Optimal Dimensionality Discriminant Analysis and Its Application to Image Recognition*

Feiping Nie[1], Shiming Xiang[2], Yangqiu Song[1] and Changshui Zhang[2]
State Key Laboratory of Intelligent Technology and Systems,
Department of Automation, Tsinghua University, Beijing 100080, China
[1]{nfp03, songyq99}@mails.tsinghua.edu.cn; [2]{xsm, zcs}@mail.tsinghua.edu.cn

Abstract

Dimensionality reduction is an important issue when facing high-dimensional data. For supervised dimensionality reduction, Linear Discriminant Analysis (LDA) is one of the most popular methods and has been successfully applied in many classification problems. However, there are several drawbacks in LDA. First, it suffers from the singularity problem, which makes it hard to preform. Second, LDA has the distribution assumption which may make it fail in applications where the distribution is more complex than Gaussian. Third, LDA can not determine the optimal dimensionality for discriminant analysis, which is an important issue but has often been neglected previously. In this paper, we propose a new algorithm and endeavor to solve all these three problems. Furthermore, we present that our method can be extended to the two-dimensional case, in which the optimal dimensionalities of the two projection matrices can be determined simultaneously. Experimental results show that our methods are effective and demonstrate much higher performance in comparison to LDA.

1. Introduction

Dimensionality reduction is an important issue when facing high-dimensional data, and many supervised dimensionality reduction algorithms have been proposed for the purpose of classification. Among those supervised algorithms, Linear Discriminant Analysis (LDA) is one of the most popular ones. It has been successfully applied in many classification tasks such as face recognition. However, there exist several drawbacks in LDA. First, it often suffer from the *small sample size* problem when dealing with high dimensional data. In this case, the within-class scatter matrix $\mathbf{S}_w$ may become singular, which makes LDA difficult to perform. Many approaches have been proposed to solve this problem [1, 2, 11]. However, these variants of LDA discard a subspace and some important discriminative information may be lost. As a result, the globally optimal subspace may not be found.

Another drawback of LDA is the parametric distribution assumption implicitly in it. LDA is optimal in the case that the data distribution of each class is homoscedastic Gaussian, which can not always be satisfied in real world applications. When the class distribution is more complex, LDA may fail to find the optimal discriminative projections.

In theory, the available projection dimensionality in LDA is smaller than the class number [3] , which is insufficient for some complex problems, especially when the class number is small. Moreover, based on the criterion of LDA, one can not determine the optimal dimensionality to be projected since the optimal value for the criterion in LDA is monotonic with respect to the projection dimensionality.

How to select a suitable dimensionality for discriminant analysis? This important issue was often neglected previously. In this paper, we tend to solve it. We propose a new criterion, of which the optimal value is not monotonic with respect to projection dimensionality. Furthermore, based on this criterion, the optimal value is guaranteed to reach the maximum in one of the reduced dimensionality. Therefore, the optimal dimensionality for discriminant analysis can be effectively determined. Simultaneously, the singularity problem in LDA does not occur naturally.

Recently, a technique called two-dimensional LDA [10] has been proposed for discriminant analysis. Unlike traditional LDA treating image as a vector by concatenating all its row vectors, two-dimensional LDA treats an image as a matrix directly. Although two-dimensional LDA has many significant merits, the optimal dimensionality still can not be determined automatically. While for the two-dimensional method, how to determine the optimal dimensionality becomes an even more important problem. Since there are two projection matrices to be determined in the two-dimensional method, it is much harder to select the suitable two dimensionalities compared with the one-

*This paper is Funded by Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

dimensional case. In this paper, we present that our method can be extended to the two-dimensional case, in which the optimal dimensionalities for the two projection matrices can be determined simultaneously.

The rest of this paper is organized as follows: In Section2, we give a brief review and an analysis of LDA. Another intrinsic drawback in LDA is deeply analyzed and the new neighborhood scatter matrices are constructed to solve this problem. In Section 3 and Section 4, we propose the optimal dimensionality discriminant analysis (ODDA) and the two-dimensional ODDA respectively. In Section 5, the toy examples and image recognition experiments are presented to demonstrate the effectiveness of our methods. Finally, we give the conclusions in Section 6.

2. Review of Linear Discriminant Analysis

Given the data matrix $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n], \boldsymbol{x}_i \in \mathbb{R}^d$, each data $\boldsymbol{x}_i (i = 1, ..., n)$ is associated with a class label from $\{1, 2, ..., c\}$. Denote $\mathcal{X}_i$ as the data set of class i and denote n_i as the number of data points in class i.

LDA is to learn a linear transformation$\mathbf{W} : \mathbb{R}^d \to \mathbb{R}^m$, and $\mathbf{W} \in \mathbb{R}^{d\times m}$. Then the original high-dimensional data $\boldsymbol{x}$ is transformed into a low-dimensional vector:

$$\boldsymbol{y} = \mathbf{W}^T \boldsymbol{x} \tag{1}$$

With the projection matrix $\mathbf{W}$, LDA tries to maximize the between-class scatter, while minimizing the within-class scatter. The within-class scatter matrix $\mathbf{S}_w$ and the between-class scatter matrix $\mathbf{S}_b$ are defined as

$$\mathbf{S}_w = \sum_{i=1}^{c} \sum_{\boldsymbol{x}_j \in \mathcal{X}_i} (\boldsymbol{x}_j - \boldsymbol{m}_i)(\boldsymbol{x}_j - \boldsymbol{m}_i)^T \tag{2}$$

$$\mathbf{S}_b = \sum_{i=1}^{c} n_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T \tag{3}$$

where $\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in \mathcal{X}_i} \boldsymbol{x}_j$ is the mean of the samples in class i and $\boldsymbol{m} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i$ is the mean of all the samples.

The projection matrix $\mathbf{W}^*$ in LDA is learned by solving the following optimization problem:

$$\mathbf{W}^* = \arg \max_{\mathbf{W} \in \mathbb{R}^{d\times m}} tr\left((\mathbf{W}^T \mathbf{S}_w \mathbf{W})^{-1} \mathbf{W}^T \mathbf{S}_b \mathbf{W} \right) \tag{4}$$

where $tr(\cdot)$ denotes the trace operator.

It has been known that the solution to this optimization problem is the m largest eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$, and the corresponding optimal value is $\sum_{i=1}^{m} \lambda_i$, where λ_i $(i = 1, 2, ..., m)$ are the first m largest eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$ and m is the projection dimensionality[4].

From the solution we can directly see the drawbacks of LDA. First, when $\mathbf{S}_w$ is singular, it is hard to be solved numerically. Second, the rank of $\mathbf{S}_b$ is usually equal to $c - 1$. This property makes the number of nonzero eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$ smaller than $c-1$, and thus result in the valid projection dimensionality not more than $c - 1$. Between the valid projection dimensionality m, the optimal value $\sum_{i=1}^{m} \lambda_i$ monotonically increase when m increase. Therefore, the "optimal" dimensionality of LDA is the rank of $\mathbf{S}_b$ (usually $c - 1$). For the case that data distribution of each class is homoscedastic Gaussian, it is certainly the genuine optimal dimensionality. However, for other cases, LDA may not find the optimal discriminative projections and the optimal dimensionality, which can be seen in the later toy examples. In the next section, we will analyze the third drawback and construct the new scatter matrices to solve it.

3. Constructing the Neighborhood Scatter Matrices

In this section, we analyze the within-class scatter and the between-class scatter in LDA, and illuminate the reason why LDA may fail to find the optimal discriminative projections in some complex cases, especially in the case that the data distribution is multimodal, i.e., the distribution of one class consists of multiple clusters.

The within-class scatter matrix in LDA can be easily reformulated as the pairwise form as follows:

$$\mathbf{S}_w = \frac{1}{2} \sum_{i=1}^{c} \sum_{\boldsymbol{x}_j \in \mathcal{X}_i} \frac{1}{n_i} \sum_{k=1}^{n_i} (\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_k)(\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_k)^T \tag{5}$$

where $\tilde{\boldsymbol{x}}_k \in \mathcal{X}_i$. We denote $\|\cdot\|$ as the L^2-norm of vector, i.e., $\|\boldsymbol{x}\|^2 = \boldsymbol{x}^T\boldsymbol{x}$. Then the within-class scatter can be written as:

$$tr(\mathbf{S}_w) = \frac{1}{2} \sum_{i=1}^{c} \sum_{\boldsymbol{x}_j \in \mathcal{X}_i} \frac{1}{n_i} \sum_{k=1}^{n_i} \|\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_k\|^2 \tag{6}$$

For the between-class scatter matrix in LDA, we can reformulate it in the pairwise form by utilizing the following relationship:

$$\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w \tag{7}$$

where $\mathbf{S}_t$ is the total scatter matrix. First, $\mathbf{S}_t$ can also be easily reformulated as the pairwise form as:

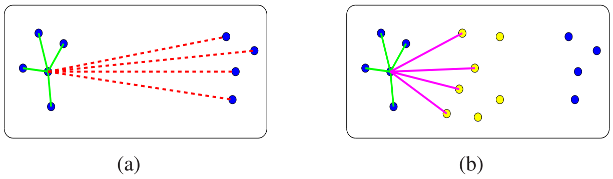
$$\mathbf{S}_t = \frac{1}{2} \sum_{i=1}^{c} \sum_{\boldsymbol{x}_j \in \mathcal{X}_i} \frac{1}{n} \sum_{k=1}^{n} (\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_k)(\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_k)^T \tag{8}$$

where $\tilde{\boldsymbol{x}}_k \in \mathbf{X}$, and the total scatter can be written as:

$$tr(\mathbf{S}_t) = \frac{1}{2} \sum_{i=1}^{c} \sum_{\boldsymbol{x}_j \in \mathcal{X}_i} \frac{1}{n} \sum_{k=1}^{n} \|\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_k\|^2 \tag{9}$$

Then the between-class scatter can be rewritten as the pairwise form according to (6), (7) and (9):

$$tr(\mathbf{S}_b) = tr(\mathbf{S}_t) - tr(\mathbf{S}_w) \tag{10}$$



(a)

(b)

Figure 1. *The blue color circles denote the points come from the same class i, while the yellow color circles denote the points come from the classes which are different to class i. (a) LDA try to minimize the distance of the far-apart pairs in the same class(dashed red line). (b) The k_w-neighbors(green line) and the k_b-neighbors(magenta line) of one data point.*

From (6) we can see, the within-class scatter in LDA is calculated from the sum of the squared Euclidean distances between each data point and all the other data points which belong to the same class. As LDA tries to minimize the within-class scatter, it would minimize all the data pairs which belong to the same class. This strategy is reasonable for the case that the class distribution is Gaussian. However, when the data in one class consists of more than one cluster, imposing all the pairs in the same class to be close is not a good strategy.

In order to eliminate this drawback, we construct the neighborhood within-class scatter to substitute the within-class scatter in LDA. In the same class, instead of calculating the distances between one data point and all the other data points, we only calculate the distances between the data point and its neighbors. As can be seen in Figure 1, by virtue of the neighborhood, we avoid to calculate the distances of the pairs which are far away from each other in the same class, as the dashed red line in Figure 1(a), which makes us focus more on the improvement of the discriminability of local structure.

Similarly, we can also construct the neighborhood total scatter and the neighborhood between-class scatter.

We first define the *k_w-neighbors* and *k_b-neighbors*. If $\boldsymbol{x}_i$ belongs to $\boldsymbol{x}_j$'s k_w nearest neighbors in the same class, we say $\boldsymbol{x}_i$ belongs to $\boldsymbol{x}_j$'s *k_w-neighbors*, which are denoted as the green line in Figure 1(b). If $\boldsymbol{x}_i$ belongs to $\boldsymbol{x}_j$'s k_b nearest neighbors in the classes that are different to the class of $\boldsymbol{x}_j$, we say $\boldsymbol{x}_i$ belongs to $\boldsymbol{x}_j$'s *k_b-neighbors*, which are denoted as the magenta line in Figure 1(b).

Then for each data point $\boldsymbol{x}_i$, we define the within-class neighborhood $\mathcal{N}_w(\boldsymbol{x}_i)$ and the between-class neighborhood $\mathcal{N}_b(\boldsymbol{x}_i)$. If $\boldsymbol{x}$ belongs to $\boldsymbol{x}_i$'s *k_w-neighbors* and $\boldsymbol{x}_i$ also belongs to $\boldsymbol{x}$'s *k_w-neighbors*, then $\boldsymbol{x}$ is in $\mathcal{N}_w(\boldsymbol{x}_i)$. If $\boldsymbol{x}$ belongs to $\boldsymbol{x}_i$'s *k_b-neighbors* and $\boldsymbol{x}_i$ also belongs to $\boldsymbol{x}$'s *k_b-neighbors*, then $\boldsymbol{x}$ is in $\mathcal{N}_b(\boldsymbol{x}_i)$.

Denote the number of data points in $\mathcal{N}_w(\boldsymbol{x}_i)$ as $k_w(i)$ and the number of data points in $\mathcal{N}_b(\boldsymbol{x}_i)$ as $k_b(i)$. Then in contrast to (6), the neighborhood within-class scatter can be defined as follows:

$$tr(\tilde{\mathbf{S}}_w) = \frac{1}{2}\sum_{i=1}^{c}\sum_{\boldsymbol{x}_j\in\mathcal{X}_i}\frac{1}{k_w(j)}\sum_{k=1}^{k_w(j)}\|\boldsymbol{x}_j-\tilde{\boldsymbol{x}}_k\|^2 \tag{11}$$

where $\tilde{\boldsymbol{x}}_k \in \mathcal{N}_w(\boldsymbol{x}_j)$. Similarly, in contrast to (9), the neighborhood total scatter can be defined as:

$$tr(\tilde{\mathbf{S}}_t) = \tfrac{1}{2}\sum_{i=1}^{c}\sum_{\boldsymbol{x}_j\in\mathcal{X}_i}\tfrac{1}{k_w(j)+k_b(j)}\sum_{k}\|\boldsymbol{x}_j-\tilde{\boldsymbol{x}}_k\|^2 \tag{12}$$

where $\tilde{\boldsymbol{x}}_k \in N_w(\boldsymbol{x}_j)\bigcup N_b(\boldsymbol{x}_j)$. Similar to (10), the neighborhood between-class scatter can be defined as:

$$tr(\tilde{\mathbf{S}}_b) = tr(\tilde{\mathbf{S}}_t) - tr(\tilde{\mathbf{S}}_w) \tag{13}$$

Based on (11) $\sim$ (13), The neighborhood within-class scatter matrix $\tilde{\mathbf{S}}_w$ and the neighborhood between-class scatter matrix $\tilde{\mathbf{S}}_b$ can be easily formulated as follows:

$$\tilde{\mathbf{S}}_w = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}^w_{ij}(\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_j)(\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_j)^T \tag{14}$$

$$\tilde{\mathbf{S}}_b = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}^b_{ij}(\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_j)(\boldsymbol{x}_i-\tilde{\boldsymbol{x}}_j)^T \tag{15}$$

where

$$\mathbf{A}^w_{ij} = \left\{\begin{array}{cl} \frac{1}{k_w(i)} & \tilde{\boldsymbol{x}}_j \in \mathcal{N}_w(\boldsymbol{x}_i) \\ 0 & otherwise \end{array}\right. \tag{16}$$

and

$$\mathbf{A}^b_{ij} = \left\{\begin{array}{cl} \frac{1}{k_w(i)+k_b(i)} & \tilde{\boldsymbol{x}}_j \in \mathcal{N}_b(\boldsymbol{x}_i) \\ \frac{1}{k_w(i)+k_b(i)} - \frac{1}{k_w(i)} & \tilde{\boldsymbol{x}}_j \in \mathcal{N}_w(\boldsymbol{x}_i) \\ 0 & otherwise \end{array}\right. \tag{17}$$

From the graph view of LDA [7], it can be seen that if $k_w(i) = n_{c(i)}$ and $k_b(i) = n - n_{c(i)}$, where $c(i)$ is the class label of data point i, the neighborhood scatter matrices $\tilde{\mathbf{S}}_w$ and $\tilde{\mathbf{S}}_b$ defined here is equal to $\mathbf{S}_w$ and $\mathbf{S}_b$ in LDA.

4. Optimal Dimensionality Discriminant Analysis

In this section, we utilize the weighted difference form to formulate the criterion based on the scatter metrics defined in Section 3, then the criterion is formulated as follows:

$$\mathcal{J} = tr\left(\tilde{\mathbf{S}}_b - \gamma\tilde{\mathbf{S}}_w\right) \tag{18}$$

where γ is the weighted coefficient. Our goal is to learn a projection matrix $\mathbf{W} \in \mathbb{R}^{d\times m}$, then the criterion in (18) becomes:

$$\mathcal{J}(\mathbf{W}) = tr\left(\mathbf{W}^T(\tilde{\mathbf{S}}_b - \gamma\tilde{\mathbf{S}}_w)\mathbf{W}\right) \tag{19}$$

We further add a constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ to avoid trivial solutions, where $\mathbf{I}$ is the $m \times m$ identity matrix. Thus the optimization problem for discriminant analysis can be formulated as:

$$\mathbf{W}^* = \arg \max_{\substack{\mathbf{W}\in\mathbb{R}^{d\times m}\\ \mathbf{W}^T\mathbf{W}=\mathbf{I}}} tr\left(\mathbf{W}^T(\tilde{\mathbf{S}}_b - \gamma\tilde{\mathbf{S}}_w)\mathbf{W}\right) \qquad (20)$$

Usually the available data is limited and there are noises in data, so we can make a reasonable assumption that the performance should be improved when the dimensionality is reduce by discriminant analysis. Under this assumption, we could suppose that the criterion in (19) could be zero as the baseline when no dimensionality reduction is performed, and would reach a positive value when the dimensionality is reduced. Thus, we have

$$\mathcal{J} = tr\left(\tilde{\mathbf{S}}_b - \gamma\tilde{\mathbf{S}}_w\right) = 0 \implies \gamma = \frac{tr\tilde{\mathbf{S}}_b}{tr\tilde{\mathbf{S}}_w} \qquad (21)$$

We further let the projection dimensionality m be a variable and optimize it so that the optimal value of the criterion (19) reaches the maximum. We define matrix $\mathbf{S}$ as:

$$\mathbf{S} = \tilde{\mathbf{S}}_b - \frac{tr\tilde{\mathbf{S}}_b}{tr\tilde{\mathbf{S}}_w}\tilde{\mathbf{S}}_w \qquad (22)$$

thus the optimization problem can be formulated as:

$$\mathbf{W}^* = \arg \max_{\substack{\mathbf{W}\in\mathbb{R}^{d\times m}\\ \mathbf{W}^T\mathbf{W}=\mathbf{I}\\ m\in\{1,\ldots,d\}}} tr\left(\mathbf{W}^T\mathbf{S}\mathbf{W}\right) \qquad (23)$$

It is interesting to note that if we substitute $\tilde{\mathbf{S}}_t(= \tilde{\mathbf{S}}_b + \tilde{\mathbf{S}}_w)$ for $\tilde{\mathbf{S}}_b$ in (22), the matrix $\mathbf{S}$ remains unchanged, and thus does not impact the solution in (23), which is similar to the property of LDA. Moreover, if $\tilde{\mathbf{S}}_b$ or $\tilde{\mathbf{S}}_w$ in (22) is scaled with a constant, the solution in (23) is also unchanged. This scale invariant property is important, especially when we adopt the difference form (18) as criterion.

Denote $\mathbf{W} \in \mathbb{R}^{d\times m}$ by $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_m]$, where $\boldsymbol{w}_i(i = 1, 2, ...m)$ are d-dimensional column vectors. Suppose the value of m is given, according to the result of Rayleigh quotient [5], when $\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_m$ are the first m largest eigenvectors of $\mathbf{S}$, the optimal value of the above optimization problem is $\sum_{i=1}^{m}\lambda_i$, where $\lambda_i(i = 1, 2, ..., m)$ are the first m largest eigenvalues of $\mathbf{S}$. Thus, when m is equal to the number of positive eigenvalues of $\mathbf{S}$, the optimal value reaches the maximum. Therefore, the optimal solution to the optimization problem in (23) can be explicitly calculated by eigenvalue decomposition.

Note that $tr(\mathbf{S}) = 0$, then $\sum_{i=1}^{d}\lambda_i = 0$, which implies that the optimal value in (23) will definitly reach the maximum in one of the reduced dimensionality.

It can be verified that $null(\tilde{\mathbf{S}}_t) = null(\tilde{\mathbf{S}}_w) \cap null(\tilde{\mathbf{S}}_b)$, where $null(\cdot)$ denotes the null space of matrix. Therefore, the null space of $\tilde{\mathbf{S}}_t$ does not consist of any useful discriminative information. We could eliminate the null space of $\tilde{\mathbf{S}}_t$ and then perform the algorithm in this lower dimensional subspace, which can significantly improve the computation speed when the dimension of original data is very high.

The algorithm is described in Table 1. We can see that the singularity problem in LDA does not exist in our algorithm naturally.

```
0. Preprocessing:
   Eliminate the null space of $\tilde{\mathbf{S}}_t$, and obtain new data
   $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$, where $rank(\mathbf{X}) = d$
1. Input:
   $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$, $k_w$, $k_b$
2. Calculate $\tilde{\mathbf{S}}_w$ and $\tilde{\mathbf{S}}_b$ according to (14) and (15)
3. Calculate the eigenvalues and the corresponding
   eigenvectors of $\mathbf{S}$ defined in (22)
4. Select the $m$ largest eigenvectors to form $\mathbf{W}$, where
   $m$ is equal to the number of positive eigenvalues of
   $\mathbf{S}$.
5. Output:
   $\mathbf{W}$, $m$ and $\boldsymbol{y} = \mathbf{W}^T\boldsymbol{x}$
```

Table 1. *Algorithm of ODDA*

5. Two-Dimensional ODDA

For the application to image recognition, traditional discriminant analysis treats each image as a vector, i.e., the image is converted to a vector by concatenating all its row vectors consecutively. Recently, two-dimensional LDA has been proposed for discriminant analysis. The technique treats an image as a matrix directly, which has three significant advantages compared with the vector based method. First, it can utilize the intrinsic spatial structure information of image. Second, the matrix dimension for eigenvalue decomposition is much smaller, which can effectively avoid the *curse of the dimensionality* dilemma and dramatically reduce the time and space cost. Finally, it avoids the *small sample size* problem and can extract more discriminative projections than $c - 1$, even than $n - 1$, where c is the number of classes and n is the number of training data.

Although two-dimensional LDA can extract more discriminative projections, the optimal dimensionality still can not be determined by the algorithm. For the two-dimensional method, how to determine the optimal dimensionality is a more important problem. In the one-dimensional case, one should only choice the dimensionality from 1 to d, while in the two-dimensional case, since there are two projection matrices to be determined, the number of possible choices becomes $h * w$ (image height and width). In this section, we extend our method and propose the algorithm called two-dimensional optimal dimensionality discriminant analysis (2DODDA).

Let $\mathbf{G}_i \in \mathbb{R}^{h\times w}$ $(i = 1, ..., n)$ be the n images, where h and w is the height and width of image respectively, n is the number of images. We consider the two projection matrices $\mathbf{U} \in \mathbb{R}^{h\times l}$ $(1 \leq l \leq h)$ and $\mathbf{V} \in \mathbb{R}^{w\times r}$ $(1 \leq r \leq w)$. The transformation on image data $\mathbf{G}_i$ is defined as:

$$\mathbf{H}_i = \mathbf{U}^T\mathbf{G}_i\mathbf{V} \tag{24}$$

The criterion in (18) can be reformulated as:

$$tr(\tilde{\mathbf{S}}_b - \gamma\tilde{\mathbf{S}}_w) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}\left\|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_j\right\|^2 \tag{25}$$

where $\mathbf{A}_{ij} = \mathbf{A}_{ij}^b - \gamma\mathbf{A}_{ij}^w$, $\mathbf{A}_{ij}^w$ and $\mathbf{A}_{ij}^b$ are defined in (16) and (17) respectively. Similarly, in the two-dimensional case, the criterion can be formulated as:

$$\mathcal{J} = tr(\tilde{\mathbf{S}}_b - \gamma\tilde{\mathbf{S}}_w) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}\left\|\mathbf{G}_i - \tilde{\mathbf{G}}_j\right\|_F^2 \tag{26}$$

where $\|\cdot\|_F$ is the Frobenius norm of matrix, i.e., $\|\mathbf{M}\|_F^2 = tr(\mathbf{M}^T\mathbf{M})$ for any matrix $\mathbf{M}$.

Like the one-dimensional case in (21), we let the value of the criterion in (26) be equal to zero, namely,

$$\mathcal{J} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}\left\|\mathbf{G}_i - \tilde{\mathbf{G}}_j\right\|_F^2 = 0 \tag{27}$$

Then the value of γ in (26) can be calculated according to (27). It can be verified that the value is just the same as the one in the one-dimensional case.

Under the projection matrices $\mathbf{U}$ and $\mathbf{V}$, the criterion in (26) becomes:

$$\mathcal{J}(\mathbf{U},\mathbf{V}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}\left\|\mathbf{U}^T\mathbf{G}_i\mathbf{V} - \mathbf{U}^T\tilde{\mathbf{G}}_j\mathbf{V}\right\|_F^2 \tag{28}$$

We define two matrices as follows:

$$\mathbf{S}^v = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}(\mathbf{G}_i - \tilde{\mathbf{G}}_j)\mathbf{V}\mathbf{V}^T(\mathbf{G}_i - \tilde{\mathbf{G}}_j)^T \tag{29}$$

$$\mathbf{S}^u = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{ij}(\mathbf{G}_i - \tilde{\mathbf{G}}_j)^T\mathbf{U}\mathbf{U}^T(\mathbf{G}_i - \tilde{\mathbf{G}}_j) \tag{30}$$

Then (28) can be rewritten as:

$$\mathcal{J}(\mathbf{U},\mathbf{V}) = tr\left(\mathbf{U}^T\mathbf{S}^v\mathbf{U}\right) = tr\left(\mathbf{V}^T\mathbf{S}^u\mathbf{V}\right) \tag{31}$$

In the two-dimensional case, our goal is turned to learn the two projection matrices $\mathbf{U} \in \mathbb{R}^{h\times l}$ and $\mathbf{V} \in \mathbb{R}^{w\times r}$. We further add the constraint $\mathbf{U}^T\mathbf{U} = \mathbf{I_u}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I_v}$ to avoid trivial solutions, where $\mathbf{I_u}$ is the $l \times l$ identity matrix and $\mathbf{I_v}$ is the $r \times r$ identity matrix. Thus the optimization problem for the discriminant analysis can be formulated as:

$$\{\mathbf{U}^*, \mathbf{V}^*\} = \arg\max_{\substack{\mathbf{U}\in\mathbb{R}^{h\times l}\\ \mathbf{U}^T\mathbf{U}=\mathbf{I_u}\\ \mathbf{V}\in\mathbb{R}^{w\times r}\\ \mathbf{V}^T\mathbf{V}=\mathbf{I_v}\\ l\in\{1,\ldots,h\}\\ r\in\{1,\ldots,w\}}} \mathcal{J}(\mathbf{U},\mathbf{V}) \tag{32}$$

It is very difficult to optimize l, r, $\mathbf{U}$ and $\mathbf{V}$ simultaneously, but if one of $\mathbf{U}$ and $\mathbf{V}$ is known, the optimization problem becomes easy to be solved. Hence we derive an iterative algorithm to solve the optimization problem in (32). More concretely, for a fixed r and $\mathbf{V}$, we can compute the optimal l and $\mathbf{U}$ by solving the following optimization problem:

$$\mathbf{U}^* = \arg\max_{\substack{\mathbf{U}\in\mathbb{R}^{h\times l}\\ \mathbf{U}^T\mathbf{U}=\mathbf{I_u}\\ l\in\{1,\ldots,h\}}} tr\left(\mathbf{U}^T\mathbf{S}^v\mathbf{U}\right) \tag{33}$$

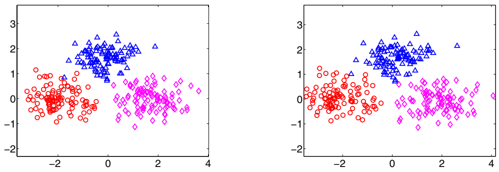
With the computed l and $\mathbf{U}$, we can then update r and $\mathbf{V}$ by solving another optimization problem as:

$$\mathbf{V}^* = \arg\max_{\substack{\mathbf{V}\in\mathbb{R}^{w\times r}\\ \mathbf{V}^T\mathbf{V}=\mathbf{I_v}\\ r\in\{1,\ldots,w\}}} tr\left(\mathbf{V}^T\mathbf{S}^u\mathbf{V}\right) \tag{34}$$

The optimization problems in (33) and (34) are just as the same as that in (23). The detailed procedure is described in Table 2.

```
1. Input:  Images $\mathbf{G}_i \in \mathbb{R}^{h\times w}(i = 1, ..., n)$ , $k_w$,
           $k_b$, $iter\_num$
2. Initialize $\mathbf{V}$ as an arbitrary colummly orthogonal
   matrix.
3. For $iter$ = 1 to $iter\_num$
     a) Calculate $\mathbf{S}^v$ according to (29).
     b) Select the $l$ largest eigenvectors of $\mathbf{S}^v$ to form
        $\mathbf{U}$, where $l$ is equal to the number of positive
        eigenvalues of $\mathbf{S}^v$.
     c) Calculate $\mathbf{S}^u$ according to (30).
     d) Select the $r$ largest eigenvectors of $\mathbf{S}^u$ to form
        $\mathbf{V}$, where $r$ is equal to the number of positive
        eigenvalues of $\mathbf{S}^u$.
   End For.
4. Output:  $\mathbf{U}, \mathbf{V}, l, r$  and $\mathbf{H} = \mathbf{U}^T\mathbf{G}\mathbf{V}$
```

Table 2. *The algorithm for the two-dimensional ODDA*



(a) LDA (b) ODDA

Figure 2. *The two dimensional projections learned by LDA and ODDA on the ten-dimensional toy dataset.*

It is easy to verify that the algorithm described in Table 2 guarantees that the value of criterion in (28) monotonically monotonously increases with respect to the iteration number. Although this alternative optimization strategy theoretically can only obtain a locally optimal solution, extensive experiments on image datasets show that the algorithm essentially converges to the same solution, regardless of the choice of the initial $\mathbf{V}$, which implies that the algorithm may always converge to the global optimum for the applications to image datasets. While in 2DLDA, the value to be optimized may sway with respect to the iteration number, which may result in the performance varying irregularly.

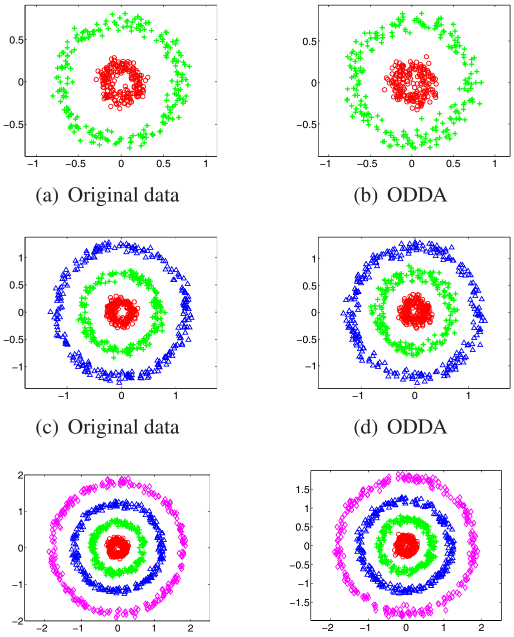
6. Experimental Results

6.1. Toy Examples

We present two toy examples to demonstrate the effectiveness of ODDA. In the first example, we generate a ten-dimensional dataset with three classes, each of which is sampled from a homoscedastic Gaussian distribution with the same covariance matrix.

Figure 2 illustrates the results learned by LDA and ODDA. The projected data of each class are shown by different colors and shapes. The results demonstrate that both LDA and ODDA can find the optimal discriminative projections and the optimal dimensionality in the case that the data distribution of each class is homoscedastic Gaussian.

In the second example, we generate three ten-dimensional datasets with two classes, three classes and four classes, respectively. In the first two dimensions, the classes are distributed in concentric circles, while the other eight dimensions are Gaussian noise with large variance. Thus, the optimal projections are the first two dimensions, and the optimal dimensionality is 2 in all the three datasets. LDA fails to find the optimal projections for all the three datasets and fails to determine the optimal dimensionality for the datasets with two classes and four classes, where the dimensionality determined by LDA are 1 and 3 respectively. Figure 3 shows the results learned by ODDA. In all of the three datasets, ODDA can find the optimal projections, and the optimal dimensionality determined by ODDA is just the real optimal number of 2.



(a) Original data (b) ODDA

(c) Original data (d) ODDA

(e) Original data (f) ODDA

Figure 3. *Left figures are the first two dimensions of the original datasets, right figures are the corresponding results learned by ODDA. LDA fails to find the optimal subspace in all the three cases and the results do not displayed here.*

6.2. Real World Image Recognition

We evaluated the algorithms ODDA and 2DODDA on four popular image databases, and compared it with LDA, 2DLDA [10], and the two variants of LDA: null space LDA (NLDA) [2] and direct LDA (DLDA) [11].

For NLDA and DLDA, we use PCA as the preprocessing step to eliminate the null space of $\mathbf{S}_t$. For LDA, due to the singularity problem in it, we further reduce the dimension of data such that $\mathbf{S}_w$ is nonsingular.

In each experiment, we randomly select several samples per class for training and the remaining samples are used for testing. The average results and standard deviations are reported over 50 random splits. The classification is based on k-nearest neighbor classifier ($k = 1$ in these experiments).

It is worth noting that the parameters in our methods are not sensitive. In the experiments, we simply set k_b to 20, and set k_w to $t/2$ for each data set, where t is the training number per class.

The experimental results are reported in Table 3. For LDA, NLDA, DLDA, the projection dimensionality is set to the rank of $\mathbf{S}_b$. For NLDA, projection dimensionality is set to the dimensionality of the null space of $\mathbf{S}_w$. Usually, these dimensionalities are equal to $c - 1$, where c is the number of classes. For 2DLDA, as it is hard to select the full dimensionalities combination for the two projection matrices, we let the two dimensionalites be the same number. The results are recorded under different dimensionalities from 1^2 to $min(h, w)^2$ and the best result is reported in Table 3, where the 'dim' is the corresponding dimensionalites. For ODDA and 2DODDA, the dimensionalities are automatically determined, and the 'dim' in Table 3 is the average value of m and $l * r$ over 50 random splits, respectively.

6.2.1 Face Recognition

For the face recognition application, we use two popular face databases to validate our algorithms.

The AT&T face database includes 40 distinct individuals and each individual has 10 different images [9]. Each image in the database is of size 112 × 92 and with 256 gray-levels. Some images are shown in Figure 4(a). Each image is down-sampled to the size of 28 × 23 to save the computation time and no other preprocessing is preformed. We randomly select 2,4 or 6 samples per class for training and the remaining samples for testing.

As can be seen in Table 3, the results of both ODDA and 2DODDA are much better than those of LDA whether in terms of accuracy or stability. Compared with 2DLDA, 2DODDA shows a better performance when the training number is very small.

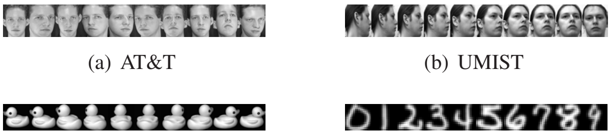
The UMIST repository is a multiview database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views. The size of each cropped image is 112×92 with 256 gray-levels per pixel [6]. Some images are shown in Figure 4(b). We down-sample the size of each image to 28 × 23 and no other preprocessing is preformed. 4,6 or 8 samples per class are randomly selected for training and the remaining samples for testing.

Although NLDA performs best on this data set, our methods also demonstrate the competitive performances and still show the better results than those of LDA. Furthermore, 2DODDA outperforms 2DLDA in all the cases.

6.2.2 Object Recognition

The COIL-20 database [8] consists of images of 20 objects viewed from varying angles at the interval of five degrees, resulting in 72 images per object. Some images are shown in Figure 4(c). Each image is down-sampled to the size of 32 × 32 and we randomly select 4,6 or 8 samples per class for training and the remaining samples for testing.

On this data set, both of our methods demonstrate the best performances. 2DODDA shows the excellent results in all the cases and ODDA also demonstrates the better results than those of LDA, NLDA, DLDA and 2DLDA.



(a) AT&T

(b) UMIST

(c) COIL-20

(d) USPS

Figure 4. *Some sample images of the four databases*

6.2.3 Digit Recognition

In this experiment, we focus on the digit recognition task using the USPS handwritten 16 × 16 digits data set[1]. Some images are shown in Figure 4(d). We randomly select 20,40 or 60 samples per class for training and the remaining samples for testing.

On this data set, LDA and NLDA do not work well, and for the cases of 40 and 60 samples per class for training, the matrix $\mathbf{S}_w$ is nonsingular, thus the null space of $\mathbf{S}_w$ does not exist, which makes NLDA not work. Both of our methods still demonstrate the best performances on this data set. Different from the previous experiments, the optimal dimensionality determined by 2DODDA is much lower than that determined by ODDA, which implies that a relatively lower dimensionality may also work well on this data set.

7. Conclusions

Extracting the optimal dimensionality is an important problem which is often neglected previously. In this paper, we propose a new method, *Optimal Dimensionality Discriminant Analysis* (ODDA), to solve this problem. ODDA focuses more on the improvement of the discriminability of local structure, which is especially useful when the distribution of each class is more complex than Gaussian. Moreover, ODDA effectively avoids the singularity problem and can automatically determine the optimal dimensionality for discriminant analysis. Toy examples and real world experiments on image recognition are presented to validate it.

For the image application, two-dimensional method has many significant merits. However, determining the optimal dimensionality becomes an more important problem in the two-dimensional case. In this paper, we extend our method and propose the algorithm called 2DODDA. In comparison with 2DLDA, 2DODDA can automatically determine the optimal dimensionalities of the two projection matrices. Furthermore, our algorithm guarantees that the solution increases monotonically with respect to the iteration number, and may always converge to the global optimum for the applications to image datasets. Experiments demonstrate that 2DODDA outperforms 2DLDA in most cases.

[1] Available at http://www.kernel-machines.org/data

data set	method	2 train			4 train			6 train		
		Acc.(%)	Dev.(%)	dim	Acc.(%)	Dev.(%)	dim	Acc.(%)	Dev.(%)	dim
AT&T	LDA	75.5	2.9	39	89.4	2.1	39	92.2	2.1	39
	NLDA	84.3	2.6	39	93.1	2.0	39	95.6	1.4	39
	DLDA	78.9	2.9	39	91.1	2.2	39	96.1	1.2	39
	2DLDA	83.3	2.1	14^2	93.8	1.8	8^2	**97.1**	1.2	8^2
	ODDA	84.1	2.8	39.0	**94.2**	1.6	52.7	97.0	1.2	65.6
	2DODDA	**85.5**	2.6	80.3	93.9	1.8	67.7	96.9	1.3	65.1

data set	method	4 train			6 train			8 train		
		Acc.(%)	Dev.(%)	dim	Acc.(%)	Dev.(%)	dim	Acc.(%)	Dev.(%)	dim
Umist	LDA	84.7	3.3	19	91.0	2.3	19	94.1	1.9	19
	NLDA	**89.6**	2.9	19	**94.6**	1.6	19	**97.0**	1.5	19
	DLDA	80.8	3.2	19	90.0	2.1	19	94.8	2.0	19
	2DLDA	87.8	3.6	11^2	93.8	1.7	11^2	96.6	1.3	9^2
	ODDA	87.5	3.2	29.6	94.1	1.8	37.6	**97.0**	1.5	44.2
	2DODDA	88.8	2.6	96.8	94.0	1.5	61.4	**97.0**	1.3	63.2

data set	method	4 train			6 train			8 train		
		Acc.(%)	Dev.(%)	dim	Acc.(%)	Dev.(%)	dim	Acc.(%)	Dev.(%)	dim
Coil20	LDA	76.8	2.5	19	82.2	1.7	19	85.0	1.7	19
	NLDA	81.2	2.8	19	86.3	1.7	19	89.1	1.7	19
	DLDA	79.6	2.6	19	86.1	1.7	19	90.1	1.3	19
	2DLDA	79.5	3.6	15^2	86.1	2.3	10^2	89.3	2.0	8^2
	ODDA	83.1	2.5	24.2	88.6	1.6	32.7	92.1	1.3	40.2
	2DODDA	**89.9**	1.9	96.5	**93.8**	1.4	90.2	**95.8**	1.0	90.8

data set	method	20 train			40 train			60 train		
		Acc.(%)	Dev.(%)	dim	Acc.(%)	Dev.(%)	dim	Acc.(%)	Dev.(%)	dim
USPS	LDA	52.8	2.6	9	72.3	1.3	9	82.2	0.9	9
	NLDA	52.3	2.7	9	-	-	-	-	-	-
	DLDA	85.8	1.1	9	88.3	0.7	9	89.2	0.5	9
	2DLDA	83.3	2.6	8^2	88.8	1.4	8^2	90.8	0.9	9^2
	ODDA	**87.5**	0.9	49.5	89.6	0.6	72.2	90.9	0.5	77.5
	2DODDA	85.9	2.6	21.6	**90.2**	0.8	27.2	**91.3**	0.6	34.2

Table 3. *Experimental results in each data set. The 'Acc.' is the accuracy over 50 random splits, the 'Dev.' is the standard deviation. For 2DLDA, the 'dim' is the corresponding dimensionality of the best result. For ODDA and 2DODDA, the 'dim' is the average value of m and $l * r$ over 50 random splits, respectively.*

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997. 1

[2] 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(10):1713–1726, October 2000. 1, 6

[3] R. O. Duda., P. E. Hart., and D. G. Stork. *Pattern Classification.* Wiley-Interscience, Hoboken, NJ, 2000. 1

[4] K. Fukunaga. *Introduction to Statistical Pattern Recognition,2nd Edition.* Academic Press, Boston, MA, 1990. 2

[5] G. H. Golub and C. F. van Loan. *Matrix Computations, 3rd Edition.* The Johns Hopkins University Press, Baltimore, MD, USA, 1996. 4

[6] D. B. Graham and N. M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. in face recognition: From theory to applications. *NATO ASI Series F, Computer and Systems Sciences.* 7

[7] X. F. He, S. C. Yan, Y. X. Hu, P. Niyogi, and H. J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, March 2005. 3

[8] S. A. Nene, S. K. Nayar, and H. Murase. *Columbia object image library (COIL-20), Technical Report CUCS-005-96.* Columbia University, 1996. 7

[9] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994. 7

[10] J. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in NIPS 17*, pages 1569–1576. MIT Press, Cambridge, MA, 2005. 1, 6

[11] H. Yu and J. Yang. A direct lda algorithm for high-dimensional data - with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001. 1, 6