Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009

# The Matrix Form for Weighted Linear Discriminant Analysis and Fractional Linear Discriminant Analysis

Tianwei Xu[1], Chong Lu[2,3], Wanquan Liu[4]

[1] *Yunan Normal University, Kunming 650092, P. R. China.*

[2] *Dalian University of Technology, Dalian, 116024, China*

[4] *Curtin University of Technology, Perth WA, 6102, Australia*

[3] *YiLi Normal College, Yining, 835000, China*

Email: xutianwei@ynnu.edu.cn;cluxjyn@gmail.com; w.liu@curtin.edu.au

## Abstract

**In this paper we will extend the recently proposed weighted linear discriminant analysis (W_LDA) and fraction-step linear discriminant analysis (F_LDA) from one dimension vector form to the case of two dimension matrix form, which are called weighted two dimensional linear discriminant analysis (W_2DLDA) and fraction-step two dimension linear discriminant analysis(F_2DLDA), respectively. The motivation of this work is based on the recent research results on two dimensional Principal Component Analysis (2DPCA) and 2DLDA showing that the two dimensional algorithms can save computational costs significantly and thus improve the classifiers' performances. First, we derived these numerical algorithms in matrix form and then we implement these two new algorithms on ORL and YALE face databases. The experimentation results show that W_2DLDA produces the best performance among F_2DLDA, F_LDA and W_LDA.**

**Keywords** Linear Discriminant Analysis, Two dimensional Linear Discriminant Analysis, Weighted Linear Discriminant Analysis, Fraction_step Linear Discriminant Analysis, Face recognition.

## 1 Introduction

In last decade, many variations for linear discriminant analysis (LDA) are proposed and applied to face recognition [1,2,3]. Different versions of LDA have proven improving classification performance.

Recently, an improved face recognition approach to boosting the worst-case performance is proposed by Chen et al.[1] and a fraction-step dimensionality reduction method is proposed by Rohit et al.[3]. Both of these two approaches are proved to have achieved better performance than the original LDA[7-9]. However the performances for these two approaches have not been compared in [1]. It is known that face image is a usually high-dimension data, therefore, different dimension reduction techniques are usually combined with linear discrimination methods in order to achieve better performance [5][10][11]. For example, The Fisher-

978-1-4244-3703-0/09/$25.00 ©2009 IEEE

face method [5] combines PCA and thus the Fisher criterion is used to extract the information that can discriminates different classes [2]. Alternatively, two dimensional LDA method is proposed by Li et al.[4], and a simplified generalized low rank approximation of matrices (SGLRAM) of dimension reduction technique is proposed by Lu et al.[6]. All these research results show that the two dimensional matrix form algorithms can reduce the computational costs significantly while achieving higher performance. In this paper we aim to extend the current vector form of W_LDA and F_LDA algorithms to the two dimensional case for face recognition respectively, and compare their performance. In order to implement these experiments efficiently, we use the SGLRAM to reduce the dimension of face matrix first so that we can use the proposed approaches in experiment conveniently. Finally we do experiments on the ORL and YALE face databases and the results shows that W_2DLDA outperforms F_2DLDA both in accuracy and computational time.

The rest of this paper is organized as follows. In section 2, we extend the W_LDA to two dimensional case, and in section 3, F_2DLDA is derived similarly. We do experiments with different approaches and compare the results in section 4. Conclusions are given in section 5.

## 2 The Weighted Two Dimensional LDA

In this section, we will extend W_LDA to the two dimensional case.

Chen et al.[1] propose the W_LDA algorithm for face recognition and its main idea is to maximize the smallest between-class distances via choosing the projection matrix instead of solving the Fisher criteria in LDA. Actually they formulate such problem as a weighted Fisher criteria for tractable solutions. In this section, we will extend W_LDA [1] in vector form to the two dimensional matrix form and expect the proposed algorithm will improve performance while reducing computational costs. This belief is from results in [4] that 2DLDA outperforms LDA in face recognition. For such an aim, we assume that dataset X has N training samples $X_{ij} \in R^{m\times n}$ with C classes and $\mu_i$ is the mean of the i-th ($1\leq i \leq C$) class. The between-class covariance matrix is defined as following in matrix form.

$$S_b = \sum_{i=1}^{C} n_i(\mu_i - \mu)(\mu_i - \mu)^\top \tag{1}$$

where $\mu$ is the mean of whole dataset X and $n_i$ is the number of samples in i-th class with $N = \sum_{i=1}^{C} n_i$. The within-class covariance matrix is defined as

$$S_w = \sum_{i=1}^{C}\sum_{j=1}^{n_i}(X_{ij} - \mu_i)(X_{ij} - \mu_i)^\top \tag{2}$$

where $X_{ij}$ denotes j-th sample in i-th class of dataset X. According to the Fisher criterion, 2DLDA aims to obtain an optimal projection matrix $V_{opt}$ via solving the following optimization problem:

$$V_{opt} = \arg\max_{V} tr\{(V^\top S_w V)^{-1} V^\top S_b V\} \tag{3}$$

where $V \in R^{m\times m}$ . If the numbers of training samples are equal in each class, it can be derived that

$$tr(V^\top S_b V) = \frac{1}{C}\sum_{i<j}\left\|V^\top \mu_i - V^\top \mu_j\right\|^2 \tag{4}$$

Let $S_{ij} = (\mu_i - \mu_j)(\mu_i - \mu_j)^\top$, then

$$tr\{(V^\top S_w V)^{-1} V^\top S_b V\} = \frac{1}{C}\sum_{i<j} tr\{(V^\top S_w V)^{-1} V^\top S_{ij} V\} \tag{5}$$

Then the objective function (3) can be written as

$$V_{opt} = \arg\max_{V} \frac{1}{C}\sum_{i<j} tr\{(V^\top S_w V)^{-1} V^\top S_{ij} V\} \tag{6}$$

In the above formulated 2DLDA, it should be noted that a higher recognition rate in testing for one class and a lower recognition rate for another class can still produce acceptable/satisfactory average classification rate from (6). This is a typical problem for both LDA and 2DlDA. Chen et al [1] aim to tackle such problem via improving the worst-case classification performance for LDA, which is an idea from control community. As it is known that 2DLDA is more powerful than LDA [4], we aim to improve the worst-case recognition rate for 2DLDA and thus propose the weighted 2DLDA. Even though this idea is to compromise the best-class performance via improving the worst-class performance, we expect the average recognition performance can be enhanced as shown in [1]. To describe this problem explicitly, the original Fisher criterion (6) should be revised as below.

$$V_{opt} = \arg\max_{V} \left( \min_{ij} tr\{(V^\top S_w V)^{-1} V^\top S_{ij} V\} \right) \quad (7)$$

This is a maxmin problem mathematically and it is hard to solve it analytically. We do not intend to solve such a hard mathematical problem here and present an alternative approach via introducing the weighting coefficients $d_{ij}$ on the between-class distances and reformulate it into a tractable optimization problem given below.

$$\tilde{V}_{opt} = \arg\max_{V} \left( tr\{(V^\top S_w V)^{-1} \sum_{i<j} d_{ij} S_{ij}\} \right) \quad (8)$$

Then, the solution of (8) can be obtained by conducting eigenvalue decomposition on $S_w{}^{-1} \sum_{i<j} d_{ij} S_{ij}$ and choosing the eigenvectors associated with large eigenvalues. However, this eigenvalue problem depends on the parameters $d_{ij}$, which we give an intuitive approach to solve it. Therefore, we propose the following **Algorithm W_2DLDA**.

---

**Input:** Training images $X_{ij}$ represented as $m \times n$ matrix.

**Output:** Projection matrix $V_{opt}$

**Algorithm W_2DLDA**

---

**Step 1:** Using SGLRAM [6] to reduce each the face dimension of X from $m \times n$ to $m' \times n'$.

**Step 2:** Calculate $V$ from the Fisher criterion (3).

**Step 3:** Calculate and record the smallest between-class distance $D_S$

**Step 4:** Adjust all weighting coefficients $d_{ij}$ by increasing $d^S$, which corresponds to the smallest between-class distance, to $d^S + \Delta d^S$, and reducing other coefficients correspondingly to $d_{ij} + \Delta d_{ij}$ while keeping the sum of $d_{ij}$ as a constant.

**Step 5:** Calculate $V_{opt}$ with updated $d_{ij}$

**Step 6:** Compute $V_{opt}$ and go to step 3 until it converges.

**Step 7:** Output the final $V_{opt}$.

---

In the above algorithm, the update of weighting coefficients plays a key role and we adjust them as follows: for the smallest between-class distance, we choose

$$\Delta d^S = \delta \times \exp(-\|D_S\|/\sigma) \quad (9)$$

and for all other between-class distances, let

$$\Delta d_{ij} = -\Delta d^S/(N_d - 1), N_d = C \times (C-1)/2 \quad (10)$$

Since the class number is C, and there are $N_d$ pairs of classes including one pair being the smallest between-class distance. Therefore, the sum of changes for all $d_{ij}$ satisfies $\Delta d^S + \Delta d_{ij} \times (N_d - 1) = 0$ . In (9), there are two parameters $\sigma$ and $\delta$. With $\sigma$, the influence of $D_S$ on $\Delta d^S$ can be limited into a small scope. With a large initial value of $\delta$, the algorithm will converge faster while achieving a coarse sub-optimal result. With the iteration of algorithm, $\delta$ can be reduced in order to ensure that the obtained result is closer to a finer optimal solution.

## 3 The Two Dimensional F_LDA

In this section, we will extend F_LDA to F_2DLDA. For such an aim, let us first describe the main idea of F_LDA.

Suppose we wish to reduce the dimensionality from m to (m-1). To do this using LDA we would compute $S_b$ in (11) and its eigenvectors $\phi_1, \phi_2, ...\phi_m$, We would then project patterns onto the subspace S spanned by $\phi_1, \phi_2, ...\phi_{m-1}$, to obtain the $m-1$ dimensional representation. When many classes are present, it is possible that there is a pair of classes (say class p and class q) such that $\mu_p - \mu_q$ has approximately the same orientation as $\phi_m$. Since $\phi_m$ is orthogonal to S, the two classes would heavily overlap in the $m-1$ dimensional subspace S. Working backwards, we conclude that the vector $\mu_p - \mu_q$ was insufficiently weighted in computing $S_b$ using (11), which implies that (for a monotonically decreasing weighting function) $\mu_p - \mu_q$ must have been large, i.e., class p and class q were well-separated in the input space. However, instead of reducing the dimensionality from say n to $m-1$ directly, suppose we were to move "incrementally" towards a dimensionality of $m-1$. More explicitly, we gradually compress the data along the direction of the eigenvector $\phi_m$. At each incremental step, we recompute the between class scatter matrix based on the changed interclass distances, then compute its eigenvectors, thereby allowing those class centers that come closer together to be increasingly weighted, causing the $m-1$ dimensional subspace to reorient and avoid severe overlap of the two classes. Such F-LDA introduces a sort of "automatic gain control" that substantially improves the robustness of the algorithm to the choice of the weighting function.

A common form of an optimality criteria to be maximized is the function $J = tr(\hat{S}_b)$ where $\hat{S}_b$ is the between class scatter matrix in the output space and is given by,

$$\hat{S}_b = \sum_{i=1}^{C}\sum_{j=1}^{C} w(d_{ij})(\hat{\mu}_i - \hat{\mu}_j)(\hat{\mu}_i - \hat{\mu}_j)^\top \quad (11)$$

A common form of an optimality criteria to be maximized is the function $J = tr(S_b)$ where $S_b$ is the between class scatter matrix in the input space and is given by,

$$S_b = \sum_{i=1}^{C}\sum_{j=1}^{C} w(d_{ij})(\mu_i - \mu_j)(\mu_i - \mu_j)^\top \quad (12)$$

where, $d_{ij} = \|\mu_i - \mu_j\|$ is the Euclidean distance between the means of class i and class j in the input space, and the weighting function $w(d_{ij})$ is generally a monotonically decreasing function because classes that are closer to one another are likely to have a greater confusion and should be given a larger weighting. The scatter matrices in the input and output space are related by $\hat{S}_b = W^\top S_b W$.

The entire idea can be expressed for a reduced dimensionality of $p$ in terms of the following pseudo code, where r is the number of fractional steps used to reduce the dimensionality by 1. Similarly as in vector form [3], we propose the following matrix form for F_LDA as below. **F_2DLDA algorithm**

---

Set $W = I(m \times m)$ (the identity matrix)
for $k = m$ to $p+1$ step (-1)
for $l = 0$ to $r-1$ step 1
Project the data using W as y=W$^\top$ x
Apply the scaling transformation (defined below) to obtain $z = \psi(y, \alpha^l)$
For the z patterns, compute the $m \times k$ between class scatter matrix $\tilde{S}_b$
Compute the ordered eigenvalues $\lambda_1, \lambda_2, ...\lambda_k$ and corresponding
eigenvectors $\phi_1, \phi_2, ...\phi_k$ of $\tilde{S}_b$.
Set $W = W\Phi$, where $\Phi = [\phi_1, \phi_2, ...\phi_k]$
endfor

Discard the last ($k^{th}$) column of $W$.

endfor

The scaling transformation compresses the last component of y by a factor $\alpha^l$ with $\alpha < 1$, such that

$$z_i = \begin{cases} \alpha^l y_i & i = k \\ y_i & i = 1, 2, ..., (k-1) \end{cases} \quad (13)$$

The reduction factor increases in geometric progression with the steps of the inner loop. In the rth step, the reduction factor is $\alpha^{r-1}$. Therefore, a dimension is removed by successively scaling it by 1,$\alpha$, $\alpha^2$,..., $\alpha^{r-1}$. In implementation, a large number of steps should be used to collapse each dimension with the scaling factor chosen in relation to the number of steps. When the number of steps is smaller, its value should be chosen larger and vice versa.

It can be seen that both W_2DLDA and F_2DLDA are formulated into a form of weighted linear discriminant analysis mathematically. However, their motivations are totally different and also their solutions are not the same. Since they are similar in mathematic formulation, their performances should be compared in experiments. Such comparisons have not been done in literature [1] and we will do it in next section.

## 4 Experiment Results

In the experiments for W_LDA and F_LDA, the face matrix's dimensions are reduced by PCA from $10304 \times 400$ and $77760 \times 165$ to $279 \times 400$ and $368 \times 165$ respectively for the two face databases, which contain 98% information of the original matrix. For W_2DLDA and F_2DLDA, all the face matrix's dimensions are reduced by Simplified generalized low rank approximations of matrices (SGLRAM) from $112 \times 92$ and $320 \times 243$ to $29 \times 21$ and $52 \times 34$ respectively, which contain 98% information of the original matrices in ORL and YALE face databases. We expect the proposed two algorithms (W_2DLDA and F_2DLDA) can produce similar good performance as W_LDA and F_LDA while they can save computational costs significantly.

### 4.1 Evaluations

In this section we will carry out several experiments to demonstrate better performances of the proposed W_2DLDA and F_2DLDA and compare them with W_LDA and F_LDA, respectively. Further we compare the performances of W_2DLDA and F_2DLDA. As to the databases, we first use the ORL face database, which contains 400 images of 40 individuals (each person has 10 different images) under various facial expressions, wearing or not wearing glasses, different lighting conditions with each images being cropped and resized to 112×92 pixels in this experiment. For F_LDA and F_2DLDA inner loop time are both set to 10 . Then we randomly select 3,5,7,9 images for each person as training and the others images are used as testing, respectively. We select d = 100 (dimension is reduced from 279 to 100) for W_LDA and F_LDA and select d =5/6(dimension is reduced from 29 to 5/6) for W_2DLDA and F_2DLDA in the database ORL, respectively. The rate of recognition accuracy and time cost are the average of five times as shown in Table 1. Then, we use the YALE face database, which contains 165 gray scale images in GIF format of 15 individuals. There are 11 images per person, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. Each image is being cropped and resized to 320×243 pixels in this experiment. For F_LDA and F_2DLDA inner loop time are both set to be 10 . Then we randomly select 4,6,8,10 images for each person as training and the others images are used as testing, respectively. We select d = 100 (dimension is reduced from 368 to 100) for W_LDA and F_LDA and select d =8/9 (dimension is reduced from 52 to 8/9) for W_2DLDA and F_2DLDA in the database YALE, respectively. The rate of recognition accuracy and time cost are the average of five times as shown in Table 2.

Table 1: The Comparison of F_LDA,F_2DLDA, W_LDA,W_2DLDA on ORL database

| traning | | 3 | 5 | 7 | 9 |
|---|---|---|---|---|---|
| W_LDA | accuracy(%) | 71.2 | 80.5 | 93.3 | 95 |
| | time cost(s) | 8.30 | 8.58 | 8.57 | 8.71 |
| W_2DLDA | accuracy | 75.7 | 84.2 | 93.5 | 95.4 |
| | time cost | 12.3 | 14.6 | 17.9 | 20.5 |
| F_LDA | accuracy(%) | 71.9 | 80.3 | 93.3 | 95 |
| | time cost(s) | 403.6 | 441.9 | 436.8 | 521.3 |
| F_2DLDA | accuracy | 76.1 | 84 | 92.5 | 93.5 |
| | time cost | 12.49 | 14.81 | 17.07 | 18.19 |

From Table 1 we can see that the performance of W_2DLDA is better than that for W_LDA and time cost is slightly expensive due to the fact that it includes more effective information (since W_2DLDA compression's rate is smaller than W_LDA's ). However the time costs are comparable. We also can see the performance for F_2DLDA and F_LDA are nearly the same. However F_LDA costs much more time than F_2DLDA and we can know the time cost are closely the same. Whereas W_2DLDA algorithm performance is slight better than F_2DLDA with the training sample increasing.

From these experiments, owe can see that the F_2DLDA algorithm can save much more time compared to F_LDA, and W_2DLDA performance are the best for all these algorithms.

Next we do the experiment on Yale face database, which has 15 persons, each person has 11 images, The rate of recognition accuracy and time cost are the average of five times in YALE database as shown in Table 2.

Table 2: The Comparison of F_LDA,F_2DLDA, W_LDA,W_2DLDA on Yale database

| traning | | 4 | 6 | 8 | 10 |
|---|---|---|---|---|---|
| W_LDA | accuracy(%) | 91.4 | 89.3 | 97.8 | 95 |
| | time cost(s) | 0.28 | 0.33 | 0.41 | 0.31 |
| W_2DLDA | accuracy | 82.9 | 89.3 | 100 | 100 |
| | time cost | 0.75 | 0.80 | 0.64 | 0.27 |
| F_LDA | accuracy(%) | 92.4 | 90.7 | 97.8 | 100 |
| | time cost(s) | 22.3 | 22.6 | 22.5 | 22.5 |
| F_2DLDA | accuracy | 83.8 | 90.7 | 97.8 | 100 |
| | time cost | 2.0 | 2.3 | 2.6 | 2.8 |

From these experiments, one can see that F_2DLDA algorithm can save much more time compared to F_LDA. With large training samples, the performances for W_2DLDA and F_2DLDA are better than W_LDA and F_LDA as well as significant reduction of computational costs. Therefore W_2DLDA is a good choice when the number of training samples is large.

## References


[1] Fang Chen, Senjian An and Wanquan Liu, "A New Face Recognition Approach to Boosting the Worst-case Performance". *PCM* 2008, pp. 927-930.

[2] Xiao-Yuan Jing, David Zhang, and Yuan-Yan Tang ,"An improved LDA Approach", *IEEE Transaction on Systems, Man, And Cybernetics—Part B: Cybernetics,* 2004,34(5):pp. 1942-1951.

[3] Rohit Lotlikar and Ravi Kothari,"Fractional-Step Dimensionality Reduction",*IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2000,22(6):pp. 623-627.

[4] Ming Li,Bing Yuan, "2DLDA: A Novel Statistical-Llinear Discriminate Analysis for Image Matrix", *Pattern Recognition Lett,* 2005, 26(25): pp. 527-532.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherface: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.,* 1997,19: pp. 711–720.

[6] Chong Lu, Wanquan Liu and senjian An,"A Simplied GLRAM for face recognition", *Neurocomputing* (2008),doi:10.1016/j.neucom.2007.11.046.

[7] Jeffery R. Price, and Timothy F. Gee ,"Face recognition using direct, weighted linear discriminant analysis and modular subspaces", *Pattern Recognition,* 2005(38): pp. 209-219.

[8] Guang Dai, Dit-Yan Yeung and Yun-Tao Qian,"Face recognition using a kernel fractional-step discriminant analysis algorithm", *Pattern Recognition,* 2007(40): pp. 229-243 .

[9] Alok Sharma and Kuldip K. Paliwal,"A Gradient Linear Discriminant Analysis for Small Sample Sized Problem", *Neural Processing Letters,*, 2008, pp. 1370-4621.

[10] Juwei Lu, Plataniotis, K.N and Venetsanopoulos, A.N,"Face recognition using LDA-based algorithms", *IEEE Transactions on Neural Networks,* 2003(14): pp. 195-200.

[11] Senjian An, Wanquan Liu, and Svetha Venkatesh, "Exploiting side information in locality preserving projection", *CVPR* 2008: pp. 1-8.