A Convergent Solution to Matrix Bidirectional Projection Based Feature Extraction with Application to Face Recognition *

Yubin Zhan[†], **Jianping Yin** , **Xinwang Liu**

School of Computer, National University of Defense Technology,
No 137, Yanwachi Street, Kaifu District,
Changsha, Hunan Province, 410073, P.R. China
E-mail: {*YubinZhan,JPYin,XWLiu*}*@nudt.edu.cn*

Received 13 May 2010
Accepted 18 March 2011

Abstract

Recently, many feature extraction methods, which are based on the matrix representation of image and matrix bidirectional projection technique, are proposed. However, these methods in solving the two projection matrices will suffer from non-optimized or non-convergent solution. To overcome this problem, a novel feature extraction method which exploits the Maximum Margin Criterion is proposed, where an iterative optimization algorithm is designed to compute the two projection matrices. A noteworthy property of the proposed iterative solution algorithm is that it can monotonously increase the optimization objective, i.e., the bidirectional projection margin. According to this property, we further theoretically prove that the objective value and the solution are convergent. Moreover, the proposed method can automatically determine suitable feature dimensionality to obtain competitive recognition performance. Extensive and systematic experiments on CMU PIE and Yale face databases demonstrate the high convergence speed of the proposed iterative optimization procedure, as well as the superiority of the proposed feature extraction method over other state-of-the-art approaches in face recognition.

Keywords: feature extraction, dimensionality reduction, maximum margin criterion, face recognition

1. Introduction

Feature extraction plays the central role in machine learning and pattern recognition. Many Dimensionality Reduction based feature extraction methods have been developed over the past few decades. Among these methods, Eigenfaces[1] and Fisherfaces[2], whose underlying ideas are Principal Component Analysis(PCA) and Linear Discriminant Analysis(LDA), respectively, are two of the most popular ones.

Since unsupervised learning cannot properly extract the most discriminative feature, in most applications, LDA is proved to be much more effective than PCA. However, it cannot be applied directly to small sample size problem(SSS)[3], because the within-class scatter matrix is singular. Many approaches have been proposed to solve SSS problem of LDA, such as PCA+LDA[2], Regularized Discriminant Analysis(RDA)[4], LDA+PCA[5], Penalized Discriminant Analysis(PDA)[6,7]. Maximum Mar-

*This work is supported by Natural Science Foundation of China (60970034, 60603015), and the Foundation for Author of National Excellent Doctoral Dissertation(Grant No. 2007B4).

[†] Corresponding author, email: Yubinzhan@nudt.edu.cn

gin Criterion(MMC)[8], which uses the difference of between-class scatter and within-class scatter as discriminant criterion, is another simple, efficient and stable method that can alleviate the SSS problem in LDA. Geometrically, MMC maximizes the average margin between classes.

Recent studies have shown that face images possibly reside on a nonlinear submanifold[9,10,11,12,13]. Some nonlinear techniques have been proposed to discover the nonlinear structure of the manifold, e.g., Isomap[12], Locally Linear Embedding(LLE)[11], Laplacian Eigenmap[9], and Local Tangent Space Alignment(LTSA)[14]. Since these manifold learning algorithms have no explicit mechanism to deal with unseen samples, it is not applicable in the real face recognition problem. Many linearized method based on these algorithms have been proposed including Neighborhood Preserving Embedding(NPE)[15], Linear Local Tangent Space Alignment(LLTSA)[16]. He et al. proposed the Locality Preserving Projection (LPP)[17], which is a linear dimensionality reduction method derived from Laplacian Eigenmap. And Laplacianfaces[10] which applies LPP for face feature extraction attempts to preserve the local geometric structure of the face image set. However, it is an unsupervised method in nature as Eigenfaces.

The above mentioned methods need to convert the image matrix into a vector. This will aggravate the "the curse of dimensionality", increase the computation complexity, and lose some useful structural information of images. Thereby, based on direct matrix representation of face image, 2-dimensional PCA[18], 2-dimensional LDA[19] and 2D Laplacianfaces[20] are proposed . A shortcoming of these 2D methods is that the extracted features still have related much high dimensionality for classification. Zuo et al. proposed bidirectional PCA[21] which exploits both image matrix column projection and row projection to further reduce the dimensionality of extracted features. However, the column projection matrix and row projection matrix, which in fact depend on each other, are computed in a separate way. This will result in non-optimized solution.

This problem also exists in the Laplacian Bidirectional Maximum Margin Criterion(LBMMC) method recently proposed by Yang et. al[22]. In LBMMC, the maximum margin criterion is adopted. And to explicitly exploit the geometry structure(i.e. manifold) of data, the Laplacian scatter matrix is defined by incorporating the Laplacian similarity as weight into the standard scatter matrix. LBMMC aims to obtain the optimal projection matrices via maximizing the projection margin, which depends on the projection matrices in both directions. For the simplicity of solution, however, LBMMC only considers projection in one direction when maximizing the projection margin and optimizes the two projection matrices in a separate way.

An improved method is Tensor Subspace Analysis(TSA)[23], in which the Fisher Discriminant Criterion is used and the projection matrices are jointly optimized in an iterative procedure. A problem of TSA is that the convergence of the iterative solution is not guaranteed.

To overcome the aforementioned problems in existing feature extraction methods based on matrix(tensor) bidirectional projection, in this paper, we redefined the Laplacian bidirectional projection margin in LBMMC respecting influence of projections in both directions. And an iterative optimization algorithm to maximize the defined projection margin is proposed, followed by a detailed convergency proof of the solution projection matrices and the projection margin. Moreover, the proposed method can automatically select suitable feature dimensionality for face recognition.

It is worthwhile to highlight several aspects of the proposed approach here:

1. Although the solution is obtained in an iterative manner, the iterative procedure is convergent, and experiments demonstrate that the solution will convergence after a few number of iterations.

2. The computation of our method is very simple. Based on the matrix representation of image in each iteration, we only need to conduct eigenvalue decomposition of matrix with small size, which is far smaller than that in vector-based methods. Moreover, the matrix representation of image can preserve structural information which is ignored by the vector-based methods.

3. In our method, the manifold structure of the image space, which is modeled by an adjacency graph, is explicitly taken into account.

4. Our method can automatically select suitable feature dimensionality with which algorithm can obtain comparable recognition performance. This is very important in practice. The previous methods need to consider all the possible dimensionality to obtain the top recognition performance. This is very time-consuming and inapplicable in real face recognition system.

2. Laplacian scatter matrix

Let matrix x represent an image with $m \times n$ pixels, then feature matrix y of image x can be obtained by:

$$y = U^T xV \tag{1}$$

where U and V are $m \times m'(m' \leqslant m)$ column projection matrix and $n \times n'(n' \leqslant n)$ row projection matrix, respectively.

Suppose we are given N training images $X = [x_1, x_2, \ldots, x_N] = [X_1, \ldots, X_i, \ldots, X_c] = [x_1^{(1)}, \ldots, x_j^{(i)}, \ldots, x_{N_c}^{(c)}]$ which belong to c different classes, the ith class has N_i images($\sum_{i=1}^{c} N_i = N$) and matrix $X_i = [x_1^{(i)}, x_2^{(i)}, \ldots, x_{N_i}^{(i)}]$ consists of the image matrices from the ith class. By representing each image matrix as an m-set of row vectors, the row total scatter matrix can be expressed as:

$$\begin{aligned} S_t^{\text{row}} &= \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{m} (x_{ij} - x^{(j)})^T (x_{ij} - x^{(j)}) && (2) \\ &= \frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})^T (x_i - \overline{x}) && (3) \\ &= \frac{1}{2N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (x_i - x_j)^T (x_i - x_j) && (4) \end{aligned}$$

where $\overline{x}$, x_{ij} and $x^{(j)}$ are the mean matrix of all training images, jth row vector of ith image matrix and mean vector of jth row vector of all training images, respectively.

The row within-class scatter matrix can be defined as:

$$\begin{aligned} S_w^{\text{row}} &= \frac{1}{N} \sum_{i=1}^{c} \sum_{j=1}^{N_i} (x_j^{(i)} - m_i)^T (x_j^{(i)} - m_i) && (5) \\ &= \frac{1}{N} \sum_{i=1}^{c} \frac{1}{2N_i} \sum_{j,k=1}^{N_i} (x_j^{(i)} - x_k^{(i)})^T (x_j^{(i)} - x_k^{(i)}) && (6) \end{aligned}$$

where $x_j^{(i)}$ and m_i are the jth image matrix of ith class and mean matrix of ith class, respectively.

The use of manifold information in feature extraction has shown the state-of-the-art face recognition performance[10,23,24]. According to graph embedding theory[25], we define an undirected weighted graph $G(X,W)$ to characterize the nonlinear manifold structure of the image set X. The real symmetric matrix W measures similarities of any pairs of samples. It can be constructed using various similarity criterion, such as Gaussian similarity in Laplacian eigenmap[9], local neighborhood relationship as in LLE[11] and also prior class information in supervised learning algorithms. Here, the Gaussian similarity is adopted:

$$w_{ij} = \exp(-\|x_i - x_j\|^2 / (2\sigma^2)) \tag{7}$$

Then in order to incorporate the nonlinear manifold structure of face images, we can define the following row total Laplacian scatter:

$$\begin{aligned} LS_t^{\text{row}} &= \frac{1}{2N^2} \sum_{i,j=1}^{N} w_{ij} (x_i - x_j)^T (x_i - x_j) && (8) \\ &= \frac{1}{N^2} \sum_{i,j=1}^{N} (w_{ij} x_i^T x_i - w_{ij} x_i^T x_j) && (9) \\ &= \frac{1}{N^2} X'^T (L \otimes I_m) X' && (10) \end{aligned}$$

where $X' = [x_1^T, x_2^T, \ldots, x_N^T]^T$, D is a diagonal matrix with $d_{ii} = \sum_{j=1}^{N} w_{ij}$, $L = D - W$ is the Laplacian matrix of graph G, I_m is identity matrix of order m and operator $\otimes$ is the Kronecker product of matrices.

Similarly, in row direction, the image within-class Laplacian scatter matrix is:

$$\begin{aligned}LS_w^{\text{row}} =& \frac{1}{N}\sum_{i=1}^{c}\frac{1}{2N_i}\sum_{j,k=1}^{N_i} w_{jk}^{(i)}(x_j^{(i)}-x_k^{(i)})^T(x_j^{(i)}-x_k^{(i)})\\ =& \frac{1}{N}\sum_{i=1}^{c}\frac{1}{N_i}X_i'^T(L_w^{(i)}\otimes I_m)X_i' \qquad (11)\\ =& \frac{1}{N}X'^T(L_w\otimes I_m)X' \qquad (12)\end{aligned}$$

where $w_{jk}^{(i)} = \exp(-\|x_j^{(i)}-x_k^{(i)}\|^2/(2\sigma^2))(1 \leqslant i \leqslant c, 1 \leqslant j,k \leqslant N_i)$, $X_i' = [x_1^{(i)T}, x_2^{(i)T}, \ldots, x_{N_i}^{(i)T}]^T$, $L_w^{(i)}$ is the Laplacian matrix of ith class and $L_w = \text{diag}(L_w^{(1)}/N_1, \ldots, L_w^{(c)}/N_c)$ is a block diagonal matrix.

Thus, in row direction, the image between-class Laplacian scatter matrix LS_b^{row} is:

$$LS_b^{\text{row}} = LS_t^{\text{row}} - LS_w^{\text{row}} \qquad (13)$$

Similarly, each image matrix can also be regarded as an n-set of column vectors. Then the corresponding scatter matrices in column can be defined as:

$$S_t^{\text{col}} = \frac{1}{2N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(x_i-x_j)(x_i-x_j)^T \qquad (14)$$

$$S_w^{\text{col}} = \frac{1}{N}\sum_{i=1}^{c}\frac{1}{2N_i}\sum_{j,k=1}^{N_i}(x_j^{(i)}-x_k^{(i)})(x_j^{(i)}-x_k^{(i)})^T \qquad (15)$$

$$LS_t^{\text{col}} = \frac{1}{N^2}X(L\otimes I_n)X^T \qquad (16)$$

$$LS_w^{\text{col}} = \frac{1}{N}X(L_w\otimes I_n)X^T \qquad (17)$$

$$LS_b^{\text{col}} = LS_t^{\text{col}} - LS_w^{\text{col}} \qquad (18)$$

3. Solution of TSA and LBMMC

Both TSA[23] and LBMMC[22] exploit the Eq.(1) to implement feature extraction. The difference between them is that LBMMC adopts Maximum Margin Criterion to solve the projection matrices U and V while TSA employs the Fisher Discriminant Criterion to iteratively compute the projection matrices.

After performing feature extraction, the real Laplacian scatter matrices in feature space are:

$$\begin{aligned}&LS_t^{\text{row}} =\\ &\frac{1}{2N^2}\sum_{i,j=1}^{N} w_{ij}V^T(x_i-x_j)^TUU^T(x_i-x_j)V \qquad (19)\\ &= \frac{1}{N^2}V^TX'^T(e\otimes U)(L\otimes I_{m'})(e\otimes U)^TX'V \qquad (20)\end{aligned}$$

$$\begin{aligned}&LS_w^{\text{row}} =\\ &\frac{1}{N}\sum_{i=1}^{c}\frac{1}{2N_i}\sum_{j,k=1}^{N_i} w_{jk}^{(i)}V^T(x_j^{(i)}-x_k^{(i)})^TUU^T(x_j^{(i)}-x_k^{(i)})V\\ &= \frac{1}{N}V^TX'^T(e\otimes U)(L_w\otimes I_{m'})(e\otimes U)^TX'V \qquad (21)\end{aligned}$$

$$\begin{aligned}LS_t^{\text{col}} &= \frac{1}{2N^2}\sum_{i,j=1}^{N} w_{ij}U^T(x_i-x_j)VV^T(x_i-x_j)^TU\\ &= \frac{1}{N^2}U^TX(e\otimes V)(L\otimes I_{n'})(e\otimes V)^TX^TU \qquad (22)\end{aligned}$$

$$\begin{aligned}&LS_w^{\text{col}} =\\ &\frac{1}{N}\sum_{i=1}^{c}\frac{1}{2N_i}\sum_{j,k=1}^{N_i} w_{jk}^{(i)}U^T(x_j^{(i)}-x_k^{(i)})VV^T(x_j^{(i)}-x_k^{(i)})^TU\\ &= \frac{1}{N}U^TX(e\otimes V)(L_w\otimes I_{n'})(e\otimes V)^TX^TU \qquad (23)\end{aligned}$$

where e is a column vector of all 1's with suitable dimensionality.

From the above, one can see that each scatter matrix in feature space depends on both projection matrix U and V. However, for simplicity of solution, LBMMC computes Laplacian scatter matrices in a simplified way: it ignores the influence of column projection U when computing row scatter matrices and ignores the influence of row projection V when computing column scatter matrices.

$$\widehat{LS}_t^{\text{row}} = \frac{1}{2N^2}\sum_{i,j=1}^{N} w_{ij}V^T(x_i-x_j)^T(x_i-x_j)V \qquad (24)$$

$$\widehat{LS}_w^{\text{row}} = \frac{1}{N}\sum_{i=1}^{c}\frac{1}{2N_i}\sum_{j,k=1}^{N_i} w_{jk}^{(i)}V^T(x_j^{(i)}-x_k^{(i)})^T(x_j^{(i)}-x_k^{(i)})V \qquad (25)$$

$$\widehat{LS}_t^{\text{col}} = \frac{1}{2N^2}\sum_{i,j=1}^{N} w_{ij}U^T(x_i-x_j)(x_i-x_j)^TU \qquad (26)$$

$$\begin{aligned}&\widehat{LS}_w^{\text{col}} =\\ &\frac{1}{N}\sum_{i=1}^{c}\frac{1}{2N_i}\sum_{j,k=1}^{N_i} w_{jk}^{(i)}U^T(x_j^{(i)}-x_k^{(i)})(x_j^{(i)}-x_k^{(i)})^TU \qquad (27)\end{aligned}$$

By this simplification, LBMMC can easily solve projection matrices U and V in a separate way via maximizing the projection margin in two directions respectively:

$$\begin{aligned} \max_{V}: \quad & \mathrm{tr}(\widehat{LS_b}^{\mathrm{row}} - \widehat{LS_w}^{\mathrm{row}}) = \mathrm{tr}(\widehat{LS_t}^{\mathrm{row}} - 2\widehat{LS_w}^{\mathrm{row}}) \\ \max_{U}: \quad & \mathrm{tr}(\widehat{LS_b}^{\mathrm{col}} - \widehat{LS_w}^{\mathrm{col}}) = \mathrm{tr}(\widehat{LS_t}^{\mathrm{col}} - 2\widehat{LS_w}^{\mathrm{col}}) \end{aligned}$$

Although the projection matrices U and V can be easily computed in LBMMC, they are not optimal in terms of face recognition performance because they do not maximize the real projection margin in feature space.

Since TSA adopts the Fisher Discriminant Criterion instead of Maximum Margin Criterion, it computes the optimal U and V via solving the following minimization problem:

$$\min_{U,V} \quad \frac{\mathrm{tr}(LS_w^{\mathrm{row}})}{\mathrm{tr}(LS_t^{\mathrm{row}})} = \frac{\mathrm{tr}(LS_w^{\mathrm{col}})}{\mathrm{tr}(LS_t^{\mathrm{col}})} \qquad (28)$$

The equivalence holds because $\mathrm{tr}(LS_w^{\mathrm{row}}) = \mathrm{tr}(LS_w^{\mathrm{col}})$ and $\mathrm{tr}(LS_t^{\mathrm{row}}) = \mathrm{tr}(LS_t^{\mathrm{col}})$. The optimal U and V are computed in a iteratively procedure, alternating between the computation of U for a given V and the computation V for a given U. As mentioned in section 1, the problem of TSA is that there is no theoretical guarantee that the iterative procedure will be convergent.

4. Our solution

The proposed method inherits the maximum margin criterion from LBMMC and iterative optimization idea from TSA and combines them together. This results in a convergent solution, details of which will be given below.

4.1. the proposed optimization objective

According to the definition of Laplacian scatter in previous section, the Laplacian projection margin in each direction can be defined as:

$$J_{\mathrm{row}}(U,V) = \mathrm{tr}(LS_t^{\mathrm{row}} - 2LS_w^{\mathrm{row}}) = \mathrm{tr}(V^T J_U V) \qquad (29)$$

$$J_{\mathrm{col}}(U,V) = \mathrm{tr}(LS_t^{\mathrm{col}} - 2LS_w^{\mathrm{col}}) = \mathrm{tr}(U^T J_V U) \qquad (30)$$

where:

$$\begin{aligned} J_U = & \frac{1}{N^2} X'^T (e \otimes U)(L \otimes I_{m'})(e \otimes U)^T X' \\ & - \frac{2}{N} X'^T (e \otimes U)(L_w \otimes I_{m'})(e \otimes U)^T X' \end{aligned} \qquad (31)$$

only depends on U, and

$$\begin{aligned} J_V = & \frac{1}{N^2} X (e \otimes V)(L \otimes I_{n'})(e \otimes V)^T X^T \\ & - \frac{2}{N} X (e \otimes V)(L_w \otimes I_{n'})(e \otimes V)^T X^T \end{aligned} \qquad (32)$$

only depends on V.

Large $J_{\mathrm{row}}(U,V)$ and $J_{\mathrm{col}}(U,V)$ means that in feature space samples are far from each other if they are from different classes, but are close to each other if they are from the same class. Moreover, since the Laplacian similarity which can capture the underlying manifold structure of image set is introduced to the scatter matrix, the local geometric structure of image set can be well preserved in the feature space. Thus our optimization objective will be:

$$\begin{aligned} \max: \quad & J(U,V) = J_{\mathrm{row}}(U,V) + J_{\mathrm{col}}(U,V) \\ s.t. \quad & U^T U = I_{m'},\ V^T V = I_{n'} \end{aligned} \qquad (33)$$

4.2. The solution and its convergency

It is difficult to obtain the optimal U and V simultaneously since they depend on each other. From the Eqs. (20)(21)(22)(23), it is easy to check that

$$J_{\mathrm{row}}(U,V) = J_{\mathrm{col}}(U,V) \qquad (34)$$

This identity plays an important role in solving the optimization problem (33). By this identity, the objective function in (33) can be expressed as:

$$J(U,V) = 2J_{\mathrm{row}}(U,V) = 2J_{\mathrm{col}}(U,V) \qquad (35)$$

Eq.(35) means that we can maximize $J(U,V)$ by maximizing $J_{\mathrm{row}}(U,V)$ or $J_{\mathrm{col}}(U,V)$ separately. This motivates us to iteratively solve the optimization problem (33). Fixing U, we refine V by maximizing $J_{\mathrm{row}}(U,V)$, and for fixed V, we obtain U by maximizing $J_{\text{col}}(U,V)$. The details of the iterative procedure to solve U and V are listed in Fig.1.

```
Algorithm 1. Procedure to iteratively solve $U$ and $V$
1: Initialization. set initial $U_0$ as arbitrary column orthogonal matrix.
2: Iterative optimization.
    For $k = 1, 2, \dots, k_{\max}$, do
        1. Compute $J_{U_{k-1}}$ according to (31)
        2. Solve the eigenvalue decomposition problem:
               $J_{U_{k-1}} v_i = \lambda_i v_i, i = 1, 2, \dots, n'$    (36)
           where $v_i$ is the eigenvector corresponding to the $i$th largest eigenvalue $\lambda_i$
        3. Set $\overline{V} = [v_1, v_2, \dots, v_{n'}]$.
        4. Reshape $\overline{V}$ for orthogonal transformation invariance, let $S = \overline{V}\overline{V}^T X'^T X' \overline{V}\overline{V}^T$, solve eigenvalue decomposition:
               $S v_i = \gamma_i v_i$    (37)
           and let $V_k = [v_1, v_2, \dots, v_{n'}]$, where $v_i$ is eigenvector corresponding to the $i$th largest eigenvalue.
        5. Compute $J_{V_k}$ according to (32)
        6. Solve the eigenvalue decomposition problem:
               $J_{V_k} u_j = \lambda'_j u_j, j = 1, 2, \dots, m'$    (38)
           where $u_j$ is the eigenvector corresponding to the $j$th largest eigenvalue $\lambda'_j$.
        7. Set $\overline{U} = [u_1, u_2, \dots, u_{m'}]$.
        8. Reshape $\overline{U}$ for orthogonal transformation invariance, let $T = \overline{U}\overline{U}^T X X^T \overline{U}\overline{U}^T$, solve eigenvalue decomposition:
               $T u_i = \gamma_i u_i$    (39)
           and let $U_k = [u_1, \dots, u_{m'}]$, where $u_i$ is eigenvector corresponding to the $i$th largest eigenvalue.
    if $\|U_k - U_{k-1}\| < \sqrt{m'}\varepsilon$ and $\|V_k - V_{k-1}\| < \sqrt{n'}\varepsilon$ (we set $\varepsilon = 0.01$ in our experiments), then return.
3: Output the projection matrices $U = U_k$, $V = V_k$.
```

Figure 1: Algorithm 1: iterative solution with manually specified dimensionality

A notable property of the proposed iterative procedure is that it can monotonously increase the objective function value as proved in Theorem 1, which directly guarantees its superiority over LB-MMC and TSA.

Theorem 1. *The objective function satisfies the following inequality:*

$$J(U_k, V_k) \leqslant J(U_k, V_{k+1}) \leqslant J(U_{k+1}, V_{k+1}) \tag{40}$$

Proof. According to algorithm 1, U_k and V_k satisfy:

$$\operatorname{tr}(V_k^T J_{U_{k-1}} V_k) = \max_{V^T V = I_{n'}} \operatorname{tr}(V^T J_{U_{k-1}} V) \tag{41}$$

$$\operatorname{tr}(U_k^T J_{V_k} U_k) = \max_{U^T U = I_{m'}} \operatorname{tr}(U^T J_{V_k} U) \tag{42}$$

Then by using (34), we have:

$$J_{\text{col}}(U_k, V_k) = J_{\text{row}}(U_k, V_k) \tag{43}$$

$$\leqslant J_{\text{row}}(U_k, V_{k+1}) = J_{\text{col}}(U_k, V_{k+1}) \tag{44}$$

$$\leqslant J_{\text{col}}(U_{k+1}, V_{k+1}) \tag{45}$$

Thus according to (35), the objective function satisfies:

$$J(U_k, V_k) \leqslant J(U_k, V_{k+1}) \leqslant J(U_{k+1}, V_{k+1}) \tag{46}$$

□

Theorem 1 means that the objective function monotonously increases with regard to the iteration number k. Another fact is that the objective function has upper bound under the constraints in (33). Therefore objective function will converge in limited iterations. Then we can further obtain the following Theorem which gives the convergency of the solution of algorithm 1:

Theorem 2. *The projection matrices sequence* $\{(U_k, V_k)\}$ *obtained by algorithm 1 will converge.*
Proof. According to Theorem 1, we can assume that there are K iterations when the objective function converges, then we have:

$$J(U_K, V_K) = J(U_K, V_{K+1}) = J(U_{K+1}, V_{K+1}) \tag{47}$$

Substituting (35) into (47) and by using (29) and (30), we can obtain that:

$$\operatorname{tr}(V_K^T J_{U_K} V_K) = \operatorname{tr}(V_{K+1}^T J_{U_K} V_{K+1}) \tag{48}$$

Then $\exists$ orthogonal matrices Q such that:

$$V_{K+1} = V_K Q. \tag{49}$$

Thus, V_{K+1} and V_K has the same column space. And in step 4 of algorithm 1, we have reshaped V_K and V_{K+1} to be the leading eigenvectors of samples projected to this column space, therefore we have:

$$V_{K+1} = V_K. \tag{50}$$

Similarly, we can obtain $U_{K+1} = U_K$. This means that when objective function converges, the projection matrices sequence also converges. □

When iteration in algorithm 1 stops, we have the following conclusion:

$$J(U,V) = 2\sum_{i=1}^{n'} \lambda_i = 2\sum_{j=1}^{m'} \lambda'_j \tag{51}$$

It is a straightforward inference of Eqs.(35),(36) and (38).

4.3. *Selecting suitable feature dimensionality*

Selecting a suitable feature dimensionality is a key issue for dimensionality reduction based feature extraction. In the previous section, we manually specify the feature dimensionality for the proposed approach beforehand. In fact, our method is capable of automatic selection of suitable dimensionality to obtain comparable performance.

As MMC, the Laplacian projection margin captures the average gap between different classes. According to (36) and (38), after the kth iteration, we have:

$$J(U_{k-1}, V_k) = 2\sum_{i=1}^{n'} \lambda_i; \tag{52}$$

$$J(U_k, V_k) = 2\sum_{i=1}^{m'} \lambda'_i \tag{53}$$

Therefore, eigenvectors with positive eigenvalues will enlarge this gap, then further make samples from different classes well separated(on average) in feature space. In contrast, samples from different classes will overlap in the directions of eigenvectors with negative eigenvalues. So we should choose the eigenvectors with positive eigenvalues to constitute the projection matrices U and V. Another consideration is that feature extraction is not only to reduce the dimensionality but also to preserve as much information as possible such as local geometric structure, so we also pick up some eigenvectors with eigenvalues slightly less than 0.

Then we can describe our method with automatic selection feature dimensionality in Fig.2:

```
Algorithm 2
1: Set initial $U = I_m$ and $V = I_n$;
2: Iteratively solve (36) and (38) until the objective function converges, in each iteration, choose eigenvectors with eigenvalues more than $\varepsilon$(we set $\varepsilon = -0.01$ in our experiments) to constitute the projection matrices $U_k$ and $V_k$.
3: Output the projection matrices $U = U_k$ and $V = V_k$.
```

Figure 2: Algorithm 2 with automatical selection suitable feature dimensionality

5. Experimental results

In this section, we first experimentally investigate the convergency of the proposed iterative solution procedure, then we evaluate the performance of the proposed method on two benchmark face databases CMU PIE face database(pose, illumination and expression)[26] and Yale face database[27], which are widely used in face recognition community. Since Ref.[23] and Ref.[22] have compared TSA and LBMMC with Eigenfaces, Fisherfaces, Laplacianfaces and other state-of-the-art approaches, respectively, and it is reported that TSA and LBMMC outperform those methods in most cases. Here we only compare our method with MMC, TSA and LBMMC.

For face recognition we first use these methods to learn the projection matrices(vectors) from the training face images, then for any new image to be identified we compute its new feature by using the learned projection matrices(vectors), finally the nearest classifier with cosine distance is employed for classification in the new feature space. And in TSA, LBMMC and our method we set σ as the half of the mean distance between any pair of samples, and the number of iterations in TSA is taken to be 10.

5.1. Data sets

We downloaded the two face data sets from He's homepage[‡] All the images have been aligned according to the positions of eyes, and each images have been cropped to a size of 32×32, with 256 gray levels per pixel. For MMC method, the image is represented as an 1024-dimensional vector, while in TSA, LBMMC and our method, the image is represented as a (32×32)-dimensional matrix.

CMU PIE face database: The whole PIE face database from CMU contains 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. The subset used in our experiments contains images from five near frontal poses(C05,C07,C09,C27,C29) under all different illuminations and expressions, so there are 170 images for each individual. We randomly select $l(l = 5,10,20)$ images of each person for training and the rest for testing. For each l, we do 20 random split experiments.

Yale face database: The Yale face database contains 165 images of 15 individuals(each person providing 11 different images) under various facial expressions and configurations. For this database, we randomly select $l(= 3,6,8)$ for training and the rest for testing. For each l, we also do 20 random split experiments.

Sample images of two people from these two face database are shown in Fig.3.

5.2. Convergency of iterative procedure

First we examine the convergency of the objective function and the projection matrices of the proposed method on these two data sets. For both projection matrices and objective function value, we compute the difference norm of two successive iterations. The details of step differences versus iteration number are shown in Fig.4.

Monotony and convergence of Objective function Value: From Fig.4(a)(d) one can see the step difference of objective value in our method is always positive until it converge to 0, this means that the objective function value monotonously increases as the iteration number and it converges after about 5–10 iterations. This confirms the claim in Theorem 1. The high convergence speed of the iterative procedure guarantee that we can obtain the optimized U and V efficiently. While step difference of the objective value in TSA oscillates over the iteration and it does not converge.

Convergence of projection matrices: Fig.4 (b,c,e,f) demonstrates that the projection matrices U and V also converge when the objective function converges in our new solution procedure. While for TSA, solution matrices U and V oscillates over the iteration and they do not converge.

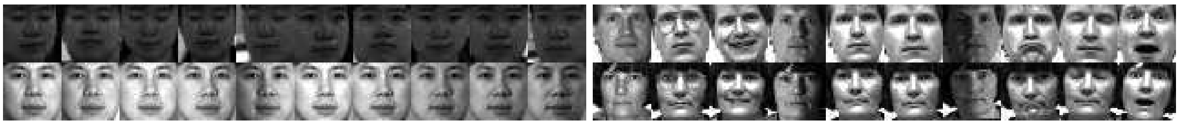
5.3. Performance for face recognition

To evaluate recognition performance of different approaches, for each l we do 20 random split experiments and in each run we obtain the recognition rate under all possible feature dimensionality. Fig.5 plots average recognition rate versus feature dimensionality for MMC, TSA, LBMMC and our method over the 20 random split experiments. For Baseline method, the recognition task is conducted in the original 1024-dimensional image space without any preprocess. For TSA, LBMMC and our method, we only show the recognition rate under feature dimensionality $d \times d$ i.e., 1,4,9, etc. For MMC, the recognition rate under feature dimensionality $10i(i = 1 : 40)$ is shown. As can be seen, recognition rate of all methods varies as feature dimensionality. However, in most cases recognition rate of algorithm 1 is higher than that of others, so we can conclude that our method outperforms others.

Then we show the top average recognition rate of different methods and corresponding feature dimensionality(in parentheses) in Table 1 for different l. Here one should note that for Algorithm 2, which can automatically select suitable feature dimensionality, in each run we can only obtain its recognition rate under feature dimensionality selected by the algorithm 2 itself which may be different in different runs.

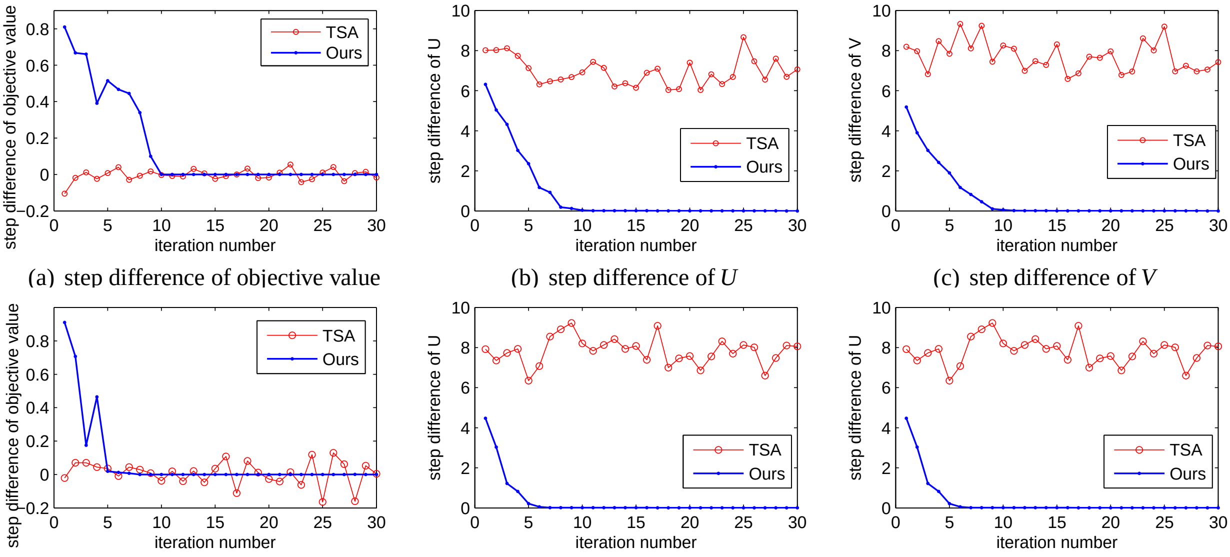
What's more, in each random split experiment, there exists a top recognition rate for each method

[‡] http://www.zjucadcg.cn/dengcai/Data/FaceData.html



(a) images of two persons in PIE (b) images of two persons in yale

Figure 3: Face images from two databases



(a) step difference of objective value (b) step difference of *U* (c) step difference of *V*

(d) step difference of objective value (e) step difference of *U* (f) step difference of *V*

Figure 4: Convergency of the iterative procedure on the two face databases. (a,b,c) results on PIE database; (d,e,f) results on Yale database

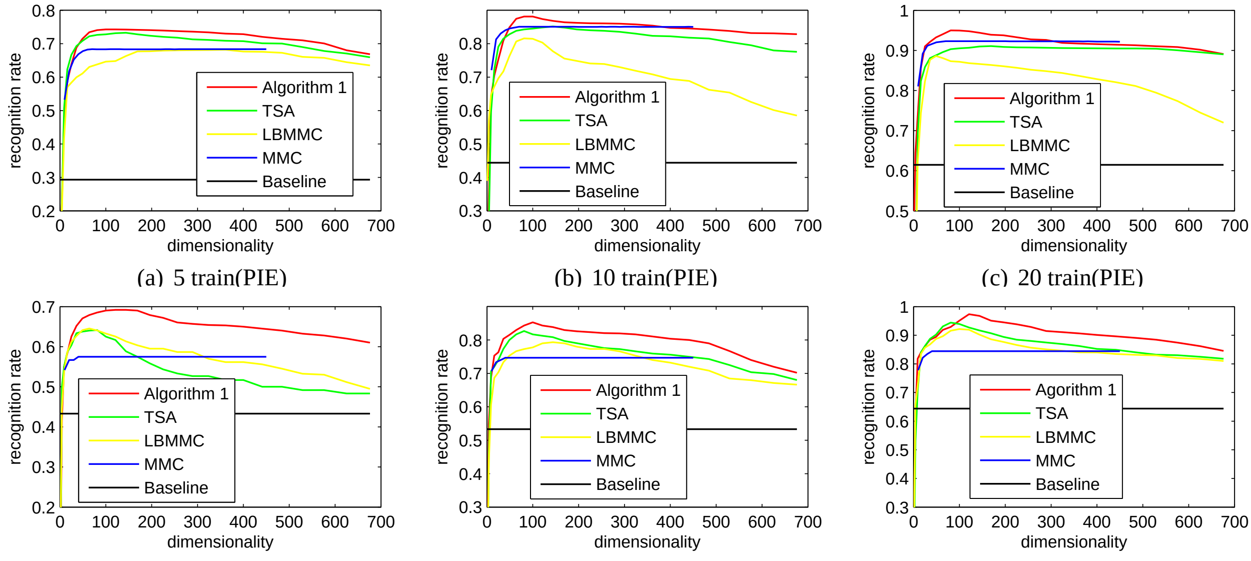
by considering all possible feature dimensionality. Table 2 shows the mean value and standard deravation(in parentheses) of these top recognition rates of all methods over 20 random split experiments for different l. In both table 1 and 2, the best results are shown in boldface. From these results, we can draw the following conclusions:

1. Our method Algorithm 1 mostly outperforms all the other methods concerned in this work.

2. To obtain the top recognition rate, other methods need to consider all the possible feature dimensionality. However, with the automatically selected feature dimensionality, our method algorithm 2 can obtain competitive recognition rate to TSA, and higher recognition rate to LBMMC and MMC.

6. Conclusions

Feature extraction is a key issue for face recognition. Conventional vector-based feature extraction need to convert the image into higher dimensional vector. Existing feature extraction approaches based on matrix(tensor) representation of image and matrix bidirectional projection encounters the problem that there is no convergent solution or that the solution is non-optimal. To overcome these problems, in this paper, we propose a novel matrix bidirectional projection based feature extraction method. Different from previous tensor subspace learning to maximize Fisher Criterion, which is trace ratio, the proposed method employs maximum margin criterion, which is trace difference. To obtain the optimal projection matrices, an iterative optimization procedure



(a) 5 train(PIE)
(b) 10 train(PIE)
(c) 20 train(PIE)
(d) 3 train(Yale)
(e) 6 train(Yale)
(f) 8 train(Yale)

Figure 5: Average recognition rate vs. feature dimensionality($d \times d$ for TSA, LBMMC and Algorithm 1, $10 * i$ for MMC is shown). (a,b,c) results on PIE database; (d,e,f) results on Yale database

Table 1: Top average recognition rate(dimensionality) of different methods(%)

method	PIE			Yale		
	5 train	10 train	20 train	3 train	6 train	8 train
Baseline	29.3	44.3	61.4	43.3	54.5	64.4
MMC	68.2(70)	85.2(70)	92.0(70)	57.5(40)	74.7(40)	84.4(40)
TSA	73.5(12^2)	85.3(13^2)	90.8(13^2)	64.1(9^2)	82.7(9^2)	94.0(9^2)
LBMMC	68.1(15^2)	81.7(9^2)	88.6(8^2)	63.7(8^2)	79.3(12^2)	92.1(10^2)
Algorithm 1	**74.2**(10^2)	**88.2**(9^2)	**94.8**(9^2)	**69.2**(11^2)	**85.1**(10^2)	**97.3**(11^2)
Algorithm 2	72.6	87.4	91.7	65.1	81.6	93.1

is proposed. There are two key properties of the proposed iterative optimization procedure: 1) it can monotonously increase the objective value; and 2) both the solution matrices and objective function are convergent. Finally extensive and systematic face recognition experiments on CMU PIE and Yale face databases demonstrate the efficiency and effectiveness of the proposed method.

References

1. M. Turk and A. Pentland. “Face recognition using eigenfaces.” “CVPR,” 586–591 (1991).
2. P. Belhumeur, J. Hespanha, D. Kriegman *et al.* “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection.” *IEEE Transactions on pattern analysis and machine intelligence*, **19(7)**, 711–720 (1997).
3. S. J. Raudys and A. K. Jain. “Small sample size effects in statistical attern recognition: recommendations for practitioners.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13(3)**, 252–264 (1991).
4. J. Friedman. “Regularized discriminant analysis.” *Journal of the American Statistical Association*, **84(405)**, 165–175 (1989).
5. L. Chen, H. Liao, M. Ko, J. Lin and G. Yu. “A new lda-based face recognition system which can solve the small sample size problem.” *Pattern Recognition*, **33**, 1713–1726 (2000).
6. T. Hastie and R. Tibshirani. “Penalized discriminant analysis.” *The Annals of Statistics*, **23**, 73–102 (1995).
7. M. C. Padma. “Language Identification Of Kan-

Table 2: The mean value(%) and standard derivations(%) of top recognition rate over 10 random split experiments

method	PIE			Yale		
	5 train	10 train	20 train	3 train	6 train	8 train
Baseline	29.3(1.0)	44.3(0.68)	61.4(0.6)	43.3(5.3)	54.5(4.7)	64.4(2.9)
MMC	68.8(1.3)	85.4(1.0)	92.1(0.4)	60.8(4.7)	76.1(3.9)	85.0(2.3)
TSA	73.6(1.2)	85.8(0.8)	91.1(0.3)	64.1(4.1)	83.1(3.3)	94.1(3.3)
LBMMC	69.1(1.1)	81.7(1.0)	88.9(0.7)	63.9(4.5)	80.7(3.5)	92.2(3.4)
Algorithm 1	**74.4**(1.0)	**88.3**(0.64)	**95.0**(0.3)	**69.4**(3.9)	**85.4**(3.0)	**97.4**(1.9)
Algorithm 2	72.6(0.9)	87.4(0.7)	91.7(0.6)	65.1(4.7)	81.6(4.3)	93.1(2.7)

nada, Hindi And English Text Words Through Visual Discriminating Features.” *International Journal of Computational Intelligence Systems*, **1(2)**, 116–126 (2008).

8. H. Li, T. Jiang and K. Zhang. “Efficient and robust feature extraction by maximum margin criterion.” *IEEE Transactions on Neural Networks*, **17(1)**, 157–165 (2006).
9. M. Belkin and P. Niyogi. “Laplacian eigenmaps for dimensionality reduction and data representation.” *Neural computation*, **15(6)**, 1373–1396 (2003).
10. X. He, S. Yan, Y. Hu, P. Niyogi and H. Zhang. “Face recognition using laplacianfaces.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27(3)**, 328–340 (2005).
11. S. Roweis and L. Saul. “Nonlinear dimensionality reduction by locally linear embedding.” *Science*, **290**, 2323–2326 (2000).
12. J. Tenenbaum, V. Silva and J. Langford. “A global geometric framework for nonlinear dimensionality reduction.” *Science*, **290**, 2319–2323 (2000).
13. N. K. Narayanan and V. Kabeer. “Face Recognition Using Nonlinear Feature Parameter and Artificial Neural Network.” *International Journal of Computational Intelligence Systems*, **3(5)**, 566–574 (2000).
14. Z. Zhang and H. Zha. “Principal manifolds and nonlinear dimension reduction via local tangent space alignment.” *SIAM Journal of Scientific Computing*, **26(1)**, 313–338 (2004).
15. X. He, D. Cai, S. Yan and H. Zhang. “Neighborhood preserving embedding.” “ICCV,” 1208–1213 (2005).
16. T. Zhang, J. Yang, D. Zhao and X. Ge. “Linear local tangent space alignment and application to face recognition.” *Neurocomputing*, **70**, 1547–1553 (2007).
17. X. He and P. Niyogi. “Locality preserving projections.” “NIPS 16,” 153–160 (2003).
18. J. Yang, D. Zhang, A. Frangi and J. Yang. “Two-dimensional pca: a new approach to appearance-based face representation and recognition.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26(1)**, 131–137 (2004).
19. M. Li and B. Yuan. “2d-lda: a statistical linear discriminant analysis for image matrix.” *Pattern Recognition Letters*, **26(5)**, 527–532 (2005).
20. B. Niu, Q. Yang, S. C. K. Shiu and S. K. Pal. “Two-dimensional laplacianfaces method for face recognition.” *Pattern Recognition*, **41(10)**, 3237–3243 (2008).
21. W. Zuo, D. Zhang and K. Wang. “Bidirectional pca with assembled matrix distance metric for image recognition.” *IEEE Transactions on systems, man, and cybernetics. Part B, Cybernetics*, **36(4)**, 863–872 (2006).
22. W. Yang, J. Wang, M. Ren, J. Yang, L. Zhang *et al.* “Feature extraction based on laplacian bidirectional maximum margin criterion.” *Pattern Recognition*, **42(11)**, 2327–2334 (2009).
23. X. He, D. Cai and P. Niyogi. “Tensor subspace analysis.” “NIPS 19,” vol. 18, 499–506 (2006).
24. D. Cai, X. He, J. Han and H.-J. Zhang. “Orthogonal laplacianfaces for face recognition.” *IEEE Trans. Image Processing*, **15(11)**, 3608–3614 (2006).
25. S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang *et al.* “Graph embedding and extensions: A general framework for dimensionality reduction.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29(1)**, 40–51 (2007).
26. T. Sim, S. Baker and M. Bsat. “The cmu pose, illumination, and expression (pie) database.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25(12)**, 1615–1618 (2003).
27. A. Georghiades, P. Belhumeur and D. Kriegman. “From few to many: Illumination cone models for face recognition under variable lighting and pose.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23(6)**, 643–660 (2001).