Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition 39 (2006) 1026–1033

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

A fast kernel-based nonlinear discriminant analysis for multi-class problems

Yong Xu[a, b], David Zhang[c,*], Zhong Jin[b], Miao Li[a], Jing-Yu Yang[b]

[a] *Bio-Computing Research Center and Shenzhen graduate school, Harbin Institute of Technology, Shenzhen, China*
[b] *Department of Computer Science & Technology, Nanjing University of Science & Technology, Nanjing, China*
[c] *The Biometrics Research Center and Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong*

Received 6 January 2004; received in revised form 12 October 2005; accepted 12 October 2005

Abstract

Nonlinear discriminant analysis may be transformed into the form of kernel-based discriminant analysis. Thus, the corresponding discriminant direction can be solved by linear equations. From the view of feature space, the nonlinear discriminant analysis is still a linear method, and it is provable that in feature space the method is *equivalent* to Fisher discriminant analysis. We consider that one linear combination of parts of training samples, called “significant nodes”, can replace the total training samples to express the corresponding discriminant vector in feature space to some extent. In this paper, an efficient algorithm is proposed to determine “significant nodes” one by one. The principle of determining “significant nodes” is simple and reasonable, and the consequent algorithm can be carried out with acceptable computation cost. Depending on the kernel functions between test samples and all “significant nodes”, classification can be implemented. The proposed method is called fast kernel-based nonlinear method (FKNM). It is noticeable that the number of “significant nodes” may be much smaller than that of the total training samples. As a result, for two-class classification problems, the FKNM will be much more efficient than the naive kernel-based nonlinear method (NKNM). The FKNM can be also applied to multi-class via two approaches: one-against-the-rest and one-against-one. Although there is a view that one-against-one is superior to one-against-the-rest in classification efficiency, it seems that for the FKNM one-against-the-rest is more efficient than one-against-one. Experiments on benchmark and real datasets illustrate that, for two-class and multi-class classifications, the FKNM is effective, feasible and much efficient.

Keywords: Fisher discriminant analysis; Kernel-based nonlinear discriminant analysis; Fast kernel-based nonlinear method; Pattern recognition; Face recognition; Feature extraction

1. Introduction

In the area of pattern recognition, kernel-based methods, e.g. kernel regression, kernel PCA and kernel Fisher discriminant analysis, have attracted much attention [1–10]. A number of applications of these methods are available. Generally, given an input space, a kernel-based method accomplishes its algorithm in a novel space (feature space) derived from the input space. Note that it does not need to actualize details of the corresponding transform by virtue of kernel functions. Thus, compared to ordinary nonlinear methods, kernel-based methods incur lower computational cost. However, when the naive kernel-based methods classify one test sample, all the kernel functions between all training samples and the test sample should be calculated in advance [1,2]. As a result, the efficiency of the methods declines while the number of the training samples increases. They will probably be very inefficient and even unfeasible for classification problems with a great number of training samples.

Obviously, it is significant to accelerate the classification process of the kernel-based methods. In practice some artifices have been proposed with the purpose of speeding up the methods [5,9,11]. The orthogonal least-squares algorithm [5] was adopted to improve kernel nonlinear discriminant analysis in classification efficiency. According to the algorithm,

* Corresponding author. Tel.: +852 2766 7271; fax: +852 2774 0842.
E-mail address: csdzhang@comp.polyu.edu.hk (D. Zhang).

0031-3203/$30.00 © 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2005.10.029

only a small quantity of training samples is exploited to classify test samples and the classification process is much more efficient. Suppose l_1 and l_2 are the numbers of samples in two classes, respectively. l/l_1 and $-l/l_2$ are taken as class labels of the two categories, respectively, by the algorithm. l is the total number of the samples. It is proved that, with class labels l/l_1 and $-l/l_2$, the (kernel) nonlinear discriminant analysis is directly related to the (kernel) Fisher discriminant analysis [5], a widely used discriminant analysis method [6,12,13,22]. However, without any regularization term, the orthogonal least-squares algorithm may lead to overfitting. The sparse kernel [11] is another method that is capable of classifying test samples efficiently. It is noticeable that the methods in Refs. [5,11] are only designed for two-class problems. Nevertheless, large quantities of multi-class classifications are needed in practice. So, it is also significant to accelerate the classification process of the multi-class kernel discriminant analysis. For support vector machine method, another kind of kernel method, many efforts have also been made to achieve more efficient algorithms [14,15].

In this paper, class labels of two categories are set as 1 and -1, respectively. It is proved that, from the view of feature space, the nonlinear discriminant analysis will be still *equivalent* to the Fisher discriminant analysis. After the nonlinear discriminant analysis is transformed into the form of a kernel method, an efficient and effective algorithm of selecting "significant" training samples is developed. Based on the algorithm, we can implement two-class classifications very efficiently. Moreover, the corresponding algorithm will generalize well with a regularization term. The method is called fast kernel-based nonlinear method (FKNM). In addition, two approaches, one-against-the-rest and one-against-one, are introduced to extend the FKNM into multi-class problems. If one-against-the-rest is combined into the FKNM, the method may be called FKNM with one-against-the-rest. If one-against-one is combined into the FKNM, the method will be called FKNM with one-against-one. Our experiments indicate that the FKNM with one-against-the-rest is superior to the FKNM with one-against-one in classification efficiency, which is distinct from the previous views on one-against-the-rest and one-against-one. The rest of the paper is organized as follows. The kernel-based nonlinear discriminant analysis is introduced in Section 2. Then the two-class FKNM is proposed in Section 3 and the multi-class FKNM is discussed in Section 4. Experimental procedures and results are shown in Section 5. Finally a concise conclusion is presented in Section 6.

2. Kernel-based nonlinear discriminant analysis

Classification tasks for two classes, c_1 and c_2, are discussed in this section. Suppose that a nonlinear function ϕ maps an input space into a high dimensional space F (feature space), and $x_1, x_2, \ldots, x_l$ are the samples of the input space while $\phi(x_1), \phi(x_2), \ldots, \phi(x_l)$ are the samples of the feature space. In addition, we assume that samples $x_1, x_2, \ldots, x_{l_1}$ are in class c_1 and samples $x_{l_1+1}, x_{l_1+2}, \ldots, x_l$ come from class c_2. The number of samples of c_1 and that of c_2 are l_1 and l_2, respectively $(l_1 + l_2 = l)$. 1 and -1 are regarded as class labels of class c_1 and class c_2, respectively.

To classify samples in space F, we construct the following formula:

$$\Phi W = B, \tag{1}$$

where

$$W = \begin{bmatrix} w_0 \\ w \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ \vdots \\ -1 \end{bmatrix}, \quad \Phi = \begin{bmatrix} 1 & \phi(x_1)' \\ 1 & \phi(x_2)' \\ \vdots & \\ 1 & \phi(x_l)' \end{bmatrix}. \tag{2}$$

We call w discriminant vector and call w_0 threshold. Here $\phi(x_i)$ can be regarded as one input, whereas the class label of x_i, 1 or -1, can be taken as corresponding output. If we deem it the base of classification to find the relation between one input and the corresponding output, Eq. (1) will play the role of achieving the relation. That is, Eq. (1) is devoted to finding W that connects the inputs and outputs well. If an input is transformed into a value very near to the corresponding output (1 or -1) by W, W will be qualified. In this sense, the least-squares solution of Eq. (1) is the optimal solution for classification. Obviously the solution can be determined by Eq. (3).

$$\Phi' \Phi W = \Phi' B. \tag{3}$$

Further more, the following equations can be derived from Eq. (3)

$$\begin{bmatrix} l & (l_1 m_1^\phi + l_2 m_2^\phi)' \\ l_1 m_1^\phi + l_2 m_2^\phi & \sum\limits_{i=1,\ldots,l} \phi(x_i)\phi(x_i)' \end{bmatrix} \begin{bmatrix} w_0 \\ w \end{bmatrix} = \begin{bmatrix} l_1 - l_2 \\ l_1 m_1^\phi - l_2 m_2^\phi \end{bmatrix}. \tag{4}$$

In the feature space the within-class scatter matrix S_w^ϕ can be evaluated by $S_w^\phi = \sum_{i=1,2} \sum_{x \in c_i} (\phi(x) - m_i^\phi)(\phi(x) - m_i^\phi)'$, where m_1^ϕ and m_2^ϕ are the means of class c_1 and class c_2, respectively. It is derivable that $\sum_{i=1,\ldots,l} \phi(x_i)\phi(x_i)' = S_w^\phi + l_1 m_1^\phi (m_1^\phi)' + l_2 m_2^\phi (m_2^\phi)'$. Substituting it into Eq. (4) yields

$$l S_w^\phi w + l_1 l_2 (m_1^\phi - m_2^\phi)[(m_1^\phi)' w - (m_2^\phi)' w] = 2 l_1 l_2 (m_1^\phi - m_2^\phi), \tag{5}$$

where $(m_1^\phi)' w$ and $(m_2^\phi)' w$ are two scalars. According to Eq. (5), the least-squares solution of Eq. (1) may be formulated by

$$w = a (S_w^\phi)^{-1} (m_1^\phi - m_2^\phi), \tag{6}$$

$$w_0 = \frac{1}{l}\left[l_1 - l_2 - (l_1 m_1^{\phi} + l_2 m_2^{\phi})' w\right], \tag{7}$$

where a is also a scalar. Note that, in the feature space, the Fisher discriminant vector is just in the form of $(S_w^{\phi})^{-1}(m_1^{\phi} - m_2^{\phi})$. Therefore, as a vector, the discriminant vector w is not different from the Fisher discriminant vector. That is, in the feature space, the discriminant analysis based on Eq. (1) is still consistent with the Fisher discriminant analysis. Another outstanding advantage of model Eq. (1) is that the subsequent classification will be very simple. The classification rule is as follows: if $w_0 + \phi(x)'w > 0$, then $x \in c_1$; otherwise $x \in c_2$.

According to the theory of reproducing kernels [6], w can be expanded in terms of

$$w = \sum_{i=1}^{l} \alpha_i \phi(x_i). \tag{8}$$

Consequently, Eq. (1) can be transformed into the following form:

$$KA = B, \tag{9}$$

where

$$A = \begin{bmatrix} w_0 \\ \alpha_1 \\ \vdots \\ \alpha_l \end{bmatrix}, K = \begin{bmatrix} 1 & k(x_1, x_1) & \ldots & k(x_1, x_l) \\ 1 & k(x_2, x_1) & \ldots & k(x_2, x_l) \\ \vdots & \vdots & & \vdots \\ 1 & k(x_l, x_1) & \ldots & k(x_l, x_l) \end{bmatrix}, \tag{10}$$

$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ and B is still defined by Eq. (2). We call A discriminant direction. The least-squares solution of Eq. (9) is given by

$$A = (K'K)^{-1}K'B. \tag{11}$$

However, because K is a l by $l+1$ matrix, its rank will be not greater than l. As a consequence, the rank of $K'K$ is less than or equal to l, although its size is $(l+1)\times(l+1)$. So $K'K$ is ill conditioned. To improve computationally numerical stability, we work out A by the following formula:

$$A = (K'K + \mu I)^{-1}K'B, \tag{12}$$

where μ is a positive constant and I is the identity matrix. In fact, Eq. (12) is also the optimal solution to optimization problem $\min\{obj(A)\}$, where $obj(A)$ is defined as

$$obj(A) = \mu A'A + (B - KA)'(B - KA). \tag{13}$$

Taking derivative of $obj(A)$ with respect to A, we have $\partial obj(A)/\partial A = 2\mu A - 2K'(B - KA)$. Equating the derivative to zero, we obtain $A = (K'K + \mu I)^{-1}K'B$. So, Eq. (12) can be taken as the solution to $\min\{obj(A)\}$. $\mu A'A$ in $obj(A)$ can be viewed as a regulation term, which can make the classification algorithm based on A generalize well. In other words, by virtue of optimization problem $\min\{obj(A)\}$, we can figure out the optimal solution of discriminant direction that performs and generalizes best in classification. In this paper, the above discriminant analysis model is called naive kernel-based nonlinear method (NKNM).

3. Fast kernel-based nonlinear method (FKNM)

The NKNM depends on all the kernel functions between all the training samples and one test sample to classify the test sample. So, the classification efficiency decreases while the number of the training samples increases. In order to make the classification process more efficient, we assume that the discriminant vector in the feature space can be approximately expressed by parts of the training samples, called “significant nodes”. In other words, we assume that w can be expanded in terms of

$$w = \sum_{i=1}^{r} \alpha_i^l \phi(x_i^0), \quad r < l, \tag{14}$$

where $x_1^0, x_2^0, \ldots, x_r^0$ are the “significant nodes” which are from the training sample set. Consequent kernel-based nonlinear model can be still formulated as

$$KA = B, \quad A = \begin{bmatrix} w_0 \\ \alpha_1^l \\ \vdots \\ \alpha_r^l \end{bmatrix},$$

$$K = \begin{bmatrix} 1 & k(x_1, x_1^0) & \ldots & k(x_1, x_r^0) \\ 1 & k(x_2, x_1^0) & \ldots & k(x_2, x_r^0) \\ \vdots & \vdots & & \vdots \\ 1 & k(x_l, x_1^0) & \ldots & k(x_l, x_r^0) \end{bmatrix}. \tag{15}$$

Here B is still defined as Eq. (2). After A is determined, the projection of test sample x onto the discriminant direction can be given as follows:

$$l_p(x) = w_0 + \sum_{i=1}^{r} \alpha_i^l k(x, x_i^0). \tag{16}$$

The following principle is proposed to determine “significant nodes”: the number of “significant nodes” should be as small as possible, while the value of $\mu\|A\|_2^2 + \|KA - B\|_2^2$ should be lower than a threshold. The meaning of the principle can be presented as follows: the less the number of the “significant nodes”, the more efficient consequent classification; the smaller $\mu\|A\|_2^2 + \|KA - B\|_2^2$, the better the discriminant direction in performing classification with appropriate capability of generalizing. There are two algorithms, the backward algorithm and the forward algorithm, to determine “significant nodes”. The main idea of the backward algorithm is that “non-significant nodes” are selected and eliminated one by one and the procedure does not end until a criterion is satisfied. All the remaining samples, which have not been eliminated yet, are taken as the “significant nodes”. Contrarily, the forward algorithm selects "significant nodes" one by one and the procedure is not terminated until one reasonable condition is satisfied. It is notable that, if the number of the training samples is very large, the backward algorithm will be very time consuming. In this section, the algorithm of determining "significant nodes" is developed as follows using the idea of the forward algorithm.

Step 1. Determine the first "significant node"

For the jth sample x_j $(j = 1, 2, \ldots, l)$, we construct the vector, $[k(x_1, x_j), k(x_2, x_j), \ldots, k(x_l, x_j)]'$, and call it "kernel vector" corresponding to the jth sample. Then, the matrix K corresponding to x_j is

$$K = \begin{bmatrix} 1 & k(x_1, x_j) \\ 1 & k(x_2, x_j) \\ \vdots & \vdots \\ 1 & k(x_l, x_j \end{bmatrix}. \tag{17}$$

A is worked out using Eq. (15) and is denoted by A_j. R_j is obtained by $R_j = [\mu\|A_j\|_2^2 + \|KA_j - B\|_2^2]^{1/2}$. After every x_j has been searched and all the R_j, $j = 1, 2, \ldots, l$, have been obtained, the sample corresponding to the minimal R_j is selected as the first "significant node", denoted by x_1^0. The minimal R_j is denoted by $R^{(1)}$, and the corresponding solution, A_j, is denoted by $A^{(1)}$. Then all the A_j and R_j are cancelled. In practice, for each step of determining "significant node", once a "significant node" is sorted out, all the A_j and R_j should be cancelled.

Step 2. Determine the sth "*significant node*"

After $s - 1$ "significant nodes", $x_1^0, x_2^0, \ldots, x_{s-1}^0$, have been determined, the corresponding matrix K is

$$K_{s-1} = \begin{bmatrix} 1 & k(x_1, x_1^0) & \ldots & k(x_1, x_{s-1}^0) \\ 1 & k(x_2, x_1^0) & \ldots & k(x_2, x_{s-1}^0) \\ \vdots & \vdots & & \vdots \\ 1 & k(x_l, x_1^0) & \ldots & k(x_l, x_{s-1}^0) \end{bmatrix}. \tag{18}$$

If the sample $x_j (j = 1, 2, \ldots, l, x_j \neq x_1^0, x_2^0, \ldots, x_{s-1}^0)$ is being investigated, the "kernel vector" $k_j = [k(x_1, x_j), k(x_2, x_j), \ldots, k(x_l, x_j)]'$ will attend the matrix K as the last column vector. Then A is determined according to Eq. (15) and is denoted by A_j. R_j is also estimated by $R_j = [\mu\|A_j\|_2^2 + \|KA_j - B\|_2^2]^{1/2}$. After each $x_j (j = 1, 2, \ldots, l, x_j \neq x_1^0, x_2^0, \ldots, x_{s-1}^0)$ has been investigated, the sample corresponding to the minimal R_j is selected as the sth "significant node", and the minimal R_j is denoted by $R^{(s)}$.

To obtain A_j, we should figure out the inverse matrix of $K'K + \mu I$ in advance. If the size of the matrix $K'K + \mu I$ is very large, the computation will be time consuming. Fortunately, because $K'K + \mu I$ is a symmetric matrix, an optimization algorithm of computing the inverse of $K'K + \mu I$ can be designed as follows. Above of all, we rewrite $K'K + \mu I$ as the following form:

$$\begin{aligned} K'K + \mu I &= \begin{bmatrix} K_{s-1}' \\ k_j' \end{bmatrix} [K_{s-1} \quad k_j] + \mu I \\ &= \begin{bmatrix} K_{s-1}'K_{s-1} + \mu I_s & K_{s-1}'k_j \\ k_j'K_{s-1} & k_j'k_j + \mu \end{bmatrix}, \end{aligned} \tag{19}$$

where I_s is the $s \times s$ identity matrix. Let $D = K'K + \mu I$, $D_{s-1} = K_{s-1}'K_{s-1} + \mu I_s$, $u = K_{s-1}'k_j$, $\gamma = k_j'k_j + \mu$, then D can be rewritten as

$$D = \begin{bmatrix} D_{s-1} & u \\ u' & \gamma \end{bmatrix}. \tag{20}$$

By virtue of the technique of matrix calculus, we obtain $\det(D) = (\gamma - u'(D_{s-1})^{-1}u)\det(D_{s-1})$,

$$adj(D) = \begin{bmatrix} (\gamma - u'(D_{s-1})^{-1}u) \cdot adj(D_{s-1}) & -adj(D_{s-1})u \\ +adj(D_{s-1})uu'(D_{s-1})^{-1} & \\ -\det(D_{s-1})u'(D_{s-1})^{-1} & \det(D_{s-1}) \end{bmatrix}.$$

Because of $D^{-1} = adj(D)/\det(D)$, the following equations can be worked out [16]:

$$(D)^{-1} = \frac{1}{\rho} \begin{bmatrix} \rho(D_{s-1})^{-1} + (D_{s-1})^{-1} & -(D_{s-1})^{-1}u \\ \times uu'(D_{s-1})^{-1} & \\ -u'(D_{s-1})^{-1} & 1 \end{bmatrix}, \tag{21}$$

where $\rho = \gamma - u'(D_{s-1})^{-1}u$. If $z = (D_{s-1})^{-1}u$, it will follow by Eqs. (15) and (21) that

$$\begin{aligned} A_j &= (D)^{-1}K'B \\ &= \frac{1}{\rho} \begin{bmatrix} \rho(D_{s-1})^{-1} + zz' & -z \\ -z' & 1 \end{bmatrix} \begin{bmatrix} K_{s-1}'B \\ k_j'B \end{bmatrix}. \end{aligned} \tag{22}$$

So, A_j can be obtained according to

$$A_j = \begin{bmatrix} A^{(s-1)} + \frac{1}{\rho}zz'K_{s-1}'B - \frac{1}{\rho}zk_j'B \\ -\frac{1}{\rho}z'K_{s-1}'B + \frac{1}{\rho}k_j'B \end{bmatrix},$$

$$A^{(s-1)} = (D_{s-1})^{-1}K_{s-1}'B, \tag{23}$$

where $A^{(s-1)}$ is the solution of Eq. (15) based on the previous $s - 1$ "significant nodes". Since $(D_{s-1})^{-1}$, $A^{(s-1)}$ and $K_{s-1}'B$ are also worked out while the $(s-1)$th "significant node" is determined, A_j will be not computationally intensive according to Eq. (23). We call the above algorithm an optimization algorithm ofA_j. The key of the algorithm is to avoid the direct computation of $(K'K + \mu I)^{-1}$.

We repeat step 2 till $|R^{(s)} - R^{(s-1)}| < \varepsilon$ (ε denotes a threshold) is satisfied. Finally, the A_j corresponding to the last "significant node" is taken as the near-optimal solution of A and is denoted by $A^{(s)}$. Strictly speaking, it is not assured that the solution is optimal among all the potential solutions. However, all the potential solutions only can be obtained based on the enumerative method, which is not feasible for classification problems with a large number of training samples. Here we only aim at finding the near optimal solution according to the obtained solutions, and regard the solution corresponding to the last "significant node" as the near-optimal solution. The above method is called fast kernel-based nonlinear method (FKNM). The main character of FKNM is selecting "significant nodes" from the total training samples and classifying test samples based on the kernel functions between them and the test samples.

4. Multi-class FKNM

The NKNM and the FKNM originally focus on two-class classification. There are two well-known approaches to extend them into multi-class classifications. The first is one-against-the-rest, and the second is one-against-one [17,18]. Suppose that there are L classes. For one-against-the-rest, one classifier is trained for one class and the other $L-1$ classes. Consequently, L classifiers are necessary. As for one-against-one, one classifier is only trained between a pair of classes, so the number of necessary classifiers is $L(L-1)/2$. With one-against-the-rest, all the L trained classifiers should run once for classifying one test sample. As for one-against-one, it is also necessary for the $L(L-1)/2$ trained classifiers to run once to classify one test sample. If different denotation codes are assigned to different categories, training and classifying can be performed according to the codes. After all the trained classifiers have run once for one test sample, the test sample is classified into the class, whose code is the closest to the computed outputs of the test sample. For details, please see Refs. [18,19].

If the NKNM is applied to multi-class problems, the computational complexity of classification is still directly proportional to the number of the training samples. Besides, it has been shown that when applied to multi-classes, a binary classification method with one-against-one is superior to that with one-against-the-rest in efficiency [19]. Naturally, as a binary classification algorithm, NKNM is also in accordance with the above statement.

However, for the FKNM, the situation may be different. Firstly, to classify one test sample, we only need to use the kernel functions between the test sample and the "significant nodes". As a result, the classification efficiency of the FKNM is directly related to the number of the "significant nodes" rather than that of the total training samples. Secondly, the FKNM with one-against-the-rest may be more efficient than the FKNM with one-against-one, if the "significant nodes" in the former are less than those in the latter. We will further discuss the issue by experiments.

5. Experiments

5.1. Experiment on benchmark datasets

An experiment is performed on several two-class benchmark datasets (http://ida.first.gmd.de/~raetsch/data/). 100 partitions are generated for each dataset and every partition includes one training sample subset and one test sample subset. A Gaussian kernel in the form of $k(x,y) = \exp(-\|x-y\|^2/(2\eta))$ is adopted. For each dataset, training is implemented on the first training subset, whereas testing is performed on all the test subsets. Because every test subset corresponds to an error rate, we can figure out the average error rate of all the test subsets in one dataset and take it as that of the whole dataset. Also, the average deviation of the error rates on all the test subsets is regarded as that of the whole dataset. Let η be equal to the square of Euclid norm of the covariance matrix of the first training subset. 0.0001, 0.001, 0.01 are assigned to μ, respectively. When the FKNM is applied, ε is set to be 0.01 for the dataset named Banana, and is set to be 0.02 for the other datasets. The classification result of the NKNM is given in Table 1, and the performance of the FKNM is shown in Table 2. It is clear that as a whole the classification performance of the FKNM is comparable with that of NKNM. Moreover, according to Table 3, it appears that the "significant nodes" are much less than the total training samples. Calculating the ratio of the number of "significant nodes" to that of the corresponding total training samples, we can see that the smallest ratio is only 3.2%, and the largest one is 18.6%. In other words, while the FKNM is used to classify one test sample, only kernel functions between a few training samples and the test sample are calculated and used. So, by contrast with the NKNM, which classifies one test sample based on all the kernel functions between the total training samples and the test sample, the FKNM must be much more efficient in classifying. With different μ, the NKNM achieves gently different error classification rates. Meanwhile, the performance of the FKNM is also slightly variable while μ varies. The greatest fluctuation of classification accuracy occurs on dataset named "thyroid".

For the same datasets, we conduct the experiment again with sigmoid kernel $k(x_i, x_j) = \tanh(sx_i \cdot x_j + t)$, where s, t are two coefficients. The setting of ε is the same as the above paragraph. The experimental results, shown in Tables 1, 2 and 3, also indicate that the classification performance of the FKNM is comparable with the NKNM, and the "significant nodes" are much less than the corresponding total training samples. The largest ratio of the "significant nodes" to the corresponding total training samples is 17.1%, and the smallest ratio is only 1.4%.

5.2. Experiment on CMU mask images

The CMU-Pittsburgh AU-Coded Face Expression Database [20] is a dataset of face images with happy, joy, angry, sad, surprised and disgusted expressions. By using the spatial adaptive triangulation technique based on local Gabor filters [21], 463 facial expression mask images are obtained. 210 images are taken as training samples, while the others are regarded as test samples. There solution of

Table 1
Experimental results of the NKNM on four two-class benchmarks

	μ	0.0001	0.001	0.01		
Result of Gaussian kernel						
The average of error	F. solar	31.7 ± 1.9	31.7 ± 1.9	31.6 ± 1.9		
classification rates	Titanic	22.7 ± 0.3	22.7 ± 0.3	22.7 ± 0.3		
and the standard	Thyroid	2.8 ± 1.3	2.8 ± 1.5	4.6 ± 2.0		
deviation	Banana	11.0 ± 0.1	11.5 ± 0.1	12.2 ± 0.1		
Result of sigmoid kernel					s	t
The average of error	F. solar	32.2 ± 1.8	32.8 ± 1.8	33.0 ± 1.8	0.2	0.6
classificationrates	Titanic	21.7 ± 0.3	21.7 ± 0.3	22.7 ± 0.3	0.2	0.6
and the standard	Thyroid	3.4 ± 1.7	1.9 ± 1.4	1.8 ± 1.3	0.2	0.6
deviation	Banana	11.2 ± 0.1	10.9 ± 0.1	11.0 ± 0.1	0.6	0.6

Table 2
Experimental results of the FKNM on four two-class benchmarks

	μ	0.0001	0.001	0.01		
Result of Gaussian kernel						
The average of error	F. solar	32.0 ± 1.8	32.0 ± 1.4	31.7 ± 1.8		
classification rates	Titanic	22.6 ± 0.3	22.6 ± 0.3	22.6 ± 0.3		
and the standard	Thyroid	2.8 ± 1.5	2.8 ± 1.5	6.9 ± 2.2		
deviation	Banana	11.4 ± 0.1	12.1 ± 0.1	12.8 ± 0.1		
Result of sigmoid kernel					s	t
The average of error	F. solar	32.8 ± 1.8	33.1 ± 1.8	33.7 ± 1.9	0.2	0.6
classificationrates	Titanic	22.6 ± 0.3	22.6 ± 0.3	22.6 ± 0.3	0.2	0.6
and the standard	Thyroid	2.0 ± 1.3	2.3 ± 1.4	1.4 ± 0.9	0.2	0.6
deviation	Banana	11.3 ± 0.1	11.2 ± 0.1	11.1 ± 0.1	0.6	0.6

Table 3
Number of the total training samples of each benchmark and the number of the “significant nodes” in the FKNM

	μ	F. Solar	Banana	Titanic	Thyroid
The Number of total training samples		666	400	150	140
The number of	0.0001	21 (3.2%)	33 (8.3%)	5 (3.3%)	23 (16.4%)
“significant nodes”	0.001	21 (3.2%)	32 (8.0%)	5 (3.3%)	26 (18.6%)
with Gaussian kernel	0.01	21 (3.2%)	32 (8.0%)	5 (3.3%)	12 (8.6%)
and varying μ					
The number of	0.0001	9 (1.4%)	41 (10.3%)	7 (4.7%)	24 (17.1%)
“significant nodes”	0.001	17 (2.5%)	41 (10.3%)	7 (4.7%)	20 (14.3%)
with sigmoid kernel	0.01	18 (2.7%)	50 (12.5%)	6 (4.0%)	23 (16.4%)
and varying μ					

Number in the bracket is the ratio of the number of “significant nodes” to that of the corresponding total training samples.

each image is 60 × 70 pixels. We will recognize the faces according to their expressions, i.e. each test sample will be sorted into one of the expression categories. In the experiment, kernel function $k(x_i, x_j) = (x_i \cdot x_j)^2$ is adopted and each image is transformed into a vector by stacking its columns. For every image, each pixel value is divided by 6000, so the order of magnitudes of $k(x_i, x_j)$ is about $o(1)$ and consequent matrix K corresponds to a low condition number. In addition, μ is set to be 0.001, and for the FKNM 0.1 is assigned to ε.

Table 4 presents the classification results of the FKNM with one-against-the-rest. The error classification rate of the method is 11%, while that of the NKNM is 11.5% (not shown in the table). It implies that the FKNM is comparable to NKNM in classification correctness. The total number of the “significant nodes” is only 70. So, while the FKNM is applied for classification decision of one test sample with one-against-the-rest approach, only 70 kernel functions should be calculated in advance. However, since there are 210 training samples in total, 210 kernel functions should

Table 4
Experimental results of the FKNM with one-against-the-rest on the CMU mask images

Classifier	$B1$	$B2$	$B3$	$B4$	$B5$	$B6$	Total	Error rate (%)	Time (s)
Number of significant nodes	8	13	15	9	10	15	70	11	14.0

$B1, B2, \ldots,$ and $B6$ denote 6 different classifiers in the FKNM with one-against-the-rest. Based on the 6 classifiers, an error classification rate of 11% is obtained. The total number of "significant nodes" used by the classifiers is 70. The time consumed for classification is 14.0 s.

Table 5
Experimental results of the FKNM with one-against-one on the CMU mask images

Classifier	$B1$	$B2$	$B3$	$B4$	$B5$	$B6$	$B7$	$B8$	$B9$	$B10$	$B11$	$B12$	$B13$	$B14$	$B15$	Total	Error rate (%)	Time (s)
Number of significant nodes	11	8	6	8	6	8	8	10	13	10	6	15	12	10	12	143	14.6	28.0

$B1, B2, \ldots,$ and $B15$ denote 15 different classifiers in the FKNM with one-against-one. Based on these classifiers, an error classification rate of 14.6% is obtained. The total number of "significant nodes" used by the classifiers is 143. The time consumed for classification is 28.0 s.

be worked out and used while the NKNM classifies one test sample. The ratio of the number of the "significant nodes" to that of the whole training samples is only 33%. The time used for classification is 14.0 s.

Table 5 shows the classification results of the FKNM with one-against-one. The classification error rate is 14.6%, bigger than that of the FKNM with one-against-the-rest. The total number of the "significant nodes" is 143, which is 68% of that of the total training samples. For classification 28.0 s is consumed, which is much longer than that of the FKNM with one-against-the-rest.

It is usually considered that one-against-one is superior to one-against-the-rest in classification efficiency. However, for the FKNM the situation is different, as is shown in the above context. With less kernel functions and classifiers, the FKNM with one-against-the-rest can classify test samples more efficient than the FKNM with one-against-one. It seems that the FKNM with one-against-the-rest benefits more from "significant nodes".

6. Conclusion

For two-class classifications, it is demonstrated that from the viewpoint of feature space the nonlinear discriminant analysis with class labels 1 and -1 is consistent with the Fisher discriminant analysis. By introducing kernel function, the nonlinear discriminant analysis may be transformed into a form of NKNM (naive kernel-based nonlinear method).

For the NKNM, the classification efficiency is so enslaved to the number of the training samples that it becomes very time-consuming and even impracticable if the number of the training samples is large. The FKNM (fast kernel-based nonlinear method), is developed to speed up the NKNM. The key of the method is to select the "significant nodes". They may replace all the training samples to expand the discriminant vector in the feature space. An efficient algorithm is developed to determine "significant nodes", and then the classification process for two-class problems can be performed much more efficiently based on the "significant nodes".

Furthermore, two well-known multi-class approaches, one-against-the-rest and one-against-one, are introduced to extend the FKNM into multi-class problems. Analysis indicates that, differing from ordinary binary classification method in which one-against-one seems to be preferable, the FKNM with one-against-the-rest can be superior to the FKNM with one-against-one in classification efficiency. The experiments on benchmarks and real images database substantially support our expectations and illustrate that the fast method proposed in this paper is effective, efficient and feasible for two-class and multi-class classifications.

Acknowledgements

We wish to thank National Natural Science Foundation of China under grants no. 60472060, no. 60473039 and no. 60472061.

References

[1] Y. Xu, J.Y. Yang, J.F. Lu, D.J. Yu, An efficient renovation on kernel Fisher discriminant analysis and face recognition experiments, Pattern Recognition 37 (2004) 2091–2094.

[2] Y. Xu, J.Y. Yang, J. Yang, A reformative Fisher discriminant analysis, Pattern Recognition 37 (2004)1299–1302.

[3] B. Schölkopf, A. Smola, K.R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.

[4] K.-R. Muller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, An introduction to kernel-based learning algorithms, IEEE Trans. Neural Network 12 (1) (2001) 181–201.

[5] S.A. Billings, K.L. Lee, Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm, Neural Networks 15 (1) (2002) 263–270.

[6] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, K.-R. Müller, Fisher discriminant analysis with kernels, in: Neural Networks for Signal Processing IX, IEEE, New York, 1999, pp. 41–48.

[7] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, New York, 1995.

[8] J.H. Xu, X.G. Zhang, Y. Li, Kernel MSE algorithm: a unified framework for KFD, LS-SVM and KRR, in: Proceedings of the International Joint Conference on Neural Networks(IJCNN-2001), Washington, DC, 2001 pp.1486-1491.

[9] G.C. Cawley, N.L.C. Talbot, Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers, Pattern Recognition 36 (11) (2003) 2585–2592.

[10] Z. Bian, X. Zhang, Pattern recognition (in Chinese), Tsinghua University Press, Beijing, 2000.

[11] S. Mika, G. Rätsch, K.-R. Müller, A mathematical programming approach to the kernel Fisher algorithm, Adv. Neural Inf. Process. Syst. 13 (2001) 591–597.

[12] Y. Xu, J.Y. Yang, Z. Jin, Theory analysis on FSLDA and ULDA, Pattern Recognition 36 (12) (2003) 3031–3033.

[13] Y. Xu, J.Y. Yang, Z. Jin, A novel method for Fisher discriminant analysis, Pattern Recognition 37 (2004) 381–384.

[14] B. Schölkopf, A. Smola, R. Williamson, P. Bartlett, New support vector algorithms, Technical Report, GMD, 1998, NC2-TR-1998-031.

[15] Y.-J. Lee, O.L. Mangasarian, RSVM: reduced support vector machines, Technical Report 00-07, University of Wisconsin, Madison, Wisconsin, 2000.

[16] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., John Hopkins University Press, Baltimore, London, 1996.

[17] D.H. Wolpert, Stacked generalization, Neural Networks 5 (1) (1992) 241–260.

[18] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Ann. Stat. 26 (2) (1998) 451–471.

[19] J. Furnkranz, Round robin classification, J. Mach. Learn. Res. 2 (2002) 721–747.

[20] T. Kanade, J.F. Cohn, Y.L. Tian, Comprehensive database for facial expression analysis, in: Proceeding of the Fourth International Conference of Face and Gesture Recognition, Grenoble, France, 2000, pp. 46–53.

[21] S. Dubuisson, F. Davoine, M. Masson, A solution for facial expression representation and recognition, Signal Process.: Image Commun. 17 (9) (2002) 657–673.

[22] J. Xu, J.Y. Yang, J.F. Lu, An efficient kernel-based nonlinear regression method for two-class classification, in: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, August, 2005, pp. 4442–4445.

About the Author—YONG XU received the B.S. degree and the M.S. degree in 1994 and 1997, respectively. His current interests include face recognition, handwritten character recognition, linear and nonlinear discriminant analysis.

About the Author—DAVID ZHANG graduated in computer science from Peking University in 1974 and received the M.Sc. and Ph.D. degrees in computer science and engineering from the Harbin Institute of Technology (HIT) in 1983 and 1985, respectively. He received a second Ph.D. degree in electrical and computer engineering from the University of Waterloo, Ontario, Canada, in 1994. After that, he was an associate professor at the City University of Hong Kong and a professor at the Hong Kong Polytechnic University. Currently, he is a founder and director of the Biometrics Technology Centre supported by the UGC of the Government of the Hong Kong SAR. He is the founder and editor-in-chief of the International Journal of Image and Graphics and an associate editor for some international journals such as the IEEE Transactions on Systems, Man, and Cybernetics, Pattern Recognition, and International Journal of Pattern Recognition and Artificial Intelligence. His research interests include automated biometrics-based identification, neural systems and applications, and image processing and pattern recognition. So far, he has published more than 180 papers as well as 10 books, and won numerous prizes. He is a senior member of the IEEE and the IEEE Computer Society.

About the Author—ZHONG JIN received the B.S. degree in mathematics, the M.S. degree in applied mathematics, and the Ph.D. degree in pattern recognition and intelligence system from Nanjing University of Science and Technology (NUST), China, in 1982, 1984, and 1999, respectively. He is a professor in the Department of Computer Science, NUST. He visited the Department of Computer Science and Engineering, the Chinese University of Hong Kong from January 2000 to June 2000 and from November 2000 to August 2001 and visited the Laboratoire HEUDIASYC, UMR CNRS 6599, Universite de Technologie de Compiegne, France, from October 2001 to July 2002. Dr. Jin is now visiting the Centre de Visio per Computador, Universitat Autonoma de Barcelona, Spain, as the Ramon y Cajal program Research Fellow. His current interests are in the areas of pattern recognition, computer vision, face recognition, facial expression analysis, and content-based image retrieval.

About the Author—MIAO LI obtained her B.S. degree in computer science from Jilin University, Changchun, China, in 2003. Now she is a MS.D. student in Bio-Computing Research Center and Shenzhen graduate school, Harbin Institute of Technology, Shenzhen, China. Her current interests include pattern recognition, image processing, and neural network.

About the Author—JING-YU YANG received the B.S. degree in computer science from Nanjing University of Science and Technology (NUST), Nanjing, China. From 1982 to 1984, he was a visiting scientistat the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 1993 to 1994, he was a visiting professor in the Department of Computer Science, Missuria University, and in 1998, he acted as a visiting professor at Concordia University in Canada. He is currently a professor and chairman in the Department of Computer Science at NUST. He is the author of more than 300 scientific papers in computer vision, pattern recognition, and artificial intelligence. He has won more than 20 provincial and national awards. His current research interests are in the areas of pattern recognition, robot vision, image processing, data fusion, and artificial intelligence.