1}(m_1^f - m^f), \ldots, \sqrt{n_C}(m_C^f - m^f)] \in \mathbb{R}^{1\times C}, \\ H_t^f &= f - m^f \mathbf{1}^T \in \mathbb{R}^{1\times N}, \end{aligned} \tag{11}$$

where m_i^f denotes the mean of the values in f corresponding to the ith class and m^f denotes the mean of all values in f. Then, we have the incremented between-class matrix and total matrix for incremented data $\tilde{X}$:

$$\tilde{H}_b = \begin{pmatrix} H_b^f \\ H_b \end{pmatrix}, \quad \tilde{H}_t = \begin{pmatrix} H_t^f \\ H_t \end{pmatrix}. \tag{12}$$

Here, H_b is the between-class matrix of X, and H_t is the total matrix of X, as defined in (4). Then the between-class scatter matrix and the total scatter matrix of $\tilde{X}$ can be got as

$$\tilde{S}_b = \tilde{H}_b(\tilde{H}_b)^T, \quad \tilde{S}_t = \tilde{H}_t(\tilde{H}_t)^T. \tag{13}$$

Denote S_t as the total scatter matrix of X, and assume that the eigen-decomposition of S_t is given as

$$S_t \approx U\Sigma^2 U^T, \tag{14}$$

where $U \in \mathbb{R}^{d\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$, d is the number of features in $\mathcal{S}$, and $r < d$ denotes the number of eigenvalues kept in S_t (the $D-r$ trivial eigenvalues and eigenvectors are discarded, as previously discussed in Section 3.2). Then we can get the SVD of H_t as

$$H_t = U\Sigma V^T, \tag{15}$$

where $V = H_t^T U \Sigma^{-1}$.

We aim to accelerate the eigen-decomposition of $\tilde{S}_t = \tilde{H}_t(\tilde{H}_t)^T$ based on the eigen-decomposition of $S_t = H_t(H_t)^T$, then the incremented generalized Fisher score $G(\tilde{X}_f) = \mathrm{tr}\left(\tilde{S}_t^{\dagger}\tilde{S}_b\right)$ can be got in less computational time than SFS. To achieve this, our method consists of two processes. Firstly, we make use of the eigen-decomposition of $(H_t)^T H_t$ to compute the eigen-decomposition of $(\tilde{H}_t)^T\tilde{H}_t$. Secondly, we utilize the eigen-decomposition of $(\tilde{H}_t)^T\tilde{H}_t$ to get the eigen-decomposition of $\tilde{S}_t = \tilde{H}_t(\tilde{H}_t)^T$, and then compute the incremented generalized Fisher score.

Please cite this article as: N. Gu, et al., Efficient sequential feature selection based on adaptive eigenspace model, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.043

In the first process, we can eigen-decompose the matrix

$$(\tilde{H}_t)^T\tilde{H}_t = (H_t^f)^T H_t^f + (H_t)^T H_t, \tag{16}$$

where the eigen-decomposition of $(H_t)^T H_t$ is known as

$$(H_t)^T H_t = V\Sigma^2 V^T. \tag{17}$$

Assume that the eigen-decomposition of $(\tilde{H}_t)^T\tilde{H}_t$ is

$$(\tilde{H}_t)^T\tilde{H}_t = \tilde{V}\tilde{\Sigma}^2(\tilde{V})^T. \tag{18}$$

According to the eigenspace merging method proposed by Hall et al. [37], the eigenspace model of $(\tilde{H}_t)^T\tilde{H}_t$ can be spanned by the eigenspace model of $(H_t)^T H_t$ and the total feature vector H_t^f. Therefore, a sufficient spanning set Φ can be formed by columns of the unitary matrix V, and the residue of $(H_t^f)^T$ with respect to the eigenspace of $(H_t^f)^T H_t^f$. That is,

$$\Phi = [V, \mathrm{orth}\left((I - VV^T)(H_t^f)^T\right)] \in \mathbb{R}^{N\times(r+1)} \tag{19}$$

where orth() is the orthonormalization function, such as QR and SVD, followed by removal of zero vectors. This basis set Φ differs from the required eigenvectors, $\tilde{V}$, by a rotation matrix R, i.e.,

$$\tilde{V} = \Phi R. \tag{20}$$

Substituting (20) into (18), we can convert the eigenproblem of $(\tilde{H}_t)^T\tilde{H}_t$ to be a smaller one:

$$\begin{aligned}&(\tilde{H}_t)^T\tilde{H}_t = \tilde{V}\tilde{\Sigma}^2\tilde{V}^T \Longrightarrow \\ &\quad \Phi^T(\tilde{H}_t)^T\tilde{H}_t\Phi = R\tilde{\Sigma}^2 R^T\end{aligned} \tag{21}$$

By computing the eigen-decomposition of matrix $\Phi^T(\tilde{H}_t)^T\tilde{H}_t\Phi \in \mathbb{R}^{(r+1)\times(r+1)}$, $\tilde{\Sigma}^2$ and the rotation matrix R are obtained as the eigenvalue and eigenvector matrices respectively. Then the eigenvector matrix $\tilde{V}$ to seek is given as $\tilde{V} = \Phi R$.

Remark 1. If d, which is the number of features that have been selected, is smaller than $N-1$, then the eigen-analysis of $\tilde{S}_t = \tilde{H}_t(\tilde{H}_t)^T$ requires $O((d+1)^3)$ computations. Comparably, the eigen-analysis of Eq. (21) takes $O((r+1)^3)$ computations, where $r < d$ is the number of eigenvalues kept in S_t. Therefore, the complexity of the eigen-analysis in Eq. (21) is much lower than that of $\tilde{S}_t = \tilde{H}_t(\tilde{H}_t)^T$. If we apply the first eigenvalue dropping method introduced in Section 3.2, the number of kept eigenvalues is fixed and then the computational time for the eigen-decomposition process needs nearly constant time, which is irrelevant to the number of selected features.

Once we get the eigen-decomposition of $(\tilde{H}_t)^T\tilde{H}_t$ as

$$(\tilde{H}_t)^T\tilde{H}_t = \tilde{V}\tilde{\Sigma}^2(\tilde{V})^T, \tag{22}$$

we can go to the second process: getting the eigen-decomposition of $\tilde{S}_t = \tilde{H}_t(\tilde{H}_t)^T$ and computing the incremented generalized Fisher score. Specifically, the left unitary matrix of $\tilde{H}_t$ can be got by

$$\tilde{U} = \tilde{H}_t\tilde{V}\tilde{\Sigma}^{-1}. \tag{23}$$

Then the SVD of $\tilde{H}_t$ is

$$\tilde{H}_t = \tilde{U}\tilde{\Sigma}(\tilde{V})^T, \tag{24}$$

and the eigendecomposition of the matrix $\tilde{S}_t = \tilde{H}_t(\tilde{H}_t)^T$ can be obtained as

$$\tilde{S}_t = \tilde{H}_t(\tilde{H}_t)^T = \tilde{U}\tilde{\Sigma}^2\tilde{U}^T. \tag{25}$$

Therefore, the incremented general Fisher score is

$$G(\tilde{X}) = \mathrm{tr}\left(\tilde{U}\tilde{\Sigma}^{-2}\tilde{U}^T\tilde{S}_b\right). \tag{26}$$

Remark 2. When an unselected feature arrives, the proposed method first computes the sufficient spanning set Φ by (19). Because H_t^f is a single feature vector, only one matrix-vector multiplication and one vector normalization are implemented at this step, which consume $O(1)$ computational time. Subsequently, the eigenproblem in (21) needs to be solved. The eigen-problem is of size $(r+1)\times(r+1)$, and thus it consumes $O((r+1)^3)$ time. At the final step, the generalized Fisher score is computed by (26), which consists of matrix multiplications and can be done in approximately $O(r+1)$ time. Therefore, the total computational time for evaluating an unselected feature is approximately $O(1) + O((r+1)^3) + O(r+1) = O((r+1)^3)$. If we fix the number r in the updating eigenspace, the computational time for evaluating an unselected feature needs nearly constant time.

3.4. The efficient sequential feature selection method

Based on the previously discussion, our Efficient Sequential Feature Selection (ES-FS) method can be presented as follows:

Input: High dimensional input data matrix $Z = [z_1, \ldots, z_N]$; the maximum number, denoted as T, of features to select.

Step 1: Apply Fisher score method to evaluate each feature (row) of Z. Without lose of generality, denote f_1 and f_2 as the features that have the first and second highest scores, respectively. Let $\mathcal{S} = \{f_1, f_2\}$, $\mathcal{U} = \{f_3, \ldots, f_D\}$, $X = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ and $d=2$.

Step 2: Let $f \in \mathcal{U}$ be an unselected feature and $\tilde{X}$ be the concatenated matrix of f and X. We apply the incremental method proposed in Section 3.3 to compute the generalized Fisher score $G(\tilde{X})$.

Step 3: Run Step 2 until all features in $\mathcal{U}$ are evaluated when combined with X, and find the feature $f^* \in \mathcal{U}$ with highest generalized Fisher score. Then, $\mathcal{U} = \mathcal{U}\backslash\{f^*\}$, $\mathcal{S} = \mathcal{S}\bigcup\{f^*\}$, $d = d+1$ and $X = \begin{bmatrix} f^* \\ X \end{bmatrix}$.

Step 4: Drop the trivial eigenvalues and eigenvectors of the total scatter matrix for X, as presented in Section 3.2.

Step 5: If $d < T$, go to Step 2. Else if $d = T$, finish the feature selection process.

Remark 3. The proposed forward ES-FS method can be extended to SBS manner and LRS manner [37]. In SBS manner, we assume $X_f = \begin{bmatrix} f \\ \hat{X} \end{bmatrix}$ where $f \in \mathcal{S}$ is a candidate feature, and $\hat{X}$ refers to the reduced data matrix which is formed by features in $\mathcal{S}\backslash\{f\}$. In each step, the feature f^* that least reduces the generalized Fisher score $G(\hat{X})$ will be removed from $\mathcal{S}$. Denote H_t and $\hat{H}_t$ as the total matrices of X_f and $\hat{X}$ respectively, and the SVD of H_t is got as $H_t = U\Sigma V^T$. From

$$(H_t)^T H_t = V\Sigma^2 V^T = (H_t^f)^T H_t^f + (\hat{H}_t)^T\hat{H}_t \tag{27}$$

and

$$V^T(\hat{H}_t)^T\hat{H}_t V = \Sigma^2 - V^T(H_t^f)^T H_t^f V \tag{28}$$

we can convert the eigenproblem of $(\hat{H}_t)^T\hat{H}_t$ to that of $V^T(\hat{H}_t)^T\hat{H}_t V$, which is a smaller one. Then the eigen-decomposition of $\hat{S}_t = \hat{H}_t(\hat{H}_t)^T$ can be obtained, and the generalized Fisher score $G(\hat{X}) = \mathrm{tr}\left(\hat{S}_t^\dagger\hat{S}_b\right)$ can be computed efficiently. Subsequently, we can easily get the efficient LRS method by integrating the proposed efficient forward and backward search methods.

Remark 4. Different from the single feature evaluation method which computes the Fisher score of each feature individually, the proposed ES-FS sequentially adds a currently unselected feature $f \in \mathcal{U}$ which maximizes the incremented generalized Fisher score $G(\tilde{X})$. Therefore, the Fisher scores of the feature sets gained by the proposed ES-FS should be higher than that of single feature evaluation method. Compared with SFS method, the proposed

Please cite this article as: N. Gu, et al., Efficient sequential feature selection based on adaptive eigenspace model, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.043

Table 2
The description of experimental datasets.

Data sets	Data points	Features	Classes
Altkom	1200	2576	80
BANCA	520	2576	52
MPEG7	3175	2576	635
CBCL	2000	361	2
ORL	400	1024	40
CMUPIE	11,554	1024	68

ES-FS method drops trivial eigenvalues during the feature selection process. Specifically, suppose that the SVD of total scatter matrix S_t can be reformulated as (9), then the generalized Fisher score of SFS is $G(X)=\mathrm{tr}\left(S_t^{\dagger}S_b\right)=\mathrm{tr}\left(U\Sigma^{-2}U^TS_b\right)$, while that of ES-FS is $\tilde{G}(X)=\mathrm{tr}\left(U_r\Sigma_r^{-2}U_r^TS_b\right)$. Therefore, ES-FS produces lower generalized Fisher scores than SFS.

4. Experiments

In this section, we implement the face recognition experiments on six public available face databases, to evaluate the proposed method in the sense of computational time and classification accuracy.

4.1. Experimental datasets

Six publicly available face recognition datasets, namely, Altkom, BANCA and MPEG7,[1] MIT CBCL,[2] CMUPIE[38] and ORL [39]’[3] are used for experimental evaluation. The brief description of the datasets is shown in Table 2.

The Altkom face database contains 1200 face images of 80 persons (15 images for each person), the BANCA face database consists of 520 face images of 52 persons (10 images for each person), and the MPEG7 face database has 3175 face images of 635 persons (5 images for each person). All images in these three database are normalized to 46 × 56 pixels using manually labeled eye positions. Some samples from Altkom, BANCA, and MPEG7 databases are shown in Figs. 1, 2, and 3 respectively.

The MIT CBCL database contains 2429 face images and 4548 non-face images. Each image has 19 × 19 pixels and is transformed into a 361-dimensional vector. This database contains two classes of data points: face and non-face. In the experiment, we use a subset of this database, which consists of 1000 face and 1000 non-face images. Fig. 4 shows some face and non-face images from this dataset.

The ORL face database contains 400 images which belong to 40 distinct persons (10 images for each person). For some persons, the images were taken at different times, varying the lighting, facial expressions (open or closed eyes, smiling or not smiling) and facial details (glasses or no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Each image from this database is resized to 32 × 32 pixels. Some images from this database are shown in Fig. 5.

CMUPIE face database consists of 41,368 images of 68 persons, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. The applied dataset only contains five near frontal poses (C05, C07, C09, C27, C29) and all the images under different illuminations and expressions. So, there are totally 11,554 face images and 170 images for each individual. Each image is resized to 32 × 32 pixels. Fig. 6 shows some images from this database.

For all experimental datasets, each image is reshaped into a long vector. For example, an image of 46 × 56 pixels is transformed to be a 2576-dimensional vector. Subsequently, each vector is normalized to have unit l^2-length, denoted as z_i $(i=1,\ldots,N)$. Then the matrix $Z=[z_1,\ldots,z_N]\in\mathbb{R}^{D\times N}$ is used as the input data matrix of the algorithms.

4.2. Experimental settings

In the experiments, the following eight approaches are to be compared: the method using all features (denoted by All-FS), single feature evaluation method based on Fisher score (denoted by Single-FS), mRMR(MID) and mRMR (MIQ) [31], SPEC [40], Relief-F [30], SFS, and the proposed ES-FS method. After selecting the features using the above approaches, the 1-Nearest Neighbor (1-NN) classifier is utilized to compare the classification accuracies.

The whole experiment is conducted as follows. For each dataset, 10-fold cross validation is employed for evaluation. In each fold, d features are selected by each of the compared algorithms using the training set, and then the 1-NN classification accuracy for the test set, based on the d selected features, can be computed. We vary the values of d, and the corresponding classification accuracies can be obtained. Then we utilize 10-fold cross validation to repeat the above process for 10 times, and the results are averaged. In this way, for each dataset, we finally get the average classification accuracies of the test set under different dimension d.

For the proposed ES-FS method, we have to decide how to discard the trivial eigenvalues and eigenvectors, as stated in the second paragraph of Section 3.2. In the experiments, we utilize the following two strategies:

1. Stipulate r as a fixed integer and simply keep the r largest eigenvalues. Here we take r as $r_1=20$, $r_2=50$, $r_3=80$, $r_4=120$.
2. Keep the eigenvalues such that their sum takes a specified fraction, denoted as θ, of energy in the eigenspectrum. Here we take θ as $\theta_1=0.9$, $\theta_2=0.95$, $\theta_3=0.99$, $\theta_4=0.999$.

Then we select the r_i $(i=1,\ldots,4)$ or θ_i $(i=1,\ldots,4)$, with which the proposed ES-FS method has the best performance.

4.3. Experimental results

For the six datasets, the 1-NN classification results (i.e., the average results of 10-fold cross validation) of the compared algorithms on the test sets are shown in Fig. 7, with the number d of selected features changing.

(1) *Altkom dataset*: Fig. 7(a) shows the classification results on Altkom dataset. As can be seen, the proposed ES-FS method has obvious advantage over the rest methods. However, the accuracy curve of ES-FS algorithm turns to decline when the number of features becomes relatively large. This is because that, for many kinds of high dimensional data, the features are redundant, noise contaminated, and highly correlated. As the number of selected features increases, the proposed ES-FS method will bring noise contaminated or redundant features because no better features left. In a word, more features do not necessarily lead to better performance.

[1] Altkom, BANCA and MPEG7 databases are downloaded from http://www.iis.ee.ic.ac.uk/icvl/code.htm

[2] MIT CBCL database is downloaded from http://cbcl.mit.edu/software-datasets

[3] CMUPIE and ORL databases are downloaded from http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

Please cite this article as: N. Gu, et al., Efficient sequential feature selection based on adaptive eigenspace model, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.043

Fig. 1. Some images from the Altkom database.

Fig. 2. Some images from the BANCA database.

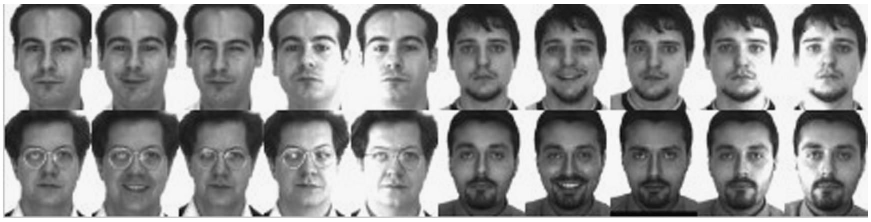


Fig. 3. Some images from the MPEG7 database.

Fig. 4. Some images from the CBCL database.

Fig. 5. Some images from the ORL database.

Fig. 6. Some images from the CMUPIE database.

(2) *BANCA dataset*: The experimental results on BANCA dataset are shown in Fig. 7(b). It can be seen that ES-FS method outperforms the compared methods. Besides, the accuracies of SFS method decline with the increase of the number of selected features. When the number of selected features is larger than 450, Single-FS method and Relief-F method show similar performances and both of them outperform SFS method. Besides, mRMR (MID) method performs better than mRMR(MIQ) method.

(3) *MPEG7 dataset*: In Fig. 7(c), we can see that ES-FS and Single-FS methods show significantly better results than the rest methods. When the number of features becomes relatively large, ES-FS outperforms Single-FS method. For each of the compared methods, when the number of features increases, the classification accuracies remain approximately constant. This may be because that the key information of data can be well represented using less than 650 features, and more features cannot benefit the classification performance.

(4) *CBCL dataset*: Fig. 7(d) shows the experimental results on CBCL dataset. As can be seen, more features lead to higher accuracies on this dataset. All-FS method obtains the highest

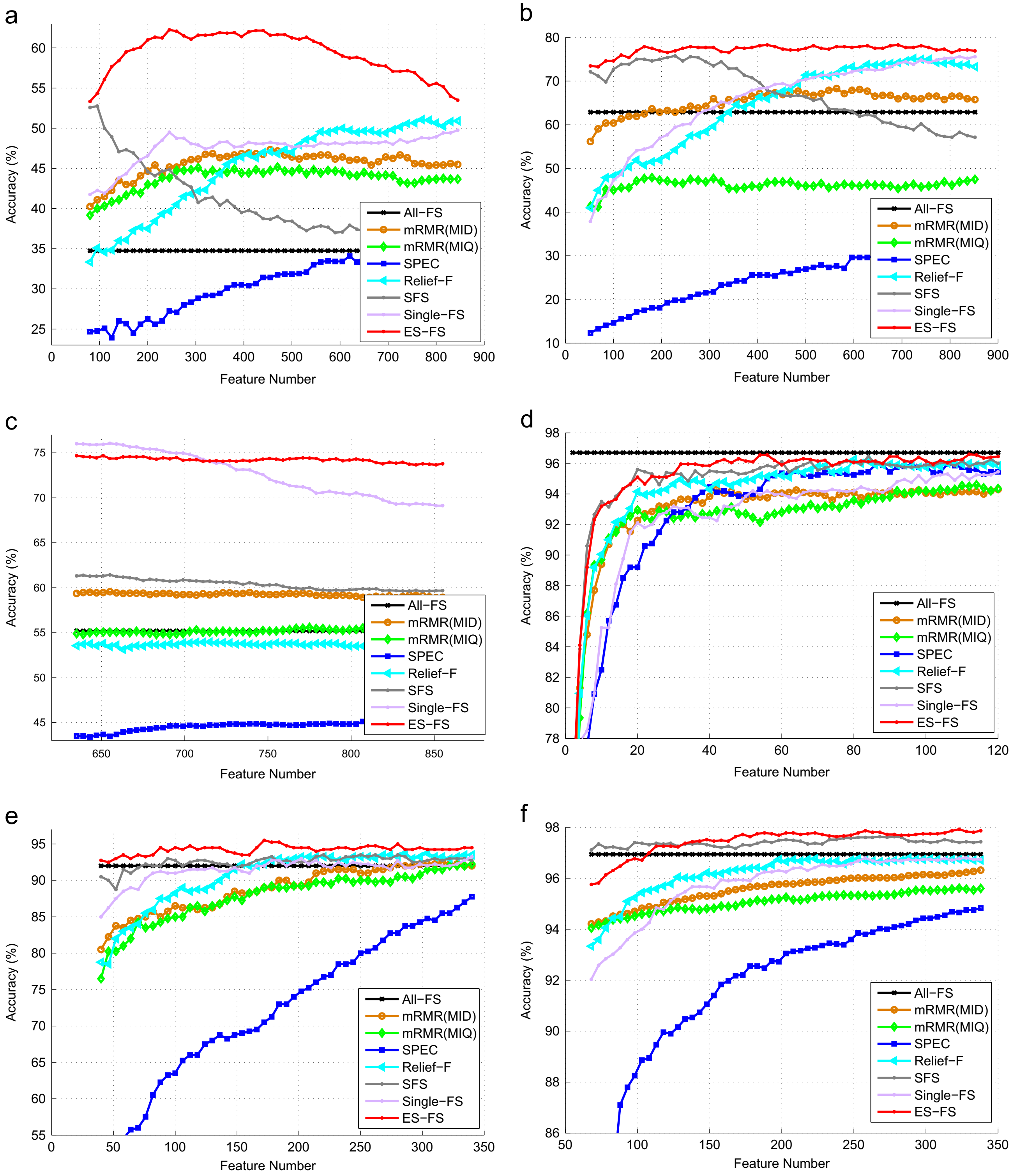


Fig. 7. Classification accuracies of the compared algorithms on (a) Altkom, (b) BANCA, (c) MPEG7, (d) CBCL, (e) ORL, (f) CMUPIE datasets.

classification results, and ES-FS has slightly better results than SFS method. All-FS, ES-FS, SFS, SPEC, Relief-F methods show similar results when the number of selected features is higher than 100.

(5) *ORL dataset*: As can be seen in Fig. 7(e), ES-FS outperforms the compared methods on ORL dataset. The ES-FS, SFS and Relief-F methods outperform All-FS method which utilizes all features of data. It means that many features of this dataset are redundant, and 50–100 features will be sufficient for 1-NN method to achieve satisfactory classification results.

(6) *CMUPIE dataset*: The performances of ES-FS and SFS methods are very close on this dataset and both of them outperform All-FS (i.e., 1-NN method with all features), as illustrated in Fig. 7(f). Single FS and Relief-F have similar accuracies, which are close to the accuracy of All-FS when the number of selected features is relatively large.

From Fig. 7 we can see that, generally speaking, the proposed ES-FS method exhibits comparable classification performance with or better performance than SFS method. This is because that the proposed ES-FS method drops trivial eigenvalues and eigenvectors

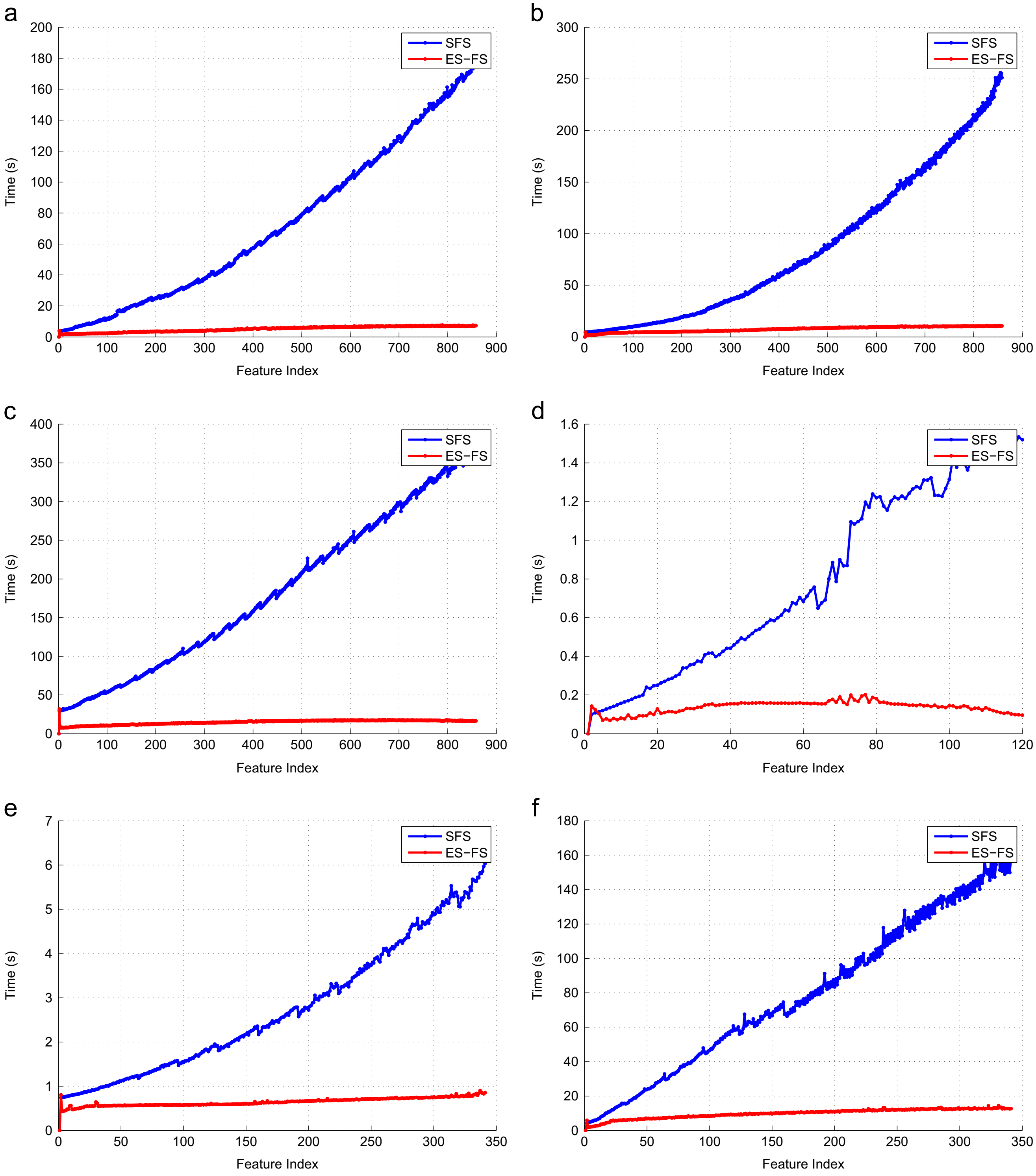


Fig. 8. Comparison of time consumption for SFS and the proposed ES-FS on (a) Altkom, (b) BANCA, (c) MPEG7, (d) CBCL, (e) ORL, (f) CMUPIE datasets. The less the better.

during the feature selection process, which makes ES-FS to filter out certain noise contained in the datasets. Fig. 7 shows that the incremental adaptive eigenspace model in ES-FS brings little error to the model, and the strategy (dropping trivial eigenvalues and eigenvectors) can improve the robustness of the proposed method on some datasets. Besides, from Fig. 7 we can see that the proposed ES-FS method has superiority over Single-FS method. This is because that Single-FS ignores the mutual information among features, while the proposed ES-FS method evaluates unselected features by taking account of the selected features and thus can handle redundant features more effectively.

Both of the Single-FS and SFS methods select features based on the Fisher criterion. SFS takes account of the feature dependence while Single-FS does not, which means that the Fisher score of SFS should be higher than that of Single-FS. However, from Fig. 7 we can see that sometimes Single-FS outperforms SFS, which seems puzzled. This is because that “Fisher score” and “classification accuracy” are two different criterions to evaluate the features. Generally speaking, higher

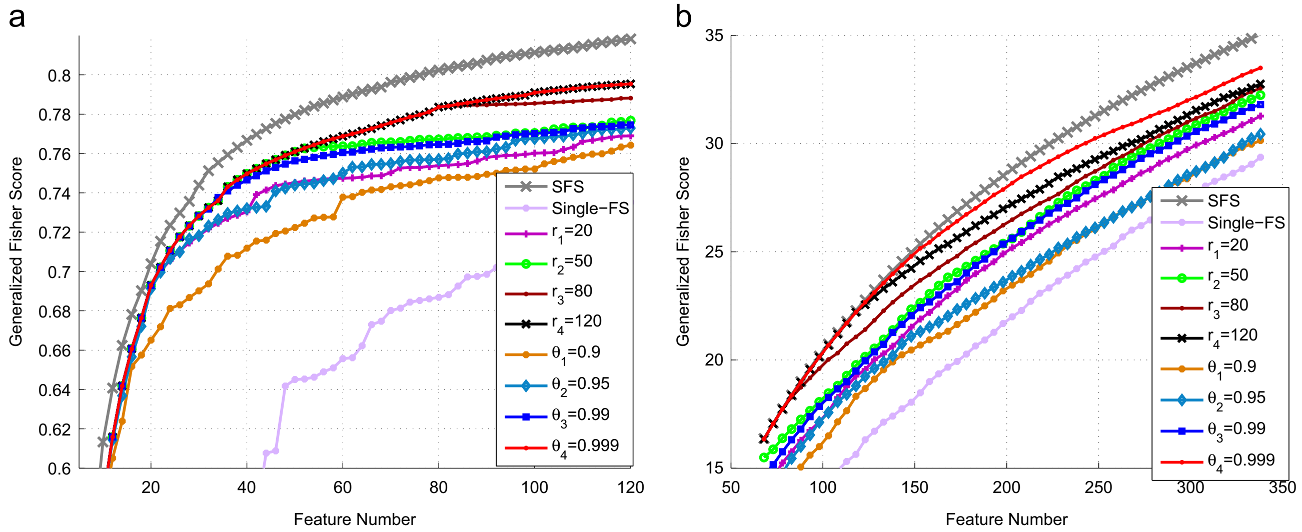


Fig. 9. The generalized Fisher scores of the feature sets got by SFS, Single-FS, and the proposed ES-FS method under different parameters. (a) CBCL dataset; (b) CMUPIE dataset.

Fisher score means better representation of data and thus leads to higher classification accuracy. However, they are not always consistent, i.e., one feature set that produces higher Fisher score possibly has lower classification accuracy than the other feature set.

Fig. 8 shows the time consumed for selecting the kth feature using SFS method and the proposed ES-FS method with the feature index k changing. From the figure we can see that the time consumed by the proposed ES-FS method increases much slower than SFS method when the feature index k becomes larger and larger. As a result, ES-FS method runs much faster than SFS method.

Fig. 9 presents the generalized Fisher scores of the feature sets got by SFS, Single-FS and the proposed ES-FS method, with the number of selected features changing. In ES-FS method, we use two different strategies to discard the trivial eigenvalues and eigenvectors, as stated in the last paragraph of Section 4.2, with different settings of parameters. From the figures we can see that the generalized Fisher scores produced by the proposed ES-FS method is higher than that of Single-FS, but lower than that of SFS. When r_i and θ_i become larger, the gained generalized Fisher scores become higher correspondingly. Besides, though we utilize two kinds of methods to discard trivial eigenvalues and eigenvectors, we find no permanent winner of the methods.

5. Conclusion

In this paper, we proposed an efficient sequential feature selection method. In the proposed method, the generalized Fisher score is utilized to measure the importance of features, which is computed by the pseudoinverse of total scatter matrix and thus can naturally deal with small size sample problem. Besides, the proposed method makes use of an adaptive eigenspace model to update the generalized Fisher score, then the method can incrementally select features based on the sequential forward search strategy with nearly constant time for a new feature. Experimental results on six benchmark face recognition datasets have shown the efficacy of the proposed method.

Acknowledgments

This work is supported in part by the NSFC Tian Yuan Special Foundation under Grant 11426159; in part by the NSFC under Grants 61203241, 61473212, 61472285 and 61305035; in part by the NSF of Zhejiang Province under Grant LQ13F030009; and in part by the 2013 Scientific Research Foundation of Capital University of Economics and Business under Grant 00591465730123.

References

[1] C.X. Ren, D.Q. Dai, X.X. Li, Z.R. Lai, Band-reweighed Gabor kernel embedding for face image representation and recognition, IEEE Trans. Image Process. 23 (2) (2014) 725–740.

[2] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.

[3] M. Shah, M. Marchand, J. Corbeil, Feature selection with conjunctions of decision stumps and learning from microarray data, IEEE Trans. Pattern Anal. Mach. Intell. 34 (1) (2012) 174–186.

[4] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience Publication, New York, USA, 2001.

[5] X. Yang, H. Qiao, Z.Y. Liu, Feature correspondence based on directed structural model matching, Image Vis. Comput. 33 (2015) 57–67.

[6] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, Springer, New York, USA, 2009.

[7] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[8] Z. Xu, I. King, M.R. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Netw. 21 (7) (2010) 1033–1047.

[9] W.J. Hua, K.S. Choic, Y.G. Gua, S.T. Wang, Minimum maximum local structure information for feature selection, Pattern Recognit. Lett. 34 (5) (2013) 527–535.

[10] X.F. He, M. Ji, C.Y. Zhang, H.J. Bao, A variance minimization criterion to feature selection using Laplacian regularization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (10) (2011) 2013–2025.

[11] X.W. Liu, L. Wang, J. Zhang, J.P. Yin, H. Liu, Global and local structure preservation for feature selection, IEEE Trans. Neural Netw. 25 (6) (2014) 1083–1095.

[12] Z.Y. Liu, H. Qiao, GNCCP—graduated nonconvexity and concavity procedure, IEEE Trans. Pattern Anal. Mach. Intell. 36 (6) (2014) 1258–1267.

[13] Z.Y. Liu, H. Qiao, X. Yang, C.H. Hoi, Graph matching by simplified convex-concave relaxation procedure, Int. J. Comput. Vis. 109 (2014) 169–186.

[14] P. Bradley, O. Mangasarian, Feature selection via concave minimization and support vector machines, in: Proceedings of the 15th International Conference on Machine Learning (ICML1998), 1998.

[15] Z.C. Li, Y. Yang, J. Liu, X.F. Zhou, H.Q. Lu, Unsupervised feature selection using nonnegative spectral analysis, in: Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI2012), 2012.

[16] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, X. Zhou, L21-Norm regularized discriminative feature selection for unsupervised learning, in: Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI2011), 2011, pp. 1589–1594.

[17] C.P. Hou, F.P. Nie, X.L. Li, D.Y. Yi, Y. Wu, Joint embedding learning and sparse regression: a framework for unsupervised feature selection, IEEE Trans. Cybern. 44 (6) (2014) 793–804.

[18] Z.C. Li, J. Liu, Y. Yang, X.F. Zhou, H.Q. Lu, Clustering-guided sparse structural learning for unsupervised feature selection, IEEE Trans. Knowl. Data Eng. 26 (9) (2014) 2138–2150.

[19] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2) (2004) 407–499.

[20] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. B 67 (2005) 301–320.

[21] S. Maldonado, R. Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Inf. Sci. 181 (2011) 115–128.

[22] S.M. Xiang, F.P. Nie, G.F. Meng, C.H. Pan, C.S. Zhang, Discriminative least squares regression for multiclass classification and feature selection, IEEE Trans. Neural Netw. Learn. Syst. 23 (11) (2012) 1738–1754.

[23] M.A. Hall, L.A. Smith, Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper, in: Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference, 1999.

[24] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (1) (2012) 389.

[25] J.B. Yang, C.J. Ong, Feature selection using probabilistic prediction of support vector regression, IEEE Trans. Neural Netw. 22 (6) (2011) 954–962.

[26] H. Zeng, Y.M. Cheung, Feature selection and kernel learning for local learning-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1532–1547.

[27] L. Wolf, A. Shashua, Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weight-based approach, J. Mach. Learn. Res. 6 (2005) 1855–1887.

[28] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Proceedings of Advances in Neural Information Processing Systems, vol. 18, 2005.

[29] L.E. Raileanu, K. Stoffel, Theoretical comparison between the gini index and information gain criteria, Ann. Math. Artif. Intell. 41 (1) (2004) 77–93.

[30] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman & Hall, Florida, USA, 2008.

[31] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.

[32] Q. Cheng, H.B. Zhou, 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 (6) (2011) 1217–1233.

[33] I. Rodriguez-Lujan, R. Huerta, C. Elkan, C. Santa Cruz, Quadratic programming feature selection, J. Mach. Learn. Res. 11 (2010) 1491–1516.

[34] Q.B. Song, J.J. Ni, G.T. Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, IEEE Trans. Knowl. Data Eng. 25 (1) (2013) 1–14.

[35] Q.Q. Gu, Z.H. Li, J.W. Han, Generalized Fisher score for feature selection, in: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI), Barcelona, Spain, 2011.

[36] J. Ye, R. Janardan, C.H. Park, H. Park, An optimization criterion for generalized discriminant analysis on undersampled problems, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 982–994.

[37] P. Hall, D. Marshall, R. Martin, Merging and splitting eigenspace models, IEEE Trans. Pattern Anal. Mach. Intell. 22 (9) (2000) 1042–1049.

[38] R. Gross, S. Baker, I. Matthews, Generic vs. Person specific active appearance models, Image Vis. Comput. 23 (11) (2005) 1080–1093.

[39] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, 1994, 138-142.

[40] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the 24th International Conference on Machine Learning (ICML2004), 2004.

Nannan Gu received the B.Sc. and M.Sc. degrees from Xi’an Jiaotong University, Xi'an, China, in 2006 and 2009 respectively. She received Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2012. She is currently a Lecturer in School of Statistics, Capital University of Economics and Business, Beijing, China. Her current research interests include semi-supervised classification, feature selection, and manifold learning.

Mingyu Fan received the B.Sc. degree from the Minzu University of China, Beijing, in 2006, and the Ph.D. degree from the Academy of Mathematics and System Science, Chinese Academy of Science, Beijing, China, in 2011. He is currently an Associate Professor with the Department of Mathematics, Wenzhou University. His current research interests include semi-supervised classification and feature learning.

Liang Du received the B.E. degree in Software Engineering from Wuhan University in 2007, and Ph.D degree in Computer Science from Institute of Software at University of Chinese Academy of Sciences in 2013. He is currently a Lecturer in Shanxi University. Prior to that, he was a Software Engineer at Alibaba Group between July 2013 and July 2014. His research interests include machine learning, data mining and big data analysis.

Dongchun Ren received the B.Sc. degree in automation from Nankai University, Tianjin, China, in 2008, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

He is currently a Researcher with Beijing 7Invensun Technology Co.,Ltd. His current research interest covers computer vision, pattern recognition, and machine learning.

Please cite this article as: N. Gu, et al., Efficient sequential feature selection based on adaptive eigenspace model, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.043