Optimal Iterative Discriminant Analysis In Kernel Space

Wei Liu [a] Hexin Chen [a,b] Mianshu Chen [b]

[a] *P.O.Box 66 Guilin Road, Changchun, 130021, Jilin , China*
davidieee@yahoo.com.cn

[b] *Institute of Communication Engineering, Nanling Campus, Jilin University, Changchun , 130025, Jilin, China*

Abstract

Kernel trick is a powerful tool being used for solving complex pattern classification problem. As long as a linear feature extraction algorithm can be expressed exclusively by dot-products, it can be extended to non-linear version by combining kernel method. In this paper, we present such an improved iterative algorithm used for linear discriminant analysis. By mapping data onto high dimensional feature space suing kernel function, we make data linearly separable and run iterative LDA there. Experiments with minimum distance classifier and nearest neighbor classifier show that our improved algorithm has a better performance than traditional Fisher discriminant and standard iterative LDA.

Key words:
feature extraction, linear discriminant analysis, kernel space, face recognition

1 Introduction

Subspace methods, such as principal component analysis (PCA) and linear discriminant analysis (LDA), have been widely employed for dimension reduction and feature extraction (Moghaddam , 2002; Yang , 2002; Belhumeur , 1997). PCA aims to find a subspace which maximize covariance and minimize reconstruction errors (Moghaddam and Pentland, 1997). Since it treats all samples as belonging to one class, PCA does not exploit the class information of samples. As a result, some unwanted variance(for example, some changes in lighting, facial expressions and viewing points) may be retained. LDA finds a subspace that has a minimal within-class scatter and maximal between-class scatter. One of the differences between PCA and LDA is the former gets orthogonal discriminant vectors while the latter does not. To get

orthogonal discriminant vectors, Guo et al. (2003) proposed an iterative algorithm that finds optimal linear discriminant vectors in global sense. All the vectors obtained by this algorithm are subject to orthogonal constraint: $\psi_i^\top \psi_j = 0, \forall\ i \neq j \quad i,j = 1,...,r$. However, due to its linear transformation characteristic, LDA fails to deal with nonlinear problems, which may be more often confronted in real world. In this paper, we generalize Guo's iterative algorithm to nonlinear situations and present kernel-based optimal iterative discrminant analysis (KOIDA) This is achieved by firstly mapping the samples non-linearly to high dimensional feature space $\boldsymbol{F}$**,** and computing Guo's linear discriminant there, thus implicitly yielding a non-linear projection in input space.

The basic idea behind the kernel trick is that a non-linear decision surface can be exactly the same as a linear decision surface in a high dimensional space. For example, we can get a quadratic discriminant in coordinates(x_1, x_2) by constructing a linear discriminant in a 5-dimentional space with coordinates $(x_1, x_2, x_1^2, x_1x_2, x_2^2)$. Theoretically speaking, any non-linear decision surface can be transformed into a linear one in another feature space. In practice, however, the extremely large, possibly infinite dimensionality of the feature space often makes this explicit mapping impossible. The kernel trick was initially proposed in Support Vector Machines to address the above "dimension curve" problem (see Burges , 1998; Osuna , 1997). Instead of mapping the data explicitly, it seek a formulation of the original algorithm using only dot-products $(\phi(\mathbf{x})\cdot\phi(\mathbf{y}))$ of the data. The kernel trick then computes dot-products of samples in feature space by using kernel function in input space. As long as any feature extraction algorithm can be expressed in terms of dot-products, the kernel method enable us to construct its non-linear variant without ever mapping explicitly to feature space. Schölkopf (1998) extended the classical PCA to Kernel Principal Component Analysis (KPCA). Mika et al. (1999), Baudat et al. (2000), Roth and Steinhage (2000) show that the same procedure can be applied to Fisher discriminant analysis, This article shows that it is possible to formulate Guo's optimal iterative linear discriminant analysis exclusively in terms of dot-products. Thus, a nonlinear version of this iterative algorithm can be constructed using kernel method. We call our version of the algorithm Kernel Optimal Iterative Discriminant Analysis (KOIDA).

The following section will firstly give a short review of optimal iterative discriminant algorithm, then formulate its main steps in a way that uses only dot-products. This section forms the basis for section 3, which presents the proposed kernel-based KOIDA algorithm. Section 4 gives experimental results. Conclusions are drawn in Section 5.

2 OIDA AND ITS VARIANT IN FEATURE SPACE

In order to get more than one linear discriminant vectors, Foley and Sammon compute each vectors step by step based on generalized Fisher criterion in the orthogonal complementary space of the subspace spanned by the discriminant vectors calculated before(Guo et al., 2003). Yet the discriminant sets obtained in this way only maximize generalized Fisher criterion in local sense. Guo's iterative algorithm is able to get discriminant sets in the sense of global optimality.

2.1 Optimal Iterative Discriminant Analysis

In a C-class problem, let $\omega_1, \ldots, \omega_C$ be C known patterns classes, and $x_1, \ldots, x_N$ be the set of n-dimensional samples. The sample number of the lth class is N_l , thus $\sum_{l=1}^{C} N_l = N$. Let $\mathbf{S}_b$, $\mathbf{S}_w$, $\mathbf{S}_t$ be the between-class scatter, the within-class scatter and the population scatter. Then Fisher's linear discriminant is given by the vector ψ which maximizes

$$J_F(\psi) = \frac{\psi^\top \mathbf{S}_b \psi}{\psi^\top \mathbf{S}_w \psi} \quad . \tag{1}$$

When more than one discriminant vector are desired, we can use generalized Fisher criterion

$$J(\mathbf{\Psi}) = \frac{tr(\mathbf{\Psi}^\top \mathbf{S}_b \mathbf{\Psi})}{tr(\mathbf{\Psi}^\top \mathbf{S}_w \mathbf{\Psi})} = \frac{\sum_{i=1}^{r} \psi_i^\top \mathbf{S}_b \psi_i}{\sum_{i=1}^{r} \psi_i^\top \mathbf{S}_w \psi_i} \tag{2}$$

where $\mathbf{\Psi} = (\psi_1, \psi_2, \ldots, \psi_r)$, and r is the number of discriminant vectors.

Instead of trying to maximize $J(\mathbf{\Psi})$ directly, Guo shows it is equivalent to solving the eigenvalue problem of $\mathbf{S}_b - \lambda_0 \mathbf{S}_w$ in an iterative way.

Theorem 1 *Suppose* $\boldsymbol{A}$ *is a real symmetric matrix of n order,* $\boldsymbol{B}$ *is a positive-definite matrix of n order, then*

$$\lambda_0 = \frac{\sum_{l=1}^{r} \tilde{\psi}_l^\top \mathbf{A} \tilde{\psi}_l}{\sum_{l=1}^{r} \tilde{\psi}_l^\top \mathbf{B} \tilde{\psi}_l} = \max_{\substack{\psi_i^\top \psi_j = 0 \\ \|\psi_i\| = 1}} \left(\frac{\sum_{l=1}^{r} \psi_l^\top \mathbf{A} \psi_l}{\sum_{l=1}^{r} \psi_l^\top \mathbf{B} \psi_l} \right)$$

$i, j = 1, ..., r, i \neq j$

if and only if

$$\sum_{l=1}^{r} \tilde{\psi}_l^\top (\mathbf{A} - \lambda_0 \mathbf{B}) \tilde{\psi}_l$$

$$= \max_{\substack{\psi_i^\top \psi_j = 0 \\ \|\psi_i\| = 1}} \left(\sum_{l=1}^{r} \psi_l^\top (\mathbf{A} - \lambda_0 \mathbf{B}) \psi_l \right) = 0$$

$i, j = 1, ..., r, i \neq j$

where $\tilde{\psi}_i^\top \tilde{\psi}_j = 0, i \neq j, i, j = 1, ..., r$

Theorem 2 *Under the assumption of Theorem 1, it holds that*

(1) $\lambda < \lambda_0$ *if and only if*

$$\max_{\substack{\psi_i^\top \psi_j = 0 \\ \|\psi_i\| = 1}} \left(\sum_{l=1}^{r} \psi_l^\top (\mathbf{A} - \lambda \mathbf{B}) \psi_l \right) > 0$$

(2) $\lambda > \lambda_0$ *if and only if*

$$\max_{\substack{\psi_i^\top \psi_j = 0 \\ \|\psi_i\| = 1}} \left(\sum_{l=1}^{r} \psi_l^\top (\mathbf{A} - \lambda \mathbf{B}) \psi_l \right) < 0$$

Thus, we can make the first r eigenvalues of $(\mathbf{S}_b - \lambda \mathbf{S}_t)$ be zero (here $\mathbf{S}_w$ is replaced by $\mathbf{S}_t$) by adjusting the value of λ at each iteration. Since it has been proved that the algorithm converges to theoretical solution, the errors of the iterative procedure would go below a given threshold after finite steps. Then, the first r eigenvectors of $(\mathbf{S}_b - \lambda \mathbf{S}_t)$ can maximize $J(\mathbf{\Psi})$. One can refer to Guo et al. (2003) for the proof procedure of the above theorems.

2.2 Optimal Iterative Discriminant Analysis In Feature Space

For most real-world data such as face images, a linear discriminant is not complex enough. Neither Fisher discriminant nor OIDA is able to deal with data that cannot be linearly separated. Here we extend OIDA to non-linear problems by implicitly mapping the data non-linearly into feature space $\boldsymbol{F}$.

Let ϕ be a non-linear mapping to some $\boldsymbol{F}$ and $\mathbf{S}_b^F$, $\mathbf{S}_w^F$, $\mathbf{S}_t^F$ be the between-class scatter, within-class scatter and population scatter in feature space $\boldsymbol{F}$. To generalize OIDA to non-linear case we need to formulate it in a way that uses exclusively dot-product. Therefore, consider an expression of dot-products in feature space $\boldsymbol{F}$ given by the following kernel function

$$k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \phi^\top(\mathbf{x}_i)\phi(\mathbf{x}_j) \tag{3}$$

For known classes p and q, this kernel function can be expressed as

$$(k_{ij})_{pq} = \phi^\top(\mathbf{x}_{pi})\phi(\mathbf{x}_{qj})$$

where $\mathbf{x}_{pi}$ is the element i of the class p. Then defining a $N \times N$ matrix $\mathbf{K}$ by

$$\mathbf{K} = (\mathbf{k}_{pq}) \quad \begin{matrix} p = 1, ..., C \\ q = 1, ..., C \end{matrix}$$

where $\mathbf{k}_{pq} = (k_{ij}) \quad \begin{matrix} i = 1, ...N_p \\ j = 1, ..., N_q \end{matrix}$

For computation simplicity, we also define the $N \times N$ block diagonal matrix

$$\mathbf{W} = (\mathbf{W}_l) \quad l = 1, ..., C$$

where $\mathbf{W}_l$ is $N_l \times N_l$ with elements all equal to $1/N$.

Now in feature space $\boldsymbol{F}$, the main step of OIDA is to solve the eigenvalue problem

$$(\mathbf{S}_b^F - \lambda_0 \mathbf{S}_t^F)\mathbf{v} = \lambda \mathbf{v} \tag{4}$$

According to the theory of reproducing kernels we know that any solution

$\mathbf{v} \in F$ lies in the span of all training samples in $\boldsymbol{F}$. So we can find an expansion for v in the form

$$\mathbf{v} = \sum_{p=1}^{C} \sum_{q=1}^{N_p} \alpha_{pq} \phi(\mathbf{x}_{pq}) \tag{5}$$

where α_{pq} is the coefficients of $\phi(\mathbf{x}_{pq})$.

Then in appendix, eq.(4) can be formulated to the following quotient:

$$\lambda = \frac{\alpha^\top (\frac{1}{N}\mathbf{KWK} - \frac{\lambda_0}{N}\mathbf{KK})\alpha}{\alpha^\top \mathbf{K}\alpha} \tag{6}$$

α is an coefficient vector with elements α_{pq}, $p = 1, \ldots, C$, $q = 1, \ldots, N_p$.

Note eq.(6) uses only dot-products of samples in $\boldsymbol{F}$ space, which can be computed by the samples in input space without having to carry out the map ϕ. To solve the eigenvalue problem of eq.(6), we decompose $\mathbf{K}$ in the following form

$$\mathbf{K} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top$$

where $\mathbf{\Lambda}$ is the diagonal matrix of non-zero eigenvalues of $\mathbf{K}$ and $\mathbf{P}$ is the matrix of normalized eigenvectors associated to $\mathbf{\Lambda}$, then we get

$$\lambda = \frac{\frac{1}{N}(\mathbf{\Lambda}\mathbf{P}^\top\alpha)^\top\mathbf{P}^\top\mathbf{W}\mathbf{P}(\mathbf{\Lambda}\mathbf{P}^\top\alpha) - \frac{\lambda_0}{N}(\mathbf{\Lambda}\mathbf{P}^\top\alpha)^\top(\mathbf{\Lambda}\mathbf{P}^\top\alpha)}{\alpha^\top\mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top\alpha}$$

Let $\beta = \mathbf{\Lambda}\mathbf{P}^\top\alpha$, we get

$$\lambda = \frac{\frac{1}{N}\beta^\top\mathbf{P}^\top\mathbf{W}\mathbf{P}\beta - \frac{\lambda_0}{N}\beta^\top\beta}{\beta^\top\mathbf{\Lambda}^{-1}\beta}$$

This equation is equivalent to

$$\lambda\beta = (\frac{1}{N}\mathbf{\Lambda}\mathbf{P}^\top\mathbf{W}\mathbf{P} - \frac{\lambda_0}{N}\mathbf{\Lambda})\beta \tag{7}$$

Thus the solution of eq.(6) can be found by solving the eigenvalue problem of eq.(7). For a given β there exists at least one α satisfying eq.(6) in the form

$$\alpha = \mathbf{P}\mathbf{\Lambda}^{-1}\beta \tag{8}$$

Because discriminant vectors $\mathbf{v}$ must be normalized in $\boldsymbol{F}$, their corresponding α are divided by $\sqrt{\alpha^\top \mathbf{K}\alpha}$, as derived by Baudat et al. (2000).

3 KERNEL-BASED OIDA ALGORITHM

After the main step of OIDA in feature space are expressed by dot-products, we present kernel-based KOIDA algorithm as follows:

(1) Since $\max_{\boldsymbol{\Psi}} J(\boldsymbol{\Psi}) \in [0,1]$, let L=0, R=1 and $\lambda = (L+R)/2 = 0.5$. By solving eq.(7) we get the first r eigenvalues of $(\mathbf{S}_b^F - \lambda \mathbf{S}_t^F)$ and vectors $\beta_1, ...\beta_r$. Let $\varepsilon = \lambda_1 + ... + \lambda_r$.

If $\varepsilon > 0$, then $\lambda < \lambda_0$ holds according to theorem 2. Thus let L=λ, R=R; if $\varepsilon < 0$, then $\lambda > \lambda_0$, holds according to theorem 2, let L=L, R=λ. After each iteration finishes, it holds that $|\lambda - \lambda_0| < |a-b|/2$.

(2) Repeat step (1) until $|a-b| < \delta$, where δ is a given small positive tolerance. At last, $|\lambda - \lambda_0| \leq \delta$, then the iteration procedure finishes. The coefficient vectors $\alpha_1, ...\alpha_r$ can be derived from $\beta_1, ...\beta_r$ by eq.(8).

For any test samples $\mathbf{z}$, the projections of $\phi(\mathbf{z})$ on r optimal discriminant in feature space $\boldsymbol{F}$ are derived by

$$\begin{bmatrix} \mathbf{v}_1^\top \\ ... \\ \mathbf{v}_r^\top \end{bmatrix} \phi(\mathbf{z}) = \begin{bmatrix} \sum_{p=1}^{N} \sum_{q=1}^{N_p} \alpha_{pq}^1 k(\mathbf{x}_{pq}, \mathbf{z}) \\ ... \\ \sum_{p=1}^{N} \sum_{q=1}^{N_p} \alpha_{pq}^r k(\mathbf{x}_{pq}, \mathbf{z}) \end{bmatrix} \tag{9}$$

where $\mathbf{v}_i$ denotes the ith projection vector, and if we let α^i be the ith coefficient vector, then α_{pq}^i denotes its element associated with the qth sample in class p.

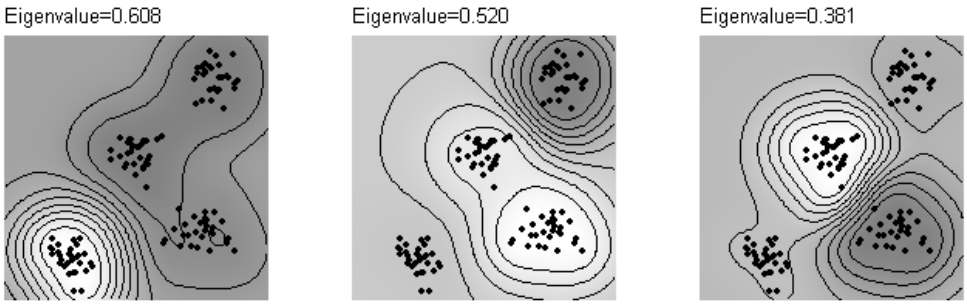

Fig. 1. 2D toy examples. Gray level denotes feature values and contour lines depict constant features. Note the three major eigenvectors are enough to classify four clusters.

4 EXPERIMENTS

We firstly use a simple toy example to illustrate how KOIDA works, In figure 1, it is shown that an artificial 2D-problem of 4 classes was solved using KOIDA with a Gaussian kernel and $\sigma = 0.7$. From left to right are the features obtained by the first three discriminant vectors in order of decreasing eigenvalue size. The features are indicated by gray level and identical feature values are expressed by contour lines. Because discriminant vectors lie in a high dimensional space, they cannot be drawn on the figure. We see although kernel trick can yield more than C non-zero eigenvalues, in this example, however, $(C-1)$ eigenvectors are sufficient for solving the problem.

To draw a comparison between our KOIDA and other algorithms, we adopted some of the datasets that were used in Mika et al. (1999). These datasets come from UCI, DELVE and STATLOG benchmark repositories(except for banana), and include both artificial and real world data, Mika et al. (1999) evaluate his two-class Kernel Fisher Discriminant on these datasets, while without loss of generality, multi-class feature extraction algorithms combined with a minimum distance classifier can be evaluated on them. In the experiment we chose 200 training samples and 200 test samples. three feature algorithms were selected and tested. They were traditional Fisher discriminant analysis (FDA), Guo's optimal iterative discriminant analysis (OIDA), and our kernel-based optimal iterative discriminant analysis (KOIDA). Their recognition rates are listed in table 1, where the data in the last column (KFD) come from Mika et al. (1999) experimental results and demonstrate the performance of Kernel Fisher Discriminant.

Table 1
Comparison between KOIDA and other algorithms on artificial and real world datasets

	Fisher	OIDA	KOIDA	KFD
banana	47.00%	14.00%	10.75%	10.8%
thyroid	12.00%	7.33%	4.27%	4.2%
titanic	22.55%	22.15%	21.80%	23.2%

In the last experiment we test these methods on a widely used pattern recognition benchmark database. As face recognition is an active area in pattern recognition and also a tough problem, the ORL face dataset was adopted in our experiments. This dataset include forty distinct subjects and each subject has ten images with resolution of 112×92. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. The variation in scale is up to 10%. The dataset are available at http://www.uk.research.att.com/facedatabase.html. From all the ten sam-

Table 2
Comparison of several feature extraction algorithms on face patterns

	Fisher	OIDA	KOIDA				
			Sigmoid	Polynomial d=2	Polynomial d=3	Polynomial d=4	Gaussian
MD&7x6	10/95.0%	20/90.0%	19/90.5%	8/96.0%	7/96.5%	7/96.5%	6/97.0%
NN&7x6	7/96.5%	14/93.0%	11/94.5%	8/96.0%	6/97.0%	7/96.5%	6/97.0%
MD&14x12	56/72.0%	21/89.5%	14/93.0%	11/94.5%	11/94.5%	11/94.5%	10/95.0%
NN&14x12	54/73.0%	19/90.5%	12/94.0%	11/94.5%	11/94.5%	11/94.5%	9/95.5%

ples in each class, we extracted five of them as training samples and the left as test samples, so there are 200 training patterns and test patterns respectively.

All the sample images were firstly sub-sampled to the resolution of 7×6 and 14×12. Then the features were obtained by lexicographic ordering of the pixel elements, yielding input vectors in R^{42} or R^{168} space. In general, class separability relies not only on the distribution of samples but also on the classifier to be used. Here two simple but persuasive distance-based classifiers were adopted: MD is the minimum distance classifier, and NN is the nearest neighbor classifier. Another factor that has influence on recognition rate is the number of projection vectors. In each combination of feature extraction algorithm and classifier, all possible numbers of projection vectors were considered before we chose the best as final performance of that combination.

Since the selection of optimal kernel and its associated parameters remains an "engineering problem", we tried different kernels available, including Gaussian RBF, $k(\mathbf{x}, \mathbf{y}) = \exp(-\|(x-y)^2\|/2\sigma^2)$, polynomial kernel, $k(\mathbf{x}, \mathbf{y}) = (1+\mathbf{x}\cdot\mathbf{y})^d$, $d = 2, 3, 4$, and sigmoid kernel, $k(\mathbf{x}, \mathbf{y}) = \tanh(a\mathbf{x}\cdot\ \mathbf{y} - b)$. All the kernel functions fulfill the Mercer's theorem, and their parameters were fine-tuned for best performance. Both the training samples and test samples were projected onto the eigenvector bases (using eq.(9)) to obtain the non-linear discriminant features, which were used in MD and NN classifier respectively. Table 2 summarizes our findings. To each combination of feature extraction algorithm and classifier we give error classification numbers and recognition rates.

Form the experiment results we have the following conclusions:

(1) When the dimension of input space is relatively low ($7 \times 6 = 42$), the performance of OIDA is not better in comparison with traditional Fisher criterion, while its kernel-based variant KOIDA has a slightly higher recognition

rate than Fisher criterion, except for sigmoid kernel and polynomials of degree two.
(2) When the dimension of input space is high ($14 \times 12 = 168$), the performance of Fisher criterion decreases dramatically, OIDA is better than Fisher, and KIODA are the best among the three method, whether the classifier is MD or NN.
(3) Among all the classical kernel functions we have used, Gaussian kernel is the best for classification. The second is polynomial kernel.
(4) All the algorithms work better in 7×6 than in 14×12. This is probably because the dimensionality of 7×6 is 42, which is close to 39, the optimal dimensionality of 40-class problem.

5 Conclusions

A kernel based optimal iterative linear discriminant analysis was given. This is done by formulating the linear transformation in dot-products and then expressing them in kernel functions. Experiments were conducted on our method and several other algorithms and indicated its good performance.

6 Appendix

In order to formulate eq.(4) with only dot-products, we multiply eq.(4) by $\phi^\top(\mathbf{x}_{ij})$

$$\phi^\top(\mathbf{x}_{ij})\mathbf{S}_b^F\mathbf{v} - \lambda_0\phi^\top(\mathbf{x}_{ij})\mathbf{S}_t^F\mathbf{v} = \lambda\phi^\top(\mathbf{x}_{ij})\mathbf{v} \tag{10}$$

Because

$$\mathbf{S}_b^F = \frac{1}{N}\sum_{l=1}^{C} N_l\bar{\phi}_l\bar{\phi}_l^\top \tag{11}$$

Where $\bar{\phi}_l$ is the mean of samples in class l

From eq.(5) and eq.(11) we get

$$\phi^\top(\mathbf{x}_{ij})\mathbf{S}_b^F\mathbf{v} = \tfrac{1}{N}\textstyle\sum_{p=1}^{C}\sum_{q=1}^{N_p}\alpha_{pq}\times$$

$$\textstyle\sum_{l=1}^{C}\left[\sum_{k=1}^{N_l}\phi^\top(\mathbf{x}_{ij})\phi(\mathbf{x}_{lk})\right]\left[\frac{1}{N_l}\right]\left[\sum_{k=1}^{N_l}\phi^\top(\mathbf{x}_{lk})\phi(\mathbf{x}_{pq})\right]$$

For all samples j in all class i, we obtain

$$\begin{bmatrix} \phi^\top(\mathbf{x}_{11}) \\ ... \\ \phi^\top(\mathbf{x}_{CN_C}) \end{bmatrix} \mathbf{S}_b^F v = \frac{1}{N}\mathbf{KWK}\alpha \tag{12}$$

Because

$$\mathbf{S}_t^F = \frac{1}{N}\sum_{l=1}^{C}\sum_{k=1}^{N_C}\phi(\mathbf{x}_{lk})\phi^\top(\mathbf{x}_{lk}) \tag{13}$$

From eq.(5) and eq.(13) we can get

$$\lambda_0\phi^\top(\mathbf{x}_{ij})\mathbf{S}_t^F\mathbf{v} = \tfrac{\lambda_0}{N}\textstyle\sum_{p=1}^{C}\sum_{q=1}^{N_p}\alpha_{pq}\times$$

$$\textstyle\sum_{l=1}^{C}\sum_{k=1}^{N_l}\left[\phi^\top(\mathbf{x}_{ij})\phi(\mathbf{x}_{lk})\right]\left[\phi^\top(\mathbf{x}_{lk})\phi(\mathbf{x}_{pq})\right]$$

Thus

$$\lambda_0\begin{bmatrix} \phi^\top(\mathbf{x}_{11}) \\ ... \\ \phi^\top(\mathbf{x}_{CN_C}) \end{bmatrix} \mathbf{S}_t^F\mathbf{v} = \frac{\lambda_0}{N}\mathbf{KK}\alpha \tag{14}$$

According to eq.(12) and eq.(14), it holds that

$$\begin{bmatrix} \phi^\top(\mathbf{x}_{11}) \\ ... \\ \phi^\top(\mathbf{x}_{CN_C}) \end{bmatrix} \mathbf{S}_b^F\mathbf{v} - \lambda_0\begin{bmatrix} \phi^\top(\mathbf{x}_{11}) \\ ... \\ \phi^\top(\mathbf{x}_{CN_C}) \end{bmatrix} \mathbf{S}_t^F\mathbf{v} \tag{15}$$

$$= \tfrac{1}{N}\mathbf{KWK}\alpha - \tfrac{\lambda_0}{N}\mathbf{KK}\alpha$$

$$\lambda\phi^\top(\mathbf{x}_{ij})\mathbf{v} = \lambda\textstyle\sum_{p=1}^{C}\sum_{q=1}^{N_p}\alpha_{pq}\phi^\top(\mathbf{x}_{ij})\phi(\mathbf{x}_{pq})$$

therefore

$$\lambda \begin{bmatrix} \phi^\top(\mathbf{x}_{11}) \\ ... \\ \phi^\top(\mathbf{x}_{1N_1}) \\ ... \\ \phi^\top(\mathbf{x}_{CN_C}) \end{bmatrix} v = \lambda \mathbf{K}\alpha \tag{16}$$

From eq.(15) and eq.(16) there exits

$$\frac{1}{N}\mathbf{KWK}\alpha - \frac{\lambda_0}{N}\mathbf{KK}\alpha = \lambda \mathbf{K}\alpha$$

That is

$$\lambda = \frac{\alpha^\top(\frac{1}{N}\mathbf{KWK} - \frac{\lambda_0}{N}\mathbf{KK})\alpha}{\alpha^\top \mathbf{K}\alpha}$$

References

B. Moghaddam and A. Pentland, 1997, Probabilistic Visual Learning for Object Representation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.19, no.7, pp. 696-710.

B. Moghaddam, 2002, Principal Manifolds and Probabilistic Subspaces for Visual Recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.24, no.6, pp. 780-788.

Baudat G. and Anouar F., 2000, Genaralized discriminant analysis using a kernel approach, *Neural Computation*, vol.12, pp. 2385-2404.

B. Schölkopf, A. Smola, and K.-R. Müller, 1998, Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Computation*, vol.10, no.5, pp. 1299-1319.

C.J.C. Burges, 1998, a tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, vol.2, no.2, pp. 1-47.

E. Osuna, R. Freund, and F. Girosi, 1997, Training Support Vector Machines: An Application To Face Detection, *in Proc. IEEE Conf. Computer Vison and Pattern Recognition (CVPR)*, pp. 130-136.

M.-H. Yang, 2002, Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods, *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 0215-0220.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müller, 1999, fisher Discriminant Analysis with Kernels, *Neural Networks for Signal Processing*, vol.9, pp. 41-48.

V. Roth and V. Steinhage, 2000, Nonlinear discriminant analysis using kernel function, *NIPS*, vol.12, pp. 568-74.

V.I. Belhumeur, J.P. Hespanha, and D.J. Kriegman,1997, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.19, no.7, pp. 711-720.

Y.-F. Guo, S.-J. Li, J.-Y. Yang, T.-T. Shu, 2003, A generalized Foley-Sammon transform based on generalized fisher discriminant criterion and its application to face recognition, *Pattern Recognition Letters*, vol.24, pp. 147-158.