Research Article

A Fuzzy Kernel Maximum Margin Criterion for Image Feature Extraction

Shibin Xuan^{1,2,3}

¹College of Mathematics and Computer Science, Guangxi University for Nationalities, Nanning 530006, China

²Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Nanning 530006, China

³The China-ASEAN Study Center of Guangxi University for Nationalities, Nanning 530006, China

Correspondence should be addressed to Shibin Xuan; xshibin1997@126.com

Received 12 November 2014; Revised 24 March 2015; Accepted 24 March 2015

Academic Editor: Hari M. Srivastava

Copyright © 2015 Shibin Xuan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Based on kernel principal component analysis, fuzzy set theory, and maximum margin criterion, a novel image feature extraction and recognition method, called fuzzy kernel maximum margin criterion (FKMMC), is proposed. In the proposed method, two new fuzzy scatter matrixes are redefined. The new fuzzy scatter matrix can reflect fully the relation between fuzzy membership degree and the offset of the training sample to subclass center. Besides, a concise reliable computational method of the fuzzy between-class scatter matrix is provided. Experimental results on four face databases (AR, extended Yale B, GTFD, and FERET) demonstrate that the proposed method outperforms other methods.

1. Introduction

The dimensionality reduction is an important research topic in computer vision and pattern recognition for many years [1–3]. As is well known, lots of methods had lower efficiency and limitations in high dimension case. Data transformation is an essential method of dimensionality reduction, which can transform high-dimensional data to relatively low-dimensional space according to certain criterions, and the problem can be solved by an existing method in low-dimensional space. In order to achieve this goal, a variety of approaches were proposed. Most famous methods might be Principal Component Analysis (PCA) [1] and Linear Discriminate Analysis (LDA) [2]. On this basis, a number of improvement algorithms have been proposed.

PCA is a kind of unsupervised learning algorithms; it reflects the overall variability of the data. Each axis has different contribution to this variety. It is well known that the axes corresponding with the larger eigenvalue possess bigger contribution, while the axes corresponding with the smaller eigenvalue often reflected noise or details. Therefore, the axes corresponding with the larger eigenvalue are chosen as transformational operator, and it not only retains the most

useful information of the original image, but also reaches the effect of smoothing and denoising. Due to the fact that PCA is a linear method based on the Gaussian distribution, which is not suitable for non-Gaussian distribution case, for this purpose, the kernel-based principal analytical method (KPCA) [3] was proposed, which is nonlinearly related to the input space. For the aim of dimensionality reduction and data interpretation, a number of principal axis selection and sparse methods have been proposed, such as the latest kernel entropy principal component analysis (KECA) [4], which chooses the principal axes by utilizing a maximized sample density Renyi entropy.

LDA is a traditional supervised learning method for dimensionality reduction, which hopes to be able to obtain a transformation operator by maximizing the between-class distance, while minimizing the within-class distance. Because algorithm needs to compute inverse matrix in its solving process, especially in the small size sample (SSS) situation, the within-class scatter matrix often is singular, which causes LDA to fail to run. To solve the singularity problem, a mass of improved algorithms has been proposed.

The LDA+PCA [5] is a well-known null subspace method, which only calculates the maximum eigenvectors to form

the transformation matrix when the within-class scatter matrix is of full rank; otherwise, it first runs PCA and then performs LDA. The regularized discriminant analysis (RDA) [6] tries to obtain more reliable estimates of the eigenvalues by correcting the eigenvalue distortion with the ridge-type regularization. Penalty discriminant analysis (PDA) [7] desires not only to overcome the small size sample problem but also to smooth the coefficients of discrimination vectors for better interpretation. The Inverse Fisher discriminant analysis (IFDA) [8] modifies the procedure of PCA and derives regular and irregular information from the within-class scatter matrix by inverse Fisher discrimination criterion. Locality Preserving Projections (LPP) [9] are a linear subspace learning method derived from Laplacian Eigenmap, which possesses a significant advantage, and it can generate an explicit map and then minimize the local scatter of the projected data. The local geometrical structure based tensor subspace analysis (TSA) [10] captures an optimal linear approximation to the face manifold in the sense of local isometrics. Maximum margin criterion (MMC) [11] used the difference of both between-class scatter and within-class scatter as discrimination criterion. Linear Laplacian discrimination (LLD) [12] formulates the within-class scatter matrix and the between-class scatter matrix by means of similarity-weighted criterions. The similarities here are computed from the exponential function of pairwise distance in the original sample spaces, which is protected from various forms of metrics. So LLD can be applicable to any linear space for classification. Kernel linear discriminant analysis (KLDA) [13] is equivalent to kernel principal component analysis (KPCA) plus Fisher linear discriminant analysis (LDA). The optimal solution for KLDA is obtained by solving a general eigenvalue problem, but the within-class scatter matrix is often singular. The fuzzy inverse Fisher discriminant analysis (FIFDA) [14] is built on the inverse Fisher discrimination criterion and fuzzy membership degree. In this method, a membership degree matrix is calculated using FKNN, and then the membership degree is incorporated into the definition of the between-class scatter matrix and within-class scatter matrix to get the fuzzy between-class scatter matrix and fuzzy within-class scatter matrix. The 2-dimensional linear discriminant analysis (2DLDA) [15] is based on 2D image matrices; that is, the image matrix does not need to be transformed into a vector. Instead, the image between-class scatter matrix and within-class scatter matrix can be constructed directly from the image matrices, and their eigenvectors are computed for image feature extraction. The Laplacian bidirectional maximum margin criterion (LBMMC) [16] formulates the image total Laplacian matrix, image within-class Laplacian matrix, and image between-class Laplacian matrix using the sample similar weight that is widely used in machine learning. Two-dimensional MMC (2DMMC) has been proposed [17], which aims to find two orthogonal projection matrices to project the original image matrices to a low-dimensional matrix subspace. In the projected subspace, a sample is close to those in the same class but far from those in different classes. Two-dimensional maximum margin criterion (B2D-MMC) [18] introduces a blockwise model for face recognition, performing one-side subspace projection inside each block manifold,

in which a block is close to those belonging to the same class but far from those belonging to different classes. The unilateral projection and the blockwise learning can avoid iterations and alternations as in current bilateral projection based two-dimensional feature extraction approaches and have advantages in complexity and locality. In recent years, representation-based face recognition methods [19, 20] have caused wide public concern in pattern recognition, but they only focus on classification techniques. In this paper, we pay close attention to feature extraction, rather than classification techniques.

In particular, the latter methods focused on embedding weight into the scatter matrix to improve the performance of an algorithm. I think this idea is highly significant because class attribution of training sample has an obvious ambiguity since training samples are not completely separable among the subclasses and are often partial overlapping. Moreover, the kernel and fuzzy approaches are ideal mathematical tools to solve such problems.

In the state of the art, kernel and fuzzy technology cannot be combined with each other, and it is uncertain how to select the bandwidth of the kernel. In the improved algorithms of LDA, weighted scatter matrix cannot reflect sufficiently the interrelation between training sample and subclass prototype. Otherwise, the eigendecomposition of the scatter matrix is influenced by its calculation since summation operator produces a computational error.

In this paper, we propose a kernel fuzzy maximum margin criterion (KFMMC) for feature extraction and recognition. This method is accomplished by means of a two-stage procedure. Firstly, the data are transformed into the kernel subspace by kernel principal component analysis (KPCA) with 98% choosing ratio. Secondly, in order to simplify calculation, we construct the fuzzy between-class scatter matrix and fuzzy within-class scatter matrix on kernel subspace by Euclid distance based basic fuzzy membership. Then the algorithm maximizes the difference of both fuzzy between-class scatter matrix and within-class scatter matrix for obtaining transformational operator. It can integrate efficiently both kernel feature analysis and classification information for achieving dimensional reduction. Our main work can be summarized as in the following several aspects. (1) The proposed algorithm replaces local Laplacian factor in LBMMC with fuzzy membership degree of training samples to sample subclass. It fully embedded fuzzy membership degree into the between-class scatter matrix and the within-class scatter matrix by transforming the between-class scatter matrix, which is different from the fuzzy embedding way of FIFDA. (2) The variance of training samples is used as the bandwidth of the Gaussian kernel, which can avoid effectively uncertainty of parameter settings, and it can meet the properties of Gaussian distribution. In order to remove noise in the training sample and retain initial data information in kernel subspace, we only abandon partial eigenvectors corresponding least eigenvalues in accordance with 2% summation of all eigenvalues. In the kernel subspace generated by KPCA, the algorithm performs the MMC algorithm embedded fuzzy factor. Finally, we can obtain a succinct kernel transformational operator. The proposed

algorithm does not need an iterative procedure as other feature decomposition-based algorithms and does not need to solve inverse matrix. Since it is not necessary to compute the inverse matrix, the small size sample (SSS) problem is alleviated in traditional LDA and its improvements.

The organization of this paper is as follows. In Section 2, the KPCA, MMC, LLD, and FIFDA are reviewed briefly. In Section 3, a new method of embedding fuzzy factor into scatter matrix is presented, and new succinct computing formulas for two scatter matrices are described in detail. In Section 4, the proposed algorithm and its computational complexity (includes training time and testing time) are discussed in detail. In Section 5, experiments are presented to demonstrate the effectiveness of the proposed algorithm. Conclusions are drawn in Section 6.

2. Related Works

For the sake of convenient description, in this section, we introduce simply several corresponding algorithms in connection with our research.

2.1. Kernel Principal Component Analysis (KPCA). For data base $X = [x_1, \dots, x_N]$, where $x_t \in R^d$, $t = 1, \dots, N$, KPCA is a nonlinearly mapping projecting the input space into the kernel feature space, and it can be given by $\phi : R^d \rightarrow F$, satisfying $x_t \rightarrow \phi(x_t)$, $t = 1, \dots, N$. Let $\Phi = [\phi(x_1), \dots, \phi(x_N)]$. A positive semidefinite kernel function, or Mercer kernel, $k_\sigma : R^d \times R^d \rightarrow R$, computes an inner product in Hilbert space:

$$k_\sigma(x_t, x_{t'}) = \langle \phi(x_t), \phi(x_{t'}) \rangle. \quad (1)$$

We may define the $(N \times N)$ kernel matrix K . The element (t, t') of K is to be $k_\sigma(x_t, x_{t'})$. Therefore, $K = \Phi^T \Phi$ is an inner product matrix in F . The kernel matrix K can be eigendecomposed as $K = EDE^T$, where D is a diagonal matrix storing the eigenvalues $\lambda_1, \dots, \lambda_N$, and E is a matrix with the corresponding eigenvectors $e_1, \dots, e_N$ as columns. Then a feature space principal axes u_i , requiring $\|u_i\|^2 = 1$, can be denoted by $u_i = \lambda_i^{-1/2} \Phi e_i$ [3]. Hence,

$$\begin{aligned} P_{u_i} \phi(x) &= u_i^T \phi(x) \\ &= \left\langle \lambda_i^{-1/2} \sum_{t=1}^N e_{i,t} \phi(x_t), \phi(x) \right\rangle \\ &= \lambda_i^{-1/2} \sum_{t=1}^N e_{i,t} k_\sigma(x_t, x), \end{aligned} \quad (2)$$

where $e_{i,t}$ denotes the t th element of e_i . In order to remove noise, after discarding partial principal axes corresponding with the least eigenvalues according to some percentage (commonly 2–5%), the remaining axes constitute projective operator, denoted by U_L ; that is, $U_L = [u_1, \dots, u_L]$, $L < N$, according to the descent order of corresponding eigenvalues. Let $E_L = [e_1, \dots, e_L]$, and $D_L = \text{diag}[\lambda_1, \dots, \lambda_L]$. Let $\Phi_Y =$

$[\phi(y_1, \dots, y_m)]$, $K_Y = \Phi_Y^T \Phi_Y$; then the map of Y on kernel space is $\Phi_Y^{\text{pca}} = P_{U_L} \Phi_Y = D_L^{-1/2} E_L^T K_Y^T$.

On these bases, Jenssen [4] proposed the Renyi entropy based way of choosing kernel feature axes. In this method, $(\sqrt{\lambda_i} e_i^T 1)^2$, $i = 1, \dots, N$ is a sequence in descending order, some axes $e_1, \dots, e_l$ can be selected as the projective operators corresponding to the top l eigenvalues, whereas we find that chosen projective operators according to Renyi entropy are not better than the direct selective eigenvectors corresponding largest eigenvalues for later classification. Therefore, in this paper, we do not use kernel entropy methods.

2.2. Maximum Margin Criterion. Suppose that there are C known pattern classes in training data set X , and N_i is the number of training samples in i th class. The between-class scatter matrix and within-class scatter matrix can be written as (3) and (4), respectively,

$$S_b = \frac{1}{N} \sum_{i=1}^C N_i (m_i - m_0)(m_i - m_0)^T, \quad (3)$$

$$S_w = \frac{1}{N} \sum_{i=1}^C \sum_{j=1}^{N_i} (x_j^i - m_i)(x_j^i - m_i)^T, \quad (4)$$

where N is the total number of training samples, $N = \sum_{i=1}^C N_i$, x_j^i denotes j th training sample in the i th class, m_i is the mean of training samples in i th class, and m_0 refers to the mean of all training samples.

In classical Fisher discriminant analysis, the discrimination criterion is maximizing the ratio of the between-class scatter to the within-class scatter. MMC defined the difference of between-class scatter matrix and within-class scatter matrix as discriminant rule and obtained a transformational matrix W , $W = [w_1, w_2, \dots, w_d]$, and $w_k^T w_k = 1$. The concerned problem can come down to the following constrained optimization:

$$\begin{aligned} \max \quad & \sum_{k=1}^d w_k^T (S_b - S_w) w_k \\ \text{subject to} \quad & w_k^T w_k - 1 = 0 \quad k = 1, \dots, d. \end{aligned} \quad (5)$$

Solving this optimization problem is really the feature decomposition of $S_b - S_w$. The generated eigenvectors are sorted in descending order according to the corresponding eigenvalue. W consists of first d eigenvectors. Comparing with LDA, the main merit of MMC is to avoid calculating inverse within-class scatter matrix. However, the within-class scatter matrix is often the singular matrix.

2.3. Linear Laplacian Discrimination (LLD) [12]. Inspired by the application of Laplacian Eigenmaps in manifold learning and its linearization LPP in clustering and recognition, Zhao et al. [12] proposed Linear Laplacian Discrimination. Its basic

theory can be described as follows. Supposing that M^d is a d -dimension sample space, $\|\cdot\|_{M^d}$ is the Euclidean norm in the original sample space. Weight w_i^j is defined by

$$w_i^j = \exp\left(-\frac{\|x_i^j - m_i\|_{M^d}}{t}\right), \quad i = 1, \dots, N_i. \quad (6)$$

Sign $W^i = \text{diag}(w_1^i, w_2^i, \dots, w_{N_i}^i)$, and 1_{N_i} denotes an all-one column vector of length N_i . Let

$$L_i = W_i - \frac{2}{N_i} W_i 1_{N_i} (1_{N_i})^T + \frac{1_{N_i} W_i (1_{N_i})^T}{N_i^2} 1_{N_i} (1_{N_i})^T. \quad (7)$$

D_i is a 0-1 indicator matrix of i th class and satisfies $X_i = X D_i$, where X_i is a training sample set of i th class. Let

$$L_w = \sum_{i=1}^C D_i L_i D_i^T. \quad (8)$$

Then within-class scatter matrix can be calculated by

$$LS_w = \frac{1}{N} \sum_{i=1}^C \sum_{j=1}^{N_i} w_j^i (x_j^i - m_i) (x_j^i - m_i)^T = X L_w X^T. \quad (9)$$

For between-class scatter matrix, weight w_i is defined as follows:

$$w_i = \exp\left(-\frac{\|m_i - m_0\|_{M^d}^2}{t}\right). \quad (10)$$

Let

$$W_b = \text{diag}(w_1, w_2, \dots, w_C), \\ L_b = W_b - \frac{2}{N_i} W_b 1_{N_i} (1_{N_i})^T + \frac{1_{N_i} W_b (1_{N_i})^T}{N_i^2} 1_{N_i} (1_{N_i})^T, \\ M = [m_1, m_2, \dots, m_C], \quad (11)$$

and then between-class scatter matrix can be defined as

$$LS_b = M L_b M^T. \quad (12)$$

Finally, transformational operator U will be solved satisfying

$$U = \arg \max_U \frac{\text{tr}(U^T LS_b U)}{\text{tr}(U^T LS_w U)} \quad (13) \\ U^T U = I.$$

Similar to what the authors pointed out, like LDA, LLD encounters the computational troubles as well when the within-class scatter matrix is singular. Although the authors proposed some methods to address this issue, this problem was not resolved in essence. Moreover, how to assign parameter t in the expression computing weight w is a problem, and two weights in both within-class scatter matrix and between-class scatter matrix are in disagreement.

2.4. Fuzzy Inverse Fisher Discriminant Analysis (FIFDA). In FIFDA, fuzzy membership degree and each class center are obtained through FKNN algorithm. The fuzzy membership degree of training sample can be computed as follows:

$$u_{ij}^p = \begin{cases} 0.51 + 0.49 \times \left(\frac{n_{ij}}{k}\right) & \text{if } i \text{ is the same as the } j\text{th label of the pattern} \\ 0.49 \times \left(\frac{n_{ij}}{k}\right) & \text{if } i \neq \text{the same as the } j\text{th label of the pattern,} \end{cases} \quad (14)$$

where k denotes neighbor size and n_{ij} is a the number of the neighbors of the j th sample in the i th class. u_{ij} satisfies two obvious properties:

$$\sum_{i=1}^C u_{ij}^p = 1, \\ 0 < \sum_{j=1}^N u_{ij}^p < N. \quad (15)$$

The mean vector of each class is

$$\bar{m}_i = \frac{\sum_{j=1}^N u_{ij}^p x_j}{\sum_{j=1}^N u_{ij}^p}. \quad (16)$$

The corresponding fuzzy within-class scatter matrix and the fuzzy between-class scatter matrix can be defined as (17) and (18), respectively,

$$FS_w = \sum_{i=1}^C \sum_{x_j \in X_i} u_{ij}^p (x_j - \bar{m}_i) (x_j - \bar{m}_i)^T, \quad (17)$$

$$FS_b = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^p (\bar{m}_i - m_0) (\bar{m}_i - m_0)^T, \quad (18)$$

where p is a constant which controls the influence of fuzzy membership degree.

Finally the fuzzy inverse Fisher criterion function can be defined as follows:

$$W_{\text{FIFDA}} = \arg \min_W \frac{\|W^T FS_w W\|}{\|W^T FS_b W\|}. \quad (19)$$

In this method, the fuzzy between-class scatter matrix and fuzzy within-class scatter matrix are redefined according to

FKNN. This method reduces the sensitivity to substantial variations between face images caused by varying illumination, viewing conditions, and facial expression. But we can find out that the embedded way of fuzzy membership degree is not very appropriate in the definition of fuzzy between-class scatter matrix. The main reason is that the fuzzy membership degree reflects relation between a training sample and some class center, but $(\bar{m}_i - m_0)$ expresses the difference between the fuzzy mean of class i and total training sample mean. So it is improper to take u_{ij}^p as the weight of $(\bar{m}_i - m_0)$. Besides, the parameter k in KNN affects also the performance of FIFDA.

3. Fuzzy Maximum Margin Criterion (FMMC)

Based on the statement in Section 2, there are several problems that merit our attention. In the LLD method, the weight is a Gaussian function, and the important property of Gaussian function is smoothing and denoising. But it is uncertain how to assign parameter t . In particular, it cannot provide further classifying information when the training samples overlap. However, fuzzy theory can deal better with this problem. FIFDA method defines fuzzy membership degree with the adjacent properties between training samples. But we can see that there are some artificial factors in the fuzzy membership definition. Moreover, the fuzzy between-class scatter matrix in FIFDA cannot integrate tightly fuzzy membership degree and the samples, and the latter is used to calculate the fuzzy membership degree. In this paper, in order to avoid the uncertainty of fuzzy membership in FIFDA, we employ traditional Euclidean distance based fuzzy membership and redefine fuzzy between-class scatter matrix and within-class scatter matrix. Finally, we give a succinct way of computing new fuzzy between-class scatter matrix.

Suppose there are C known pattern classes in training data set X , N_i is the number of training samples in i th class. N is the number of total training samples. $N = \sum_{i=1}^C N_i$, x_j^i denotes j th training sample in the i th class, m_i is the mean of training samples in i th class, and m_0 refers to total mean of all training samples. u_{ij} is the membership degree of j th training sample to class i . Consider

$$u_{ij} = \frac{\|x_j - m_i\|^{-1}}{\sum_{i=1}^C \|x_j - m_i\|^{-1}}. \quad (20)$$

Corresponding fuzzy within-class scatter matrix can be defined as

$$FS_w = \sum_{i=1}^C \sum_{x_j \in X_i} u_{ij}^2 (x_j - m_i)(x_j - m_i)^T. \quad (21)$$

Let $U_i = \text{diag}\{u_{i1}, u_{i2}, \dots, u_{iN_i}\}$, and 1_{N_i} indicates an all-one row vector of length N_i . P_i satisfies $X_i = XP_i$, where X_i is a matrix with the samples in class i as columns.

Set

$$S_{w,i} = (XP_i - m_i 1_{N_i})U_i. \quad (22)$$

Then

$$FS_w = \sum_{i=1}^C S_{w,i} S_{w,i}^T \quad (23)$$

because

$$\begin{aligned} S_b &= \sum_{i=1}^C N_i (m_i - m_0)(m_i - m_0)^T \\ &= \sum_{i=1}^C N_i \frac{1}{N} \sum_{j=1}^N (m_i - x_j) \frac{1}{N} \sum_{k=1}^N (m_i - x_k)^T \\ &= \frac{1}{N^2} \sum_{i=1}^C N_i \sum_{j=1}^N \sum_{k=1}^N (m_i - x_j)(m_i - x_k)^T. \end{aligned} \quad (24)$$

Therefore, the fuzzy between-class scatter matrix can be defined as follows:

$$FS_b = \frac{1}{N^2} \sum_{i=1}^C N_i \sum_{j=1}^N \sum_{k=1}^N u_{ij} u_{ik} (x_j - m_i)(x_k - m_i)^T. \quad (25)$$

In order to reduce computational complexity, the above fuzzy between-class scatter matrix can be further be simplified as follows:

$$\begin{aligned} FS_b &= \sum_{i=1}^C N_i \frac{1}{N} \sum_{j=1}^N u_{ij} (x_j - m_i) \frac{1}{N} \sum_{k=1}^N u_{ik} (x_k - m_i)^T \\ &= \sum_{i=1}^C \left[\frac{\sqrt{N_i}}{N} \left(\sum_{j=1}^N u_{ij} x_j - m_i \sum_{j=1}^N u_{ij} \right) \right] \\ &\quad \cdot \left[\frac{\sqrt{N_i}}{N} \left(\sum_{k=1}^N u_{ik} x_k - m_i \sum_{k=1}^N u_{ik} \right) \right]^T. \end{aligned} \quad (26)$$

Let $U = \{u_{ij}\}_{C \times N}$, let $M = [m_1, m_2, \dots, m_C]$, and let $N' = \text{diag}\{\sqrt{N_1}/N, \sqrt{N_2}/N, \dots, \sqrt{N_C}/N\}$. Consider

$$Q = (XU^T - M \text{diag}(1_N U^T)) N', \quad (27)$$

and then we have

$$FS_b = QQ^T. \quad (28)$$

Therefore, the fuzzy maximum margin problem can be translated into an objective optimization problem as

$$W = \arg \max_W \sum_{k=1}^d w_k^T (FS_b - FS_w) w_k \quad (29)$$

$$\text{subject to } w_k^T w_k - 1 = 0 \quad k = 1, \dots, d.$$

We can obtain transformational operators from eigendecomposition of $FS_b - FS_w$. That is, firstly, we eigendecompose $FS_b - FS_w$ and then sort the acquired eigenvalues in descending order, which are denoted by $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$,

and corresponding eigenvectors are $w_1, w_2, \dots, w_d$. Finally we obtain transformational operator which consisted of the first eigenvectors according to dimensional reduction requirement.

In the above statements, the standard between-class scatter matrix is unfolded as the expression about the difference between training sample and subclass mean $x_j - m_i$, which is convenient for fuzzy membership degree embeds to between-class scatter matrix. Since fuzzy membership degree u_{ij} reflects the affiliation of sample x_j to subclass i , it is the most efficient way to let fuzzy membership degree u_{ij} act directly on $x_j - m_i$ for reflecting the constraint role of fuzzy membership to sample deviation. It enjoys larger superiority than any other fuzzy scatter matrix definition. The later experiments will reveal the flexibility of the definition.

In calculating fuzzy between-class scatter matrix, the triple accumulation operation is transformed into the succinct matrix operation. It can use effectively the matrix computation superiority provided by MATLAB. In particular, (28) shows that fuzzy between-class scatter matrix can be computed by the product of both matrix Q and its inverse matrix. It can effectively avoid that FS_b is an inexact real symmetric matrix caused by machine precision and computational errors, whereas the scatter matrix obtained by summation is usually inexact real symmetric matrix. In particular, when the number of training samples is very large, this phenomenon easier emerges. However, the real symmetry is very important to ensure that the obtained eigenvalues are real in successive eigendecomposition. In reality, in experimentation on the AR face database, we can find that some eigenvalues are a complex number when we calculated directly fuzzy between-class scatter matrix according to (25). We are not seeking to let this situation occur in feature extraction process. Therefore, our way offers a kind of concise reliable computational method of fuzzy between-class scatter matrix for the other researchers to embed fuzzy factor (or other weights) into the scatter matrix.

4. Fuzzy Kernel Maximum Margin Criterion Based Algorithm for Feature Extraction

In this section, we provide a novel fuzzy kernel maximum margin criterion (FKMMC) for feature extraction, which consists of FMMC and KPCA. In KPCA step, we obtain transformational operator P_K according to 98% selection ratios for eigenvalue. Other eigenvectors are abandoned to achieve the purpose of denoising. The concrete algorithm can be described as follows.

Algorithm 1 (fuzzy kernel maximum margin criterion (FKMMC)).

Step 1. Compute the standard variance of training samples and refer to it as σ .

Step 2. Compute kernel matrix K of training samples by Gaussian kernel function with bandwidth σ .

Step 3. K is eigendecomposed as $K = EDE^T$, $E = [e_1, e_2, \dots, e_N]$, $D = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_N]$, $\lambda_1 \geq \lambda_2 \geq$

$\dots \geq \lambda_N$. We choose first L values with 98%, and let $E_L = [e_1, e_2, \dots, e_L]$ and let $D_L = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_L]$. Then kernel projective operator is as follows: $P_K = D_L^{-1/2} E_L^T$.

Step 4. Projecting original samples into kernel subspace, we have $X^K = P_K K^T$, for convenience sake, yet let $X \triangleq X^K$.

Step 5. First compute total samples mean m_0 and subclass samples mean m_i , $i = 1, \dots, C$ and then compute the fuzzy membership degree of each sample to subclass mean $u_{i,j}$, $i = 1, \dots, C$; $j = 1, \dots, N$ according to (20).

Step 6. Compute fuzzy within-class scatter matrix FS_w and fuzzy between-class scatter matrix FS_b according to (23) and (28).

Step 7. Eigendecompose $FS_b - FS_w$ and then rank eigenvector according to eigenvalues descending order, denoted by $w_1, w_2, \dots, w_L$. We choose first S eigenvectors to reconstruct transformational operator $P_{\text{FMMC}} = [w_1, w_2, \dots, w_S]$ according to the desire of dimensional reduction.

Step 8. Formulate ultimate FKMMC transformational operator: $P_{\text{FKMMC}} = P_{\text{FMMC}}^T P_K$.

Step 9. Output P_{FKMMC} .

Algorithm 2 (FKMMC based recognition algorithm).

Step 1. Input recognized sample $y \in R^d$.

Step 2. Compute $K_y = \phi(y)^T \Phi$.

Step 3. Output $P_{\text{FKMMC}} K_y^T$.

From the above algorithm we can see that the algorithm only includes matrix product, matrix transposition, diagonalization, and eigendecomposition. The eigendecomposed matrixes are all real symmetrical matrixes and the matrix possesses real eigenvalues according to matrix theory. Thus, these matrixes can be all eigendecomposed, and the algorithm does not need to calculate inverse matrix. Therefore, the total computational process is feasible. In Algorithm 1, P_K is a $L \times N$ matrix and P_{FMMC} is a $L \times S$ matrix, so that the output of Algorithm 1 P_{FKMMC} is a $S \times N$ matrix. In Algorithm 2, K_y is a $1 \times N$ vector, and its output is a $S \times 1$ vector. In the image recognition, commonly satisfy $S \ll N \ll d$. Therefore, the proposed algorithm can reduce efficiently the data dimension. Since the kernel map makes data in kernel space to be separated in an easier way, the FMMC can provide more classified information. So the proposed algorithm includes more classified information while data dimension is reduced. On the other hand, the computational complexity of the sample standard variance is $O(N \times d)$, and the kernel matrix K 's is $O(N^2 \times d)$. The computational complexity of eigendecomposition kernel matrix K is $O(N^2)$. The computational complexity of both FS_w and FS_b is all $O(L^3)$. Considering $L < N \ll d$, so the computational

complexity of the proposed algorithm is $O(N^2 \times d)$, that is, the computational complexity of computing the kernel matrix.

5. Experimental Results

In our experiments, four face image databases, namely, AR database, Yale B database, FERET database, and Georgia Tech face database, are used to compare the performance of the proposed fuzzy kernel maximum margin criterion (FKMMC) approach with some other algorithms: KPCA [3], KLDA [13], LLD [12], LPP [9], 2DLDA [15], FIFDA [14], TSA [10], LBMMC [16], 2DMMC [17], and B2DMMC [18]. The experiments are implemented on an Intel(R) Core (TM) 2 Duo CPU T7500 @ 2.20 GHz Dell Computer with 1 G RAM and the programming environment is MATLAB (version 2006a).

5.1. Experiments on Georgia Tech Face Database. The face image database used in our experiments is the Georgia Tech Face Database (GTFD) [21, 22], which consists of 50 subjects with 15 face images available for each subject. These face images vary in size, facial expression, illumination, and rotation along the image plane direction and perpendicular direction to the image plane. In our experiments, all images in the database were manually cropped and resized to 60×40 . After the images were cropped, most of the complex background has been excluded. Also, in-plane rotation was partially eliminated, but the out-of-plane rotation was left untouched. They are further converted to gray level images for both training and testing purposes.

In our first experiment, we choose first k samples per individual for training and the remaining individual for testing, and let $k = 2, 3, 4, 5, 6$, respectively. For each k , KPCA, KLDA, LLD, 2DLDA, LPP, FIFDA, TSA, LBMMC, 2DMMC, B2DMMC, and the proposed FKMMC are used for feature extraction, respectively. In the PCA stage of LLD and FIFDA, the eigenvectors are selected as transformational operator keeping nearly 99% image energy. In the FIFDA, let $p = 2$, and the FKNN parameter k is set as $K = k - 1$. In the LBMMC algorithm, the parameter t of the similarity is set as $t = 100$. In the LPP and TSA algorithms, t is set as default. In the 2DLDA, LPP, TSA, LBMMC algorithms, the selected eigenvectors (projection vectors) are full rank. In the KPCA, KLDA, LLD, and the proposed FKMMC, the number of selected eigenvectors (projection vectors) is 20% of the number of total training samples. In the TSA algorithm, the number of iterations is taken to be 10. In the B2D-MMC algorithm, the number of the layer is set as 6. Finally, a nearest neighbor classifier with Euclid distance is employed. The final results are given in Figure 1. From Figure 1, we can see that the proposed method enjoys the best recognition rate. Although TSA and 2DLDA near the result of the proposed algorithm, TSA needs 20 eigendecomposition for 10 recursions, and 2DLDA needs to calculate the Moore-Penrose pseudo inverse of a matrix. But the proposed algorithm does not need to calculate the inverse of matrix and recursion, and moreover, its stability and true recognition rate are also higher than 2DLDA and TSA.

FIGURE 1: The comparison of the performance of algorithms: KLDA, KPCA, FIFDA, 2DLDA, TSA, LPP, LBMMC, 2DMMC, B2D-MMC, and the proposed algorithm FKMMC on the GTFD face database in the context of first $k = 2, 3, 4, 5, 6$ images are chosen as training samples from every class individually.

In the second experiment, we randomly choose $k = 2, 3, 4, 5$ samples from every individual for training, while the remaining samples are used for testing. The various assumptions in the first experiment will be retained. The test results are reported in Table 1, which lists the average recognition rates crossing 20 runs of each algorithm under the nearest neighbor classifier with Euclid distance metrics and their corresponding standard deviation (std). Table 1 shows that the result of our method is a little better than that of TSA and 2DLDA and is much better than the other methods. The little std shows that our method enjoys more stability. This result further justifies the conclusion of the first experience.

5.2. Experiments on Extended Yale B Database. Extended Yale face database B contains 2535 images of 39 human subjects (each person providing 65 different images) under various poses and illumination conditions. In our experiment, we choose its cropped version images set, which was finished by Lee et al. [23]. All images were resized to 60×40 .

In the experiment, we choose randomly $k = 2, 3, 4, 5$ samples from every individual for training and the remaining images for testing. KPCA, KLDA, LLD, 2DLDA, LPP, FIFDA, TSA, LBMMC, 2DMMC, B2D-MMC, and the proposed FKMMC are used for feature extraction. In the PCA stage

TABLE 1: Average recognition rates and standard deviation on the GTFD face database for sample numbers per class $kk = 2, 3, 4, 5$.

Algorithm (feature dim.)	$kk = 2$	$kk = 3$	$kk = 4$	$kk = 5$
LPP	32.41 (2.94)	35.95 (2.70)	39.82 (2.52)	40.84 (3.13)
TSA	49.32 (6.44)	65.51 (2.75)	70.90 (3.30)	73.73 (3.96)
KPCA (35)	52.56 (2.05)	58.45 (4.04)	62.00 (2.61)	65.46 (2.60)
KLDA (35)	25.95 (2.90)	40.37 (7.55)	35.74 (5.63)	43.11 (2.43)
2DLDA (120)	47.91 (6.01)	54.92 (3.13)	56.74 (3.69)	59.83 (2.32)
LLD (35)	52.12 (3.09)	59.73 (2.17)	62.00 (3.07)	63.22 (3.41)
FIFDA (35)	21.72 (3.91)	30.47 (2.94)	33.45 (3.94)	37.58 (2.41)
LBMMC	54.47 (2.59)	59.81 (3.28)	65.17 (1.39)	66.60 (2.71)
2D-MMC (120)	57.23 (3.05)	60.71 (2.75)	63.00 (2.56)	65.12 (2.56)
B2D-MMC (120)	50.49 (2.74)	56.39 (3.35)	57.32 (2.47)	58.79 (2.94)
FKMMC (35)	52.58 (4.87)	61.18 (4.21)	69.53 (3.62)	74.25 (2.38)

TABLE 2: Average recognition rates and standard deviation on the extended Yale B face database for sample numbers per class $kk = 2, 3, 4, 5$.

Algorithm (feature dim.)	$kk = 2$	$kk = 3$	$kk = 4$	$kk = 5$
LPP	63.75 (12.08)	81.88 (9.86)	84.37 (7.10)	90.51 (5.67)
TSA	46.45 (14.13)	57.68 (17.40)	59.35 (15.84)	56.10 (10.41)
KPCA (35)	34.03 (9.23)	35.85 (9.10)	38.85 (7.36)	37.83 (6.10)
KLDA (35)	54.70 (12.48)	73.18 (12.6)	84.55 (9.40)	89.17 (8.41)
2DLDA (120)	46.96 (8.72)	56.87 (7.27)	65.11 (6.18)	63.71 (5.02)
LLD (35)	30.11 (7.96)	36.58 (11.33)	38.26 (9.29)	39.38 (8.95)
FIFDA (35)	26.23 (3.78)	28.85 (3.98)	30.91 (2.04)	35.44 (3.08)
LBMMC	38.21 (8.72)	37.12 (10.27)	45.43 (9.71)	47.86 (6.79)
2D-MMC (120)	42.00 (9.07)	48.26 (11.44)	54.89 (8.99)	56.92 (9.17)
B2D-MMC (120)	33.36 (8.04)	40.66 (8.35)	40.32 (8.52)	42.93 (6.26)
FKMMC (35)	60.63 (12.63)	76.53 (9.77)	84.92 (6.29)	90.91 (5.05)

of LLD, FIFDA, the eigenvectors are selected as transformational operator keeping nearly 99% image energy. In the LBMMC algorithm, the parameter t of the similarity is set as $t = 100$, and in the LPP and TSA algorithms, t is set as default. In the 2DLDA, LPP, TSA, and LBMMC algorithms, the selected eigenvectors (projection vectors) are full rank. In the KPCA, KLDA, LLD, and the proposed FKMMC, the number of selected eigenvectors (projection vectors) is 15% of the number of total training samples. In the TSA algorithm, the number of iterations is taken to be 10. In the B2D-MMC algorithm, the number of the layers is set as 6. Finally, a nearest neighbor classifier with Euclid distance is employed. The final results are given in Table 2 and Figure 2. Just like you see, the proposed method has the best recognition rate.

Table 2 and Figure 2 show that our proposed algorithm shows better performance as compared with other algorithms on the extended Yale B database. We can see also that the results of both 2DLDA and KLDA are nearer to our algorithm, but 2DLDA needs to compute the Moore-Penrose pseudo inverse of a matrix which costs more calculation time than matrix multiplication. At the same time, we also see that the performance of TSA is not perfect on the extended Yale B database although TSA has a good performance on the Georgia Tech face database, and so our algorithm is more stable than TSA.

5.3. Experiments on AR Database and FERET Face Database.

The AR face database [24] was created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the U.A.B. It contains over 3300 color images corresponding to 126 people's faces (70 men and 56 women). Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarves). The pictures were taken at the CVC under strictly controlled conditions. No restrictions on wear (clothes, glasses, etc.), makeup, hair style, and so forth were imposed on participants. Each person participated in two sessions, separated by two weeks (14 days). The same pictures were taken in both sessions. In our experiments, each image was manually cropped and resized to 60×40 .

The FERET face database [25] is from the FERET Program sponsored by the US Department of Defense's Counterdrug Technology Development Program through the Defense Advanced Research Projects Agency (DARPA), and it has become the de facto standard for evaluating state-of-the-art face recognition algorithms. The whole database contains 13,539 face images of 1565 subjects taken during different photo sessions with variations in size, pose, illumination, facial expression, and even age. The subset we use in our experiments includes 200 subjects each with four different images. All images are obtained by cropping based on the

FIGURE 2: The comparison of the performance of algorithms: LPP, TSA, KPCA, KLDA, 2DLDA, LLD, FIFDA, LBMMC, 2D-MMC, B2D-MMC, and the proposed algorithm FKMMC on the Yale B face database in the context of $k = 2, 3, 4, 5$ images are chosen as training samples from every class individually.

FIGURE 3: The comparison of the performance of algorithms: LPP, TSA, KPCA, KLDA, 2DLDA, LLD, FIFDA, LBMMC, 2D-MMC, B2D-MMC, and the proposed algorithm FKMMC on the AR face database in the context of $k = 2, 3, 4, 5$ images are chosen as training samples from every class individually.

manually located centers of the eyes and are normalized to the same size of 40×40 with 256 gray levels.

We repeat the second experiment in Section 5.2 and choose randomly $k = 2, 3, 4, 5$ samples from every individual for training on the AR face database and three samples from every individual for training on the FERET face database. The results are reported in Tables 3 and 4, respectively, and are shown in Figures 3 and 4. On the AR face database, the proposed algorithm is not obviously advantage than KLDA and TSA for true recognition rate, but our algorithm enjoys a lower std. This result shows that our algorithm has more opportunities to get high recognition rate when the number of testing samples is larger. On FERET face database, our algorithm is evidently superior to other algorithms at both true mean recognition rate and std.

5.4. Friedman Test and Nemenyi Test. In order to compare with relative recognition methods, we use Friedman test and Nemenyi test [26, 27]. The Friedman test is a nonparametric equivalent of the repeated-measures ANOVA [27]. It ranks the algorithms for each data set separately; the best performing algorithm gets the rank of 1, and the second best rank is 2, ..., as shown in Table 5. Let r_i^j be the rank of the j th of k algorithms on the i th of N data sets. The Friedman test compares the average ranks of algorithms, $R_j =$

$(1/N) \sum_i r_i^j$. Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks R_j should be equal, the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (30)$$

is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and K are big enough (as a rule of a thumb, $N > 10$ and $k > 5$), and a derived statistic is

$$F_F = \frac{(N-1) \chi_F^2}{N(k-1) - \chi_F^2}. \quad (31)$$

In my experiments, the ranks of each method and its average ranks are listed in Table 5. Obviously, the methods number $k = 11$, the number of experiments $N = 16$ in this paper, and corresponding F_F is distributed according to the F distribution with $11-1 = 10$ and $(10-1) \times (16-1) = 150$ degrees of freedom. The critical value of $F(10, 150)$ for $\alpha = 0.01$ is 2.4412. Since the F_F is 16.2241, The Friedman test rejects the null-hypothesis. In order to check how the performance of two methods is significantly different, we

TABLE 3: Average recognition rates and standard deviation on the AR face database for sample numbers per class $kk = 2, 3, 4, 5$.

Algorithm (feature dim.)	$kk = 2$	$kk = 3$	$kk = 4$	$kk = 5$
LPP	34.83 (4.24)	44.68 (6.24)	48.36 (7.60)	55.61 (6.90)
TSA	58.60 (6.43)	63.65 (7.71)	65.21 (10.18)	70.03 (10.07)
KPCA (35)	42.10 (12.37)	47.43 (8.94)	53.63 (5.05)	59.56 (8.18)
KLDA (35)	32.09 (9.20)	58.89 (4.93)	68.48 (9.62)	70.18 (8.22)
2DLDA (120)	40.14 (5.14)	41.94 (4.94)	44.77 (3.89)	44.78 (5.30)
LLD (35)	45.53 (7.80)	46.18 (7.37)	51.97 (5.96)	49.79 (6.67)
FIFDA (35)	26.19 (1.94)	32.25 (2.17)	37.21 (3.16)	40.66 (4.18)
LBMMC	48.32 (13.29)	57.99 (11.87)	60.99 (7.08)	62.47 (7.50)
2D-MMC (120)	39.41 (7.95)	43.88 (5.24)	44.52 (5.03)	47.71 (3.07)
B2D-MMC (120)	30.10 (7.22)	30.11 (7.43)	34.12 (6.95)	35.58 (4.30)
FKMMC (35)	53.21 (12.86)	62.90 (8.61)	68.51 (9.37)	70.51 (11.25)

TABLE 4: Average recognition rates and standard deviation on the FERET face database for sample numbers per class $kk = 2, 3, 4, 5$.

Algorithm (feature dim.)	$kk = 2$	$kk = 3$	$kk = 4$	$kk = 5$
LPP	6.96 (2.40)	6.60 (2.49)	5.86 (2.68)	5.66 (2.97)
TSA	52.43 (8.07)	63.93 (10.24)	67.56 (14.22)	69.82 (13.22)
KPCA (35)	45.02 (7.74)	51.52 (10.74)	55.40 (8.50)	57.01 (14.57)
KLDA (35)	8.69 (3.26)	37.76 (9.73)	57.55 (15.16)	66.73 (19.70)
2DLDA (80)	28.08 (5.36)	29.60 (5.89)	30.15 (6.84)	32.54 (11.10)
LLD (35)	43.94 (8.25)	49.60 (10.56)	51.92 (8.33)	55.61 (14.52)
FIFDA (35)	4.11 (0.56)	10.35 (3.61)	16.77 (5.18)	23.37 (9.10)
LBMMC	50.00 (9.33)	58.93 (10.10)	60.34 (11.59)	63.70 (15.38)
2D-MMC (80)	56.57 (8.68)	65.63 (7.86)	65.70 (7.78)	69.29 (11.61)
B2D-MMC (80)	54.29 (9.10)	66.68 (7.61)	65.34 (10.42)	70.32 (12.17)
FKMMC (35)	57.04 (9.71)	66.74 (10.45)	67.65 (14.90)	70.45 (17.54)

TABLE 5: The ranks and mean ranks on four face databases: GTFD, Yale B, AR, and FERET for sample numbers per class $kk = 2, 3, 4, 5$.

	GTFD				Yale B				AR				FERET				Average R	$R_j - R$
kk	2	3	4	5	2	3	4	5	2	3	4	5	2	3	4	5		
LPP	9	10	9	10	1	1	3	2	8	7	7	6	10	11	11	11	7.25	5.75
TSA	7	1	1	2	5	4	5	6	1	1	3	3	4	4	2	3	3.25	1.75
KPCA	4	6	5	4	8	10	9	10	5	5	5	5	6	6	7	7	6.375	4.875
KLDA	10	9	10	9	3	3	2	3	9	3	2	2	9	8	6	5	5.813	4.313
2DLDA	8	8	8	7	4	5	4	4	6	9	8	9	8	9	9	9	7.188	5.688
LLD	5	5	6	6	10	9	10	9	4	6	6	7	7	7	8	8	7.063	5.563
FIFDA	11	11	11	11	11	11	11	11	11	10	10	10	11	10	10	10	10.63	9.53
LBMMC	2	4	3	3	7	8	7	7	3	4	4	4	5	5	5	6	4.813	3.313
2D-MMC	1	3	4	5	6	6	6	5	7	8	9	8	2	3	3	4	5	3.5
B2D-MMC	6	7	7	8	9	7	8	8	10	11	11	11	3	2	4	2	7.125	5.625
FKMMC	3	2	2	1	2	2	1	1	2	2	1	1	1	1	1	1	1.5	

can proceed with a post hoc test and the Nemenyi multiple comparison test. A critical difference (CD) is defined by

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}}. \quad (32)$$

We use $\alpha = 0.1$ and get $q_{0.1} = 2.978$ for comparisons among eleven methods. The CD is 3.492. The Nemenyi test is shown in Figure 5. In the figure, the mean rank of each method

is denoted by a circle. The horizontal bar, which is across the circle, indicates the “critical difference.” Two methods are significantly different if two bars not overlapping in horizontal direction; otherwise it means the two methods are similar in the ranks. For the recognition results, the proposed method always ranks the 1st among the competitors. There is no significant difference between FKMMC and TSA, but only a half overlapping. We can see that the proposed method presents significant advantage compared to other

FIGURE 4: The comparison of the performance of algorithms: LPP, TSA, KPCA, KLDA, 2DLDA, LLD, FIFDA, LBMMC, 2D-MMC, B2D-MMC, and the proposed algorithm FKMMC on the FERET face database in the context of $k = 2, 3, 4, 5$ images are chosen as training samples from every class individually.

methods besides TSA. In particular, the proposed method is improvement over LBMMC, but its bars are not overlapping in horizontal direction. This shows my improvement is a meaningful work.

From the above, we can see that the proposed method enjoy better performance than other competitors. In the proposed method, the kernel technique is used to enhance the separability of samples set, and we take the fuzzy set theory to reduce the sensitivity to substantial variations between face images caused by varying illumination, viewing conditions, and facial expression since the fuzzy membership degree can reflect relation between a training sample and some class center. Using the two techniques, the proposed method FKMMC improves markedly the performance of the original method LBMMC in two respects of true recognition rate and training time. Although it is seen that the proposed method costs more running time than LPP, 2DLDA, LLD, 2D-MMC, and B2D-MMC from Table 6, its average rank has significant advantage than that of those methods. For kernel approaches, the average training time and the test time of the proposed method are lower than those of KPCA and KLDA due to the fact that our method adapted a new way of calculating fuzzy kernel scatter matrix.

FIGURE 5: Nemenyi test results of algorithms: LPP, TSA, KPCA, KLDA, 2DLDA, LLD, FIFDA, LBMMC, 2D-MMC, B2D-MMC, and the proposed algorithm FKMMC.

6. Conclusion

In the pattern recognition, the feature extraction techniques are widely employed to reduce the dimensionality of data and to enhance the discriminatory information. In this paper, fuzzy kernel maximum margin criterion method is proposed. The proposed method absorbs efficiently the advantages of both the kernel method and maximum margin criterion and redefines the fuzzy between-class scatter matrix. The new fuzzy scatter matrix can fully reflect the relation between fuzzy membership degree and the offset of the training sample to subclass center. The new methods can effectively extract the most discriminatory information while they achieve dimensional reduction and do not suffer from the small sample size problem. The final transformational operator is a $S \times N$ matrix. In image recognition, the number of training samples N is far smaller than the sample dimension d . Therefore, the proposed method is faster than non-kernel method LBMMC if we do not consider the time cost of computing kernel projecting. The experimental results show that the proposed method in this paper is effective and robust. In particular, the definition of fuzzy between-class scatter matrix can offer a kind of concise reliable computational method for the other researchers hoping to embed fuzzy factor (or other weights) into the scatter matrix.

Notations

- KPCA: Kernel principal component analysis
- MMC: Maximum margin criterion
- LLD: Linear Laplacian discrimination
- FIFDA: Fuzzy inverse Fisher discriminant analysis
- FKNN: Fuzzy kernel nearest neighboring
- LBMMC: Laplacian bidirectional maximum margin criterion
- SSS: The small size sample
- KECA: The latest kernel entropy principal component analysis

TABLE 6: The average training times and the average testing times of eleven methods on four face databases: GTFD, Yale B, AR, and FERET for sample numbers per class $kk = 2, 3, 4, 5$.

	GTFD		Yale B		AR		FERET	
	Train (s)	Test (s)	Train (s)	Test (s)	Train (s)	Test (s)	Train (s)	Test (s)
LPP	0.3985	0.00038	0.2104	0.00031	1.5729	0.00077	3.5654	0.0012
TSA	2.1942	0.0031	1.6920	0.0023	4.6643	0.0065	4.7604	0.0079
KPCA	0.4531	0.0042	0.2843	0.0032	6.2557	0.0099	17.0743	0.0113
KLDA	0.5560	0.0042	0.2990	0.0031	6.373	0.0097	17.9793	0.0114
2DLDA	0.0711	0.0007	0.0991	0.00054	0.1917	0.0016	0.1390	0.0027
LLD	0.3056	0.00035	0.2064	0.00028	2.5272	0.00068	10.0023	0.0011
FIFDA	1.0285	0.00058	0.3920	0.00044	10.2263	0.0011	47.3983	0.0016
LBMMC	17.0785	0.0018	8.7614	0.0014	86.9606	0.0040	178.0315	0.0055
2D-MMC	0.1071	0.0010	0.1647	0.00079	0.3161	0.0024	0.2689	0.0037
B2D-MMC	0.2024	0.00084	0.2450	0.00064	0.4989	0.0018	0.6488	0.0029
FKMMC	0.4472	0.0013	0.2574	0.00098	5.4658	0.0033	15.4058	0.0040

RDA: Regularized discriminant analysis
PDA: Penalty discriminant analysis
LPP: Locality preserving projections
2DLDA: Two-dimensional linear discriminant analysis
B2D-MMC: Blockwise two-dimensional maximum margin criterion
KFMMC: Kernel fuzzy maximum margin criterion.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research is supported by the National Science Foundation Council of Guangxi (2012GX NSFAA053227).

References

- [1] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [2] M. Zhu and A. M. Martinez, "Subclass discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1274–1286, 2006.
- [3] T.-J. Chin and D. Suter, "Incremental kernel principal component analysis," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1662–1674, 2007.
- [4] R. Jenssen, "Kernel entropy component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 847–860, 2010.
- [5] L. F. Chen, H. Y. M. Liao, M. T. Ko, J. C. Lin, and G. J. Yu, "New LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.
- [6] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, 1989.
- [7] T. Hastie and R. Tibshirani, "Penalized discriminant analysis," *The Annals of Statistics*, vol. 23, no. 1, pp. 73–102, 1995.
- [8] X.-S. Zhuang and D.-Q. Dai, "Inverse Fisher discriminate criteria for small sample size problem and its application to face recognition," *Pattern Recognition*, vol. 38, no. 11, pp. 2192–2194, 2005.
- [9] J. Yang, D. Zhang, and J. Y. Yang, "Face recognition using Laplacian faces," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 650–664, 2007.
- [10] X. F. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems*, vol. 18, MIT Press, Vancouver, Canada, 2005.
- [11] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 157–165, 2006.
- [12] D. Zhao, Z. C. Lin, R. Xiao, and X. O. Tang, "Linear laplacian discrimination for feature extraction," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.
- [13] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000.
- [14] W. K. Yang, J. G. Wang, M. W. Ren, L. Zhang, and J. Y. Yang, "Feature extraction using fuzzy inverse FDA," *Neurocomputing*, vol. 72, no. 13–15, pp. 3384–3390, 2009.
- [15] X.-Y. Jing, H.-S. Wong, and D. Zhang, "Face recognition based on 2D Fisherface approach," *Pattern Recognition*, vol. 39, no. 4, pp. 707–710, 2006.
- [16] W. K. Yang, J. G. Wang, M. W. Ren, J. Y. Yang, L. Zhang, and G. H. Liu, "Feature extraction based on Laplacian bidirectional maximum margin criterion," *Pattern Recognition*, vol. 42, no. 11, pp. 2327–2334, 2009.
- [17] Q. Gu and J. Zhou, "Two dimensional maximum margin criterion," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 1621–1624, Taipei, Taiwan, April 2009.
- [18] X.-Z. Liu and G. Yang, "Block-wise two-dimensional maximum margin criterion for face recognition," *The Scientific World Journal*, vol. 2014, Article ID 875090, 9 pages, 2014.
- [19] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image

- classification,” *International Journal of Computer Vision*, vol. 109, no. 3, pp. 209–232, 2014.
- [20] J. Xu, G. Yang, H. Man, and H. He, “ L_1 graph based on sparse coding for feature selection,” in *Advances in Neural Networks—ISNN 2013*, vol. 7951 of *Lecture Notes in Computer Science*, pp. 594–601, Springer, Berlin, Germany, 2013.
- [21] Georgia Tech Face Database, <http://www.anefian.com/research/gt.db.zip>.
- [22] L. Chen, H. Man, and A. V. Nefian, “Face recognition based on multi-class mapping of Fisher scores,” *Pattern Recognition*, vol. 38, no. 6, pp. 799–811, 2005.
- [23] K.-C. Lee, J. Ho, and D. J. Kriegman, “Acquiring linear subspaces for face recognition under variable lighting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.
- [24] A. M. Martinez and R. Benavente, “The AR face database,” CVC Technical Report 24, 1998.
- [25] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [26] J. Xu, H. He, and H. Man, “DCPE co-training for classification,” *Neurocomputing*, vol. 86, pp. 75–85, 2012.
- [27] J. Demmar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.