Discriminant Common Vecotors Versus Neighbourhood Components Analysis and Laplacianfaces: A comparative study in small sample size problem

Jun Liu Songcan Chen*

Dept of Computer Science & Engineering, Nanjing University of Aeronautics & Astronautics

Nanjing, 210016, P.R. China

Abstract: Discriminant Common Vecotors (DCV), Neighbourhood Components Analysis (NCA) and Laplacianfaces (LAP) are three recently proposed methods which can effectively learn linear projection matrices for dimensionality reduction in face recognition, where the dimension of the sample space is typically larger than the number of samples in the training set and consequently the so-called small sample size (SSS) problem exists. The three methods obtained their respective projection matrices based on different objective functions and all claimed to be superior to such methods as Principal Component Analysis (PCA) and PCA plus Linear Discriminant Analysis (PCA+LDA) in terms of classification accuracy. However, in literature, no comparative study is carried out among them. In this paper, we carry out a comparative study among them in face recognition (or generally in the SSS problem), and argue that the projection matrix yielded by DCV is the optimal solution to both NCA and LAP in terms of their respective objective functions, whereas neither NCA nor LAP may get their own optimal solutions. In addition, we show that DCV is more efficient than both NCA and LAP for both linear dimensionality reduction and subsequent classification in SSS problem. Finally, experiments are conducted on ORL, AR and YALE face databases to verify our arguments and to present some insights for future study.

Key words: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Discriminant Common Vecotors (DCV), Neighbourhood Components Analysis (NCA), Laplacianfaces (LAP), Small Sample Size (SSS), Face Recognition

* Corresponding author: Tel: +86-25-84892452, Fax: +86-25-84498069. Email: s.chen@nuaa.edu.cn, j.liu@nuaa.edu.cn.

1 Introduction

In face recognition, we usually employ appearance-based methods [1, 2]. One primary advantage of appearance-based methods is that it is not necessary to create representations or models for face images since, for a given face image, its model is now implicitly defined in the face image itself [3]. When using appearance-based methods, we usually represent an image of size $r \times c$ pixels by a vector in a d-dimensional space, where $d=rc$. Although such an appearance based representation is simple in form, the corresponding dimensionality d is too large to realize robust and fast recognition [3], and is typically lager than the number of samples in the training set which leads to the so-called small sample size (SSS) problem. A common way to resolve this problem is to use dimensionality reduction techniques. Discriminant Common Vecotors (DCV) [7, 8], Laplacianfaces (LAP) [17] and Neighbourhood Components Analysis (NCA) [18] are three recently proposed methods which can effectively learn linear projection matrices for dimensionality reduction in face recognition.

DCV [7, 8] aims at solving the small sample size (SSS) problem in Linear Discriminant Analysis (LDA) [4-6] which maximizes the Fisher's Linear Discriminat criterion as follows:

$$J_{FLD}(W_{opt}) = \arg\max_{W} \left| W^T S_b W \right| \Big/ \left| W^T S_w W \right| \tag{1}$$

where S_w is the within-class scatter matrix and S_b is the between-scatter matrix. When the SSS problem takes place, S_w will be typically singular and LDA can not be applied directly. DCV[1] remedies this by calculating the projection matrix in the null space of S_w, and as a result gets an optimum (infinite) of objective function (1).

LAP [17] originates from viewpoint of preserving the locality structure of the image space. To this end, it models a manifold [13-15] structure by a nearest-neighbor graph, constructs a face subspace by Locality Preserving Projections (LPP) [16], and performs dimensionality reduction by a set of feature images called Laplacianfaces.

NCA [18] aims at learning a Mahalanobis distance measure to be used in the k Nearest Neighbor (KNN) classification. Subtly, it boils learning such Mahalanobis distance down to learning a linear projection (or transformation) matrix, and at the same time avoids the inverse

[1] Note that, we have proved in [9] that such null space based methods [10] as Generalised K-L Expansion (GKLE) [11], PCA plus Null Space (PNS) [12] and DCV are in fact equivalent, so our discussion of DCV in this paper can naturally be extended to both GKLE and PNS.

operation of the matrix in calculating traditional Mahalanobis distance metric. The projection matrix in NCA is obtained through optimizing to the KNN leave-one-out (LOO) classification performance on the training set, so the learned Mahalanobis distance metric or equivalently the projection matrix is directly related to the classification performance. This is the main characteristic of NCA, and is quite different from the dimensionality reduction methods mentioned above (e.g., DCV, LAP) whose objective functions are not directly associated with the classification decision. By restricting the projection matrix in the distance measure learning to a non-square one, NCA can be used for dimensionality reduction [18].

The three methods all claimed to be superior to Principal Component Analysis (PCA) [1] and PCA+LDA [4-6], namely: 1) DCV is superior to PCA and PCA+LDA in terms of recognition accuracy, efficiency and numerical stability [8]; 2) PCA and PCA+LDA can be obtained from different graph models in LAP, and LAP provides a better representation and achieves lower classification error rates in face recognition [17]; and 3) When labeled data is available, NCA performs better both in terms of classification performance in the projected representation and in terms of visualization of class separation as compared to the standard methods of PCA and LDA [18]. However, there is no comparative study among them in literature. The purpose of this paper is to compensate this by a comparative study among them, and to get some sight from such comparative study. It is worthwhile to highlight our contributions in this paper as follows:

1) We for the first time in literature perform a comparative study among DCV, LAP and NCA, and argue that in SSS problem (e.g. face recognition) the projection matrix yielded by DCV is the optimal solution to both NCA and LAP in terms of their respective objective functions, whereas neither NCA nor LAP may get their own optimal solutions.
2) We show that DCV is more efficient than both NCA and LAP for both linear dimensionality reduction and subsequent classification in SSS problem
3) We reveal the essence of DCV, i.e., calculating the projection matrix is equivalent to solving a thin QR decomposition problem, which is easier for both understanding and being extended to its nonlinear version by kernel trick [22-25].
4) We experimentally give the application scope of DCV, namely, when *MSV* (defined in section 4) is relatively small, it performs well while on the contrary when *MSV* is relatively large, it performs poorly.

The rest of the paper is organized as follows. In section 2, DCV, NCA and LAP are respectively reviewed. In section 3, we carry out a comparative study among these three methods in SSS problem. In section 4, we report experimental results on several face databases. Finally in section 5, we provide some concluding remarks and suggestions for future work.

2 Review of the three methods

Let the training set be composed of C classes, with the i-th class containing N_i (>1) d-dimensional samples. Suppose that the training samples are linearly independent, which can be generally satisfied in such applications as face classification. Then there will be a total of $M=N_1+N_2+...+N_C$ linearly independent training samples. Note that in such high dimension data classification as face recognition, the SSS problem exists, namely $d>>M$ generally holds.

We give two equivalent descriptions of the training samples in order to review the three methods clearly and concisely utilizing the corresponding descriptions in [8, 17, 18] respectively. More specifically, such two equivalent descriptions are as follows:

1) Description $\{x_j^i\}$: Let x_j^i be a d-dimensional column vector which denotes the j-th sample from the i-th class, and then $X=[x_1^1, x_2^1, \cdots, x_{N_1}^1, x_1^2 \cdots x_{N_C}^C]$ contains all the training samples;

2) Description $\{y_i, z_i\}$: Let y_i be the i-th column in X and the according class label in z_i.

It is obvious that the two descriptions have the following relationship:

$$y_k = x_j^i \quad iff \quad k = N_1 + N_2 + \cdots N_{i-1} + j \quad and \quad z_k = i \tag{2}$$

We use the first description for DCV and the second description for both NCA and LAP.

2.1 Discriminant Common Vectors (DCV)

Before describing DCV, we first introduce the idea of common vectors from which DCV is originated. The idea of common vectors is originally introduced for isolated word recognition problems [19, 20] in the case where the number of samples in each class is less than or equal to the dimensionality of sample space. These approaches extract the common properties of classes in the training set by eliminating the differences of the samples in each class. A common vector for each individual class is obtained by removing all the features that are in the range space of the scatter

matrix of its own class and then the obtained common vectors are used for recognition.

To solve the small sample size problem in (1), the DCV method utilizes the idea of common vector. However, instead of using a given class's own scatter matrix, it uses the within-class scatter matrix of all the classes to obtain the common vectors. The major characteristic of the DCV method is that its projection matrix P_{DCV} resides in the null space of the within-class scatter matrix. Consequently P_{DCV} concentrates the samples from the same class to a unique discriminnant common vector and the Fisher's Linear Discriminat criterion defined in (2) achieves a maximum (infinite in fact). In [8], the authors gave two theoretically identical ways for implementing the DCV method, i.e., one by eigen-decomposition and the other by difference subspace and the Gram-Schmidt Orthogonalization Procedure. Due to the latter's efficiency over the former, we introduce DCV implemented by the latter procedure as follows:

Step 1: Calculate the range space of the within-class matrix, which is identical to the range space of the difference subspace H_w. Here, H_w is defined as

$$H_w = [b_1^1, \cdots, b_{N_1-1}^1, b_1^2, \cdots, b_{N_C-1}^1] \tag{3}$$

where

$$b_j^i = x_j^i - x_{N_i}^i,\ i = 1, 2, \cdots, C,\ j = 1, 2, \cdots, N_i - 1 \tag{4}$$

is the j-th difference vector of the i-th class. Apply the Gram-Schmidt orthogonalization procedure to H_w, and get

$$H_w = UV_1 \tag{5}$$

then U is an orthonormal matrix whose column vectors span the range space of the within-class matrix.

Step 2: Choose any sample from each class (typically, the last sample of the i-th class x^i_{Ni}) and project it to the null space of the within-class matrix through the following equation:

$$x_{com}^i = x_j^i - UU^T x_j^i = x_{N_i}^i - UU^T x_{N_i}^i \tag{6}$$

where x^i_{com} is a common vector of the i-th class and is independent of index j.

Step 3: Form the matrix B_{com}, where

$$B_{com} = [b_{com}^1, b_{com}^2, \cdots, b_{com}^{C-1}] \tag{7}$$

and

$$b_{com}^{i} = x_{com}^{i} - x_{com}^{C}, \quad i = 1, 2, \cdots C-1 \tag{8}$$

Apply the Gram-Schmidt orthogonalization procedure to B_{com}, and get

$$B_{com} = P_{DCV} V_2 \tag{9}$$

then P_{DCV} is the projection matrix calculated by DCV.

2.2 Neighbourhood Components Analysis (NCA)

NCA aims at learning a Mahalanobis distance metric which can be denoted as

$$dist(y_i, y_j) = (y_i - y_j)^T AA^T (y_i - y_j) = (A^T y_i - A^T y_j)^T (A^T y_i - A^T y_j) \tag{10}$$

where A is a projection matrix that transforms the data. NCA subtly converts learning the Mahalanobis metric to learning the projection matrix A, which can be clearly observed from (10). NCA looks for the distance metric (or equivalently the projection matrix A) through maximizing the KNN leave-one-out (LOO) performance on the training data. To this end, NCA adopts a differentiable cost function based on stochastic ("soft") neighbor assignments in the transformed space to measure the KNN performance as follows:

Each data sample y_i selects another data sample y_j as its neighbor with some probability p_{ij}, and inherits its class label from the data sample it selects. The probability p_{ij} is defined using a softmax over Euclidean distances in the transformed space (transformed by projection matrix A):

$$p_{ij} = \frac{\exp(-\left\| A^T y_i - A^T y_j \right\|^2)}{\sum_{k \neq i} \exp(-\left\| A^T y_i - A^T y_k \right\|^2)} \quad , p_{ii} = 0 \tag{11}$$

Under the stochastic selection rule, the possibility p_i that data sample y_i will be correctly classified can be computed as:

$$p_i = \sum_{j \in C_i} p_{ij} \tag{12}$$

where

$$C_i = \{ j \mid z_j = z_i \} \tag{13}$$

denotes the set of data samples in the same class as y_i.

NCA then calculates the projection matrix A by maximizing *the expected number of points correctly classified* under the scheme:

$$f(A) = \sum_i p_i = \sum_i \sum_{j \in C_i} p_{ij} \tag{14}$$

Differentiating f with respect to the projection matrix A yields a gradient rule in the following equation:

$$\frac{\partial f}{\partial A} = 2\sum_i (p_i \sum_k p_{ik}(y_i - y_j)(y_i - y_j)^T - \sum_{j \in C_i} p_{ij}(y_i - y_j)(y_i - y_j)^T)A \tag{15}$$

In maximizing the criterion in (14), we can simply employ a gradient based optimizer such as conjugate gradients based on (15). Furthermore, by restricting A to be a nonsquare matrix of $d \times m$ ($m << d$) NCA can also do linear dimensionality reduction such as face recognition [18].

The algorithm for NCA can be explained as follows:

Step 1: Initialize the projection matrix A.

Step 2: Use the conjugate gradients optimizer to optimize (14), where each iteration step can be decomposed into the following four sub-steps:

a) Project the training samples by the projection matrix A to yield $A^T y_i$, for i=1, 2, …, M

b) Calculate the square Euclidean distance among the training samples in the transformed space, i.e., $||A^T y_i - A^T y_j||^2$, for i, j=1, 2, …, M

c) Compute p_{ij} and p_i according to (11) and (12) respectively

d) Calculate (15) and update the projection matrix A by a conjugate gradients optimizer.

Repeat a), b), c) and d) l times for the convergence of (14) and yield the projection matrix $P_{NCA}=A$ for NCA.

2.3 Laplacianfaces (LAP)

The LAP method aims at preserving the local structure of the image space. To this end, a face subspace called Laplacianfaces is obtained by locality preserving projections (LPP) [16], and each face image in the d-dimensional image space is mapped to m($<<d$)-dimensional Laplacianfaces subspace. The objective function for LAP is defined as:

$$\min \quad \sum_{ij} (w^T y_i - w^T y_j)^2 S_{ij} \tag{16}$$

where w is a projection vector, $w^T y_i$ is the one-dimensional representation of y_i and the matrix S is a similarity matrix defined as follows:

$$S_{ij} = \begin{cases} \exp(-\|y_i - y_j\|^2 / t), & \text{if } y_i \text{ is among } k \text{ nearest neighbors of } y_j \\ & \text{or } y_j \text{ is among } k \text{ nearest neighbors of } y_i \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

or

$$S_{ij} = \begin{cases} \exp(-\|y_i - y_j\|^2 / t), & \text{if } y_i \text{ is among } k \text{ nearest neighbors of } y_j \\ & \text{or } y_j \text{ is among } k \text{ nearest neighbors of } y_i, \text{ and } z_i = z_j \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

where t is a suitable value set to the average of $\|y_i-y_j\|^2$, namely

$$t = \frac{1}{M^2} \sum_{ij} \|y_i - y_j\|^2 \quad (19)$$

Equation (17) gives a definition of the similarity matrix S in an unsupervised manner while (18) gives a definition in a supervised manner. Note that according to our personal communications with one of the authors of [17], the LAP method is in fact performed in a supervised manner in their experimental parts, namely, the KNN search is actually restricted to a single class rather than the whole database[1], so in this article, we will focus on the similarity matrix S defined in a supervised way. And consequently, when not specially noted, by LAP in the following discussion, we mean that (18) is used.

By minimizing (16), LAP incurs a heavy penalty if neighboring points (belonging to the same class) y_i and y_j are mapped far apart, i.e., if $(w^Ty_i-w^Ty_j)^2$ is large. Therefore, LAP attempts to assure that, if y_i and y_j are "close", then w^Ty_i and w^Ty_j are close as well. As a result, the local information in the original image space is preserved in the computed LAP.

Minimizing (16) is equivalent to minimizing the following equation:

$$\frac{1}{2} \sum_{ij} (w^T y_i - w^T y_j)^2 S_{ij} = w^T Y (D - S) Y^T w = w^T Y L Y^T w \quad (20)$$

where $Y=[y_1, y_2, \ldots, y_M]$, and D is a diagonal matrix with its entries being the column (or row since S is symmetric) sums of S, $D_{ii}=\Sigma_j S_{ji}$ and $L=D-S$ is the Laplacian matrix. Matrix D provides a natural measure on the data samples, namely the bigger the value D_{ii} (corresponding to y_i) is, the

[1] Besides, he kindly tells us that k should be less than the number of training samples in each class., and t defined in (19) is also under his suggestion.

more "important" is y_i. By imposing a constraint $w^T YDY^T w=1$, the minimization problem reduces to finding:

$$\begin{aligned} &\arg\min_{w} w^T YLY^T w \\ &w^T YDY^T w = 1 \end{aligned} \tag{21}$$

The projection vector w that minimizes (21) can be solved through the generalized eigenvalue problem

$$YLY^T w = \lambda YDY^T w \tag{22}$$

As is described in the beginning of section 2, in such application as face recognition, the dimensionality of the image d is typically larger than the number samples M, i.e., $d>>M$. The rank of YDY^T is at most M, while YDY^T is a $d\times d$ matrix, which implies that YDY^T is singular. To overcome the singularity of YDY^T, LAP employs a procedure similar to the PCA+LDA or the Fisherface method proposed by Belhumeur et al [5], namely applying a PCA projection first. More specifically, LAP operates as follows:

Step 1: **PCA projection**. Project the face images y_i, i=1, 2, …, M to the PCA subspace by keeping the 98 percent information in the sense of reconstruction. For sake of simplicity, y_i is used to denote the images in the PCA subspace in the following steps. And this PCA subspace is denoted as W_{PCA}.

Step 2: **Construct the nearest-neighbor graph in a supervised manner and calculate the similarity matrix *S***. Let G denote a graph with M nodes. The i-th node corresponds to the face image y_i. An edge is put between node i and j, if y_i and y_j satisfies the following two conditions: 1) they belong to the same class, or in other words, $z_i=z_j$; and 2) y_i is among KNN of y_j, or y_j is among KNN of y_i. Based on the constructed nearest neighbor graph, the similarity matrix S can be calculated through (18).

Step 3: **Eigenmap**. Compute the eigenvectors and eigenvalues for the generalized eigenvector problem in (22). Let W_{LPP} be a matrix whose column vectors are the m eigenvectors corresponding to the first m smallest eigenvalues. Then

$$P_{LAP} = W_{PCA} W_{LPP} \tag{23}$$

is the calculated projection matrix by LAP.

3 A comparative study among the three methods

From section 2, we can clearly observe that the three methods are originated from different starting points, namely: a) DCV aims at solve the small sample size problem in LDA, and to this end, it restricts the solution to the null space of the within-class matrix S_w, and gets an optimum (infinite in value) of objective function (1); b) NCA intends to learn a Mahalanobis distance metric or equivalently a projection matrix through maximizing the KNN LOO classification performance on the training set through (15); c) LAP models a manifold structure by a nearest-neighbor graph, aims at preserving the local structure of the image space and optimizes (17). In what follows, we carry out a comparative study among DCV, NCA and LAP. For convenience of comparison, we first give the properties of DCV, NCA and LAP in section 3.1. Then we give comparison between DCV and NCA in section 3.2, and comparison between DCV and LAP in section 3.3. The computational costs in calculating the projection matrices by the three methods are investigated in section 3.4, and storage cost and the computational cost for classifying a given unknown sample are discussed in section 3.5.

3.1 Properties of DCV, NCA and LAP

3.1.1 Properties of DCV

The main characteristic of DCV is that, the projection matrix P_{DCV} resides in the null space of the within-scatter matrix and concentrates the samples from the same class to a unique discriminant common vector. This fact is stated in the theorem 1.

Theorem 1 [8] The projection matrix yielded by DCV concentrates the samples from the same class to a unique common discriminant common vector, namely

$$P_{DCV}^T x_j^i = P_{DCV}^T x_{N_i}^i \tag{24}$$

which is independent of index j.

Considering the fact that $\{x_j^i\}$ and $\{y_i, z_i\}$ are the equivalent descriptions of the training sample set, we can easily draw the following corollary.

Corollary 1 P_{DCV} concentrates the samples from the same class to a unique common discriminant common vector, i.e., if $z_i=z_j$

$$P_{DCV}^T y_i = P_{DCV}^T y_j \tag{25}$$

From Corollary 1, we can easily get the following corollary which shows that P_S, a projection matrix derived from P_{DCV}, also concentrates the samples from the same class to a unique common vector.

Corollary 2 Denote $P_{DCV}=[w_1, w_2, \ldots, w_{C-1}]$, and $S=[s_1, s_2, \ldots, s_{C-1}]$, where s_i is an integer satisfying $1 \le s_i \le C-1$ for $i=1, 2, \ldots C-1$. Let P_S be defined as:

$$P_S = [w_1^S, w_2^S, \cdots, w_{C-1}^S] \tag{26}$$

where

$$w_i^S = w_{s_i}, \quad i = 1, 2, \cdots, C-1 \tag{27}$$

Then when $z_i=z_j$, the following equation holds.

$$P_S^T y_i = P_S^T y_j \tag{28}$$

***Proof*:** From corollary 1, we can easily get that when $z_i=z_j$, $w_k^T y_i = w_k^T y_j$, where w_k is the k-th column vector in P_{DCV}. Since the column vectors in P_S are the corresponding column vectors sampled from P_{DCV}, then naturally if $z_i=z_j$, (28) holds.

Now let us reveal the essence of DCV, namely, computing the projection matrix by DCV is equivalent to solving a thin QR decomposition problem, in the following theorem:

Theorem 2 Let matrix H_b be defined as

$$H_b = [b_1, b_2, \cdots, b_{C-1}] \tag{29}$$

where

$$b_i = x_{N_i}^i - x_{N_C}^C, \quad i = 1, 2, \cdots C-1 \tag{30}$$

Apply a thin QR decomposition to $[H_w \; H_b]$, and have

$$[H_w \quad H_b] = [Q_1 \quad Q_2] \begin{bmatrix} R_1 & L \\ O & R_2 \end{bmatrix} \tag{31}$$

where Q_1 and Q_2 are orthonormal matrices, R_1 and R_2 are upper triangular matrices, O a zero matrix, and L a matrix. Then we have

$$P_{DCV} = Q_2 \tag{32}$$

***Proof*:** Considering the thin QR decomposition in (32), we can easily get

$$H_w = Q_1 R_1 \tag{33}$$

$$H_b = Q_1 L + Q_2 R_2 \tag{34}$$

$$L = Q_1^T H_b \tag{35}$$

$$H_b - Q_1 Q_1^T H_b = Q_2 R_2 \tag{36}$$

Employing (6-8), (29-30), we can get

$$B_{com} = H_b - UU^T H_b \tag{37}$$

Note that Gram-Schmidt orthogonalization procedure is one way to implement the thin QR decomposition [21], and such decomposition is unique if the matrix has full column rank. H_w, B_{com} and $[H_w \ H_b]$ have full column ranks due to the fact that the training samples are linearly independent. Consequently, from (5) and (33), we have

$$U = Q_1 \tag{38}$$

$$V_1 = R_1 \tag{39}$$

Similarly, employing (9), (36) and (37), we have

$$P_{DCV} = Q_2 \tag{40}$$

$$V_2 = R_2 \tag{41}$$

This ends the proof of this theorem.

Favored by this revealed essence, DCV can be understood more clearly than the three steps described in section 2.1 and be easily extended to its nonlinear version utilizing kernel QR decomposition [25]. In addition, again based on this revealed essence, we can verify that when the training samples are linearly independent, the extracted discriminant common vectors for different classes are different in the following theorem.

Theorem 3 When the training samples are linearly independent, the extracted features for training samples from different classes are different, namely,

$$P_{DCV}^T x_{N_i}^i \neq P_{DCV}^T x_{N_j}^j, \ for\ all\ i \neq j \tag{42}$$

***Proof*:** Since Q_1 and Q_2 in (31) are orthogonal, then from (34) and (40), we have

$$P_{DCV}^T H_b = R_2 \tag{43}$$

When the training samples are linearly independent, the matrix $[H_w\ H_b]$ has full column rank, and consequently the diagonal entries of R_2 are all positive. Denote $R_2=[r_1, r_2, \ldots, r_{C\text{-}1}]$, and r_C a $d\times 1$ zero column vector, we can easily verify that

$$r_i - r_j \neq 0, \quad i \neq j, and\ i, j = 1, 2, \cdots, C \tag{44}$$

Employing (29), (30) and (43) together, we have

$$P_{DCV}^T (x_{N_i}^i - x_{N_C}^C) = r_i, i = 1, 2, \cdots C \tag{45}$$

which leads to

$$P_{DCV}^T x_{N_i}^i - P_{DCV}^T x_{N_j}^j = r_i - r_j, \quad i, j = 1, 2, \cdots, C \tag{46}$$

From (44) and (46), it is obvious that (42) holds and this ends the proof of this theorem.

Again recalling that $\{x_j^i\}$ and $\{y_i, z_i\}$ are equivalent description of the training sample set, we can easily draw the following corollary:

Corollary 3 The extracted features for training samples from different classes by P_{DCV} are different, i.e., if $z_i \neq z_j$,

$$P_{DCV}^T y_i \neq P_{DCV}^T y_j \tag{47}$$

3.1.2 Property of NCA

We show that the value of NCA's objective function in (14) is at most M, namely, the number of training samples, in the following theorem.

Theorem 4 The possibility p_i that data sample y_i will be correctly classified is at most 1, and as a result the objective function in (14) is at most M, namely

$$p_i \leq 1 \tag{48}$$

and

$$f(A) \leq M \tag{49}$$

***Proof*:** It is easy to verify that

$$p_i = \sum_{j \in C_i} p_{ij} \le \sum_j p_{ij} = 1 \tag{50}$$

and consequently,

$$f(A) = \sum_i p_i \le \sum_i 1 = M \tag{51}$$

which ends the proof of this theorem.

3.1.3 Property of LAP

We show that the value of the objective function of LAP is at least 0 in the following theorem.

Theorem 5 For any projection matrix W, the objective function (18) or (20) is at least 0, namely

$$trace(W^T YLY^T W) \ge 0 \tag{52}$$

***Proof*:** It is obvious that $(w^T y_i - w^T y_j)^2 \ge 0$, and $S_{ij} \ge 0$, then $(w^T y_i - w^T y_j)^2 S_{ij} \ge 0$. From (20), we have

$$w^T YLY^T w = 1/2 \sum_{ij} (w^T y_i - w^T y_j)^2 S_{ij} \ge 0 \tag{53}$$

Denote $W=[w_1, w_2, \ldots, w_m]$, we have $w_i^T YLY^T w_i \ge 0$, hence trace$(W^T YLY^T W) \ge 0$. And this ends the proof of this theorem.

3.2 DCV versus NCA

We first show that for a sufficiently large positive number β, βP_{DCV} is the optimal result of NCA. We formally verify this argument in theorem 6 as follows:

Theorem 6 When β is a sufficiently large positive number, βP_{DCV} becomes the optimal result of NCA with respect to the objective function (14) or equivalently the following equation holds

$$\lim_{\beta \to \infty} f(\beta P_{DCV}) = M \tag{54}$$

***Proof*:** Let

$$\alpha_1 = \min(\left\| P_{DCV}^T y_i - P_{DCV}^T y_j \right\|^2), i, j = 1, 2, \cdots M \ and \ z_i \ne z_j \tag{55}$$

α_1 is ensured to be positive from (47) in Corollary 3. Furthermore, employing (25) in Corollary 1, we have

$$\exp(-\left\| (\beta P_{DCV})^T y_i - (\beta P_{DCV})^T y_j \right\|^2) = 1, if \ z_i = z_j \tag{56}$$

From (11), (12), (55) and (56), we have

$$(N_i - 1) \le \sum_{k \ne i} \exp(-\left\| (\beta P_{DCV})^T y_i - (\beta P_{DCV})^T y_k \right\|^2) \le (N_i - 1) + (M - N_i)\exp(-\alpha_1 \beta^2) \tag{57}$$

and

$$(N_i - 1)/[(N_i - 1) + (M - N_i)\exp(-\alpha_1 \beta^2)] \le p_i \le (N_i - 1)/(N_i - 1) = 1 \tag{58}$$

Then from (14) and (58), we have

$$\lim_{\beta \to \infty} p_i = 1, \quad i = 1, 2, \cdots, M \tag{59}$$

and

$$\lim_{\beta \to \infty} f(\beta P_{DCV}) = \lim_{\beta \to \infty} \sum_{i=1}^{M} p_i = \sum_{i=1}^{M} \lim_{\beta \to \infty} p_i = M \tag{60}$$

This ends the proof of this theorem.

In fact, when β is only a relatively large number, e.g., $\alpha_1\beta^2 \ge 1000$, or equivalently $\beta \ge \beta_1$, where β_1 is defined as follows:

$$\beta_1 = \sqrt{1000 / \alpha_1} \tag{61}$$

we have $\exp(-\alpha_1\beta^2)=0$ numerically, and consequently, we have $p_i=1$, for $i=1, 2, \ldots, M$, and $f(\beta P_{DCV})=M$ numerically.

Now we will show that there is a category of projection matrices derived from P_{DCV} which will achieve the optimal value of (14) in the following Corollary.

Corollary 4 Let $P_{DCV}=[w_1, w_2, \ldots, w_{C-1}]$, and $S=[s_1, s_2, \ldots, s_{C-1}]$, where s_i is an integer satisfying $1 \le s_i \le C-1$ for $i=1, 2, \ldots C-1$. Let P_S be a matrix defined through (26) and (27), and further satisfies that when $z_i \ne z_j$, $P_S^T y_i \ne P_S^T y_j$, then we have:

$$\lim_{\beta \to \infty} f(\beta P_S) = M \tag{62}$$

***Proof*:** From Corollary 2, we have if $z_i=z_j$ $P_S^T y_i = P_S^T y_j$. Similar to the proof in theorem 6, we can define α_2 to be the minimal distance among the training samples belonging to different classes in the subspace spanned by P_S as:

$$\alpha_2 = \min(\left\| P_S^T y_i - P_S^T y_j \right\|^2), i, j = 1, 2, \cdots M \text{ and } z_i \ne z_j \tag{63}$$

and α_2 is positive due to the fact that when $z_i \ne z_j$, $P_S^T y_i \ne P_S^T y_j$. Following a similar proof procedure, we can easily get (62), and this ends the proof of this Corollary.

Similarly, when β is only a relatively large number, e.g., $\alpha_2\beta^2 \ge 1000$, or equivalently $\beta \ge \beta_2$, where

β_2 is defined as follows:

$$\beta_2 = \sqrt{1000/\alpha_2} \tag{64}$$

we have $\exp(-\alpha_2\beta^2)=0$ numerically, and consequently, we have $f(\beta P_S)=M$ numerically.

Corollary 4 says that if P_S is a projection matrix that satisfies: 1) its column vectors are sampled randomly from those column vectors of P_{DCV} and 2) when $z_i \neq z_j$, $P_S^T y_i \neq P_S^T y_j$, then βP_S is the optimal result of NCA considering (14). For simplicity, in the following discussion, we give a special P_S defined as follows: the first $C/2$ column vectors are the corresponding first $C/2$ column vectors of P_{DCV}, and the last $C/2$-1 column vectors are also composed of the first $C/2$-1 column vectors of P_{DCV}, namely

$$\begin{aligned} P_S(:,1:(C/2)) &= P_{DCV}(:,1:(C/2)) \\ P_S(:,(C/2+1):(C-1)) &= P_{DCV}(:,1:(C/2-1)) \end{aligned} \tag{65}$$

It is obvious that, compared to P_{DCV}, P_S has a loss in the discriminant information to some degree. However, βP_S is also the optimal solution to (14), which will be verified in the experiment parts.

Furthermore, NCA employs a conjugate gradient optimizer to optimize (14), and as a result, P_{NCA} can only obtain a local minimum in the sense of (14), which will again be shown in the experimental parts.

Finally, from the above analyses, we can draw some conclusions as follows: 1) the projection matrix yielded by DCV can become an optimal projection matrix for NCA under the objective function (14); 2) from Corollary 4, there is a category of matrices derived from P_{DCV} that can yield the optimal result in sense of (14) and meanwhile losses certain discriminant information to some extend, so study should be carried on NCA to tackle this problem; and 3) due to the possibility of trapping into the local minima in the optimizing procedure, NCA maybe never gets its optimized projection matrix in the sense of (14).

3.3 DCV versus LAP

Like the description in section 3.2, we first show that P_{DCV} is optimal in the sense of LAP's objective function (16) in the following theorem:

Theorem 7 P_{DCV} is the optimal solution to the objective function (16) of LAP, namely:

$$trace(P_{DCV}^T Y L Y^T P_{DCV}) = 0 \tag{66}$$

***Proof*:** For any given pair of samples y_i and y_j, we have the following two facts: 1) if their corresponding class labels meets $z_i \neq z_j$, then according to the definition of the similarity matrix S in (18), we have $S_{ij}=0$. Thus, in this case, we have $(w_k^T y_i - w_k^T y_j)^2 S_{ij}=0$, where w_k is the k-th column vector of $P_{DCV}=[w_1, w_2, \ldots, w_{C-1}]$; 2) if their corresponding class labels meets $z_i=z_j$, then we have $w_k^T y_i = w_k^T y_j$ from Corollary 1. Thus in this case, we still have $(w_k^T y_i - w_k^T y_j)^2 S_{ij}=0$. In summary, for any pair of y_i and y_j and all k, we always have $(w_k^T y_i - w_k^T y_j)^2 S_{ij}=0$. Furthermore, combining (20), we get the following equation

$$trace(P_{DCV}^T Y L Y^T P_{DCV}) = 1/2 \sum_{ijk} (w_k^T y_i - w_k^T y_j)^2 S_{ij} = 0 \tag{67}$$

Recalling that we have presented in theorem 5 that trace$(W^T Y L Y^T W) \geq 0$, then P_{DCV} is the optimal result of LAP when the objective function (16) is considered. This ends the proof of this theorem.

In the LAP method, the authors employed the PCA projection as the first step, and kept 98 percent information in the sense of reconstruction error, then solved the generalized eigenvalue problem in (22) in the projected PCA subspace. However, doing so may lead to such a shortcoming that some directions corresponding to the small eigenvalues are thrown away in the PCA step, which has a potential to remove directions that contain discriminative information. Furthermore, we will show in the experimental parts that trace$(P_{LAP}^T Y L Y^T P_{LAP})$ is generally positive, which means that the projection matrix P_{LAP} yielded by LAP is generally not optimal in the sense of (16).

In section 2.3, we describe the definition of the similarity matrix S in two ways, namely the unsupervised and supervised. Our discussion on the LAP method in this article refers to the supervised manner. Now, we will also make remarks on the Laplacianfaces method in the unsupervised form, and denote it as ULAP with the obtained corresponding projection matrix as P_{ULAP}.

Assuming that in the ULAP method, the similarity matrix S has been calculated through (17), and accordingly, the matrix L and D are calculated respectively. As described in section 2.3, YLY^T is a $d \times d$ matrix with a rank of at most M, then its null space has a rank over d-M. Enlightened by the idea of DCV, the optimal solution to the ULAP method in terms of (16) or equivalently (20) should reside in the null space of YLY^T (Notice: when the projection vectors reside in this null space, both (16) and (20) get the optimal value 0). However, we have the following two facts: 1)

For a given sample y_i, the obtained projection matrix will concentrate the samples in its neighborhood to a common vector; and 2) The k nearest neighbors of a given sample y_i inevitably include the samples from the different class from y_i. These two facts tell us that the extracted features are not good for classification, since the training samples from different classes may be concentrated to a common vector. Furthermore, considering the extreme condition where k is set to M-1 in (17), it is obvious that the obtained optimal projection vectors definitively reside in the null space of the total scatter matrix. Consequently, the extracted features for the training samples become a unique common vector, and contain no discriminative information for classification.

Although in the ULAP method, a PCA stage is applied prior to the optimization procedure, its objective function is in fact (16) or equivalently (20). According to our discussion in the above paragraph, the extracted features of ULAP are not good for classification, which will further be shown in our experiment parts.

3.4 Computational cost in calculating the projection matrices

We follow Golub and Van Loan [21] in definition of floating point operation (flop) count, counting a scalar addition, multiplication, or division as one flop. From theorem 2, we see that calculating P_{DCV} is in fact a thin QR decomposition to $[H_w\ H_b]$, and as a result, consuming $2d(M\text{-}1)^2$ flops utilizing the modified Gram-Schmidt algorithm [21].

We calculate the flops that needed by NCA as follows. In each iteration step, NCA consumes $2dmM$ flops in a), $3mM^2$ flops in b), $2M^2$ flops in c) and $dmM+dmM^2$ flops in d). Then NCA consumes about $3l(dmM+dmM^2)$ flops, where l stands for the total number of iteration steps.

As for the LAP method, it first applies a PCA stage which keeps 98 percent information in the sense of reconstruction error. Let the number of kept eigenvectors in the PCA stage be g, where $g<M$-1. PCA converts calculating the leading g eigenvectors corresponding to the $d\times d$ total scatter matrix $S_t=BB^T$ to solving the eigen-decomposition problem of the $M\times M$ matrix B^TB [1]. Employing the symmetric QR decomposition algorithm [21] for the eigen-decomposition problem, the PCA stage consumes about $9M^3+2dM^2+2dMg$ flops. The LAP method then solves the generalized eigenvalue problem in (22), and costs another $14g^3$ flops utilizing both Cholesky decomposition and symmetric QR decomposition [21]. So LAP consumes about $9M^3+2dM^2+2dMg+14g^3$ flops in total.

For better comparison among their computation costs, we list them in Table 1, from which we can observe that DCV is the most efficient, followed by LAP and NCA.

Table 1 Flops needed to calculate the projection matrix

DCV	NCA	LAP
$2d(M\text{-}1)^2$	$3l(dmM+dmM^2)$	$9M^3+2dM^2+2dMg+14g^3$

3.5 Storage cost

We first analyze the storage complexity in the process of computation for these three methods as follows: 1) From Theorem 2, we get that calculating the projection matrix by DCV is equivalent to solving a thin QR decomposition to matrix $[H_w\ H_b]$ which is of size d by M-1. Thus the space complexity for DCV in process of computation is $O(dM)$. 2) For LAP, it first applies a PCA stage whose space complexity is $O(dM)$ [27], and then LAP manipulates on matrices whose sizes are less than d by M for calculating its projection matrix. Thus the space complexity for LAP is also $O(dM)$. 3) In NCA's Step1, it needs to store A, which is of size d by m; in Step 2(a), it takes a space complexity of O(dm) to calculate $A^T y_i$; in Step 2(b-d), NCA manipulates on matrices whose sizes are less than d by m (keeping in mind that $m<<d$ and $M<<d$). Thus the space complexity for NCA is O(dm) in process of calculating the projection matrix. In summary, NCA has the least space complexity in process of computation, followed by LAP and DCV

Let the three methods all keep C-1 projection vectors. In the DCV method, P_{DCV} concentrates the training samples from the same class to a unique common vector. Then it needs to store $(C\text{-}1)C$ elements for the extracted features of all the training samples, while both NCA and LAP need to store $(C\text{-}1)M$ elements. As a result, DCV consumes much less memory than both NCA and LAP.

When used to classifying an unknown sample, DCV only needs to compare with the C (C-1)-dimensional features rather than the M (C-1)-dimensional features by both NCA and LAP. Consequently, DCV is computationally more efficient during classification than both NCA and LAP.

4 Experiments

In section 3, we have presented our theoretical arguments. And now, we carry out experiments on three face datasets: ORL, YALE and AR in order to: 1) verify experimentally that the

projection matrix of DCV provides the optimal solution to both NCA and LAP in terms of (14) and (16) respectively; 2) investigate the values of objective functions (14) and (16) by the projection matrices of NCA and LAP respectively; 3) report the classification accuracies of DCV, NCA and LAP, and give the application scope of DCV from the analysis of the results. In what follows, we will give dataset description and experiment setting in section 4.1, detail 1) and 2) in section 4.2, and 3) in section 4.3.

4.1 Dataset description and experiment setting

The ORL face dataset contains 400 images of 40 persons, where each person has 10 gray level images with a resolution 112×92. Fig.1 shows ten images of the first person in this dataset. All the images were taken against a dark homogeneous background with the subjects in an upright frontal position, and with tolerance for some tilting and rotation of up to about 20°. The images for each person have variations in facial expression (open/closed eyes, smiling/ non-smiling), and facial details (glasses/no glasses), and there is some variation in scale up to about 10%. We resize the images to a resolution of 56×46 for computation convenience, and rescale the gray level values of all images to [0 1].

(1) (2) (3) (4) (5)

(6) (7) (8) (9) (10)

Fig. 1Ten images from one person in the ORL face database

The YALE face dataset contains 165 grayscale face images of 15 persons, with each one having 11 images. The images are cropped into 50×50, and the gray level values of all images are rescaled to [0 1]. The challenge of this dataset is expression and illumination. Fig. 2 shows the eleven images of one person from this dataset.

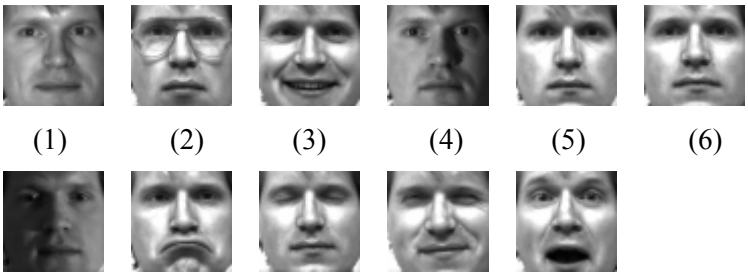

Fig. 2 Eleven images from one person in the YALE face database

The AR [26] face dataset consists of over 3200 images of frontal images of faces of 126 subjects. Each subject has 26 different images which were grabbed in two different sessions separated by two weeks, 13 images in each session were recorded. For the 13 images, the first one is of neutral expression, the second to the fourth are of smile, anger and scream expression, the others are either light or scarf variation. In our experiments here, we use the 1400 gray level images from 100 objects, where each object has 14 images. More specifically, these 14 images correspond to (a)-(g) and (n)-(t), as are illustrated in Fig. 3. The 1400 images are preprocessed by Martinez [26] with a resolution of 165×120. Here, for computational convenience, we resize them to 66×48 and the gray level values are rescaled to [0 1].

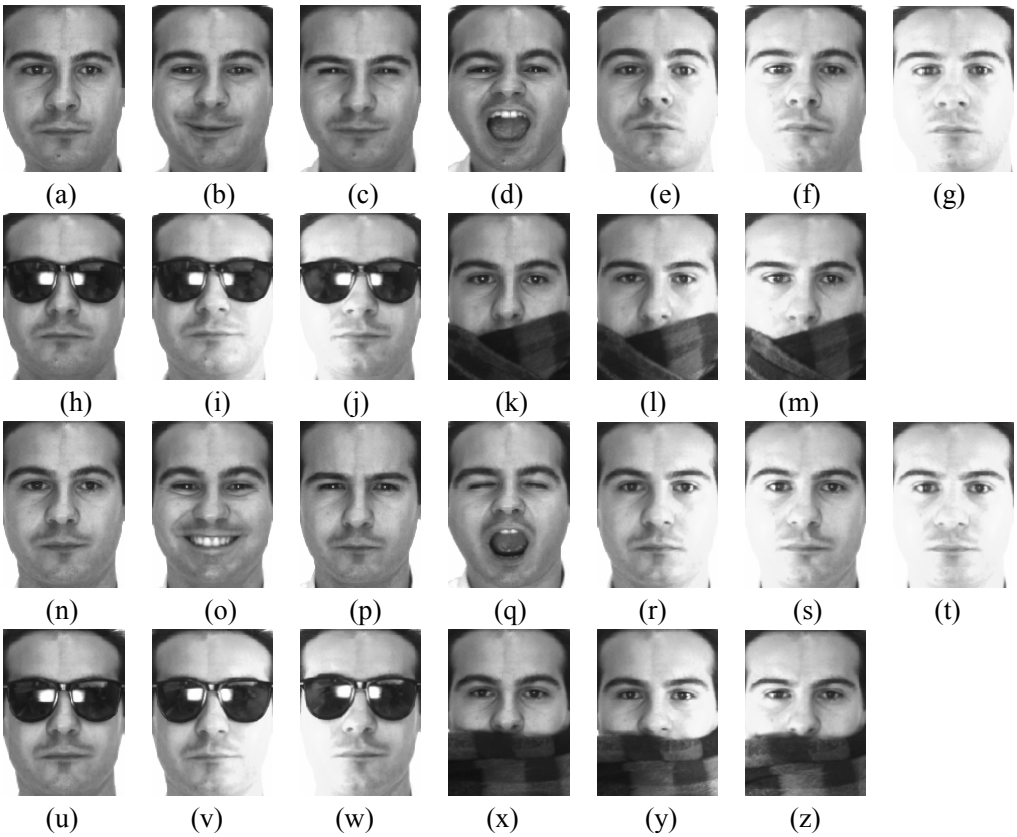

Fig. 3 Images from one person in the AR face dataset

After describing the three data set used in our experiments, it is worthwhile to make some remarks on the experiment settings as follows:

1) We perform experiments on these three dataset in two manners: 1) in a deterministic manner, more specifically, the training samples and testing samples are organized as depicted in Table 2, 3 and 4 respectively; 2) in a random manner, i.e., in each run, randomly selecting r (5 for ORL, 6 for YALE and 7 for AR) samples of each person for training and the rest for testing, and such experiments are independently repeated 20 times. We carry out the deterministic experiments based on the following two considerations: 1) they can be reproducible; and 2) they enable us to have a good look at the performance of specific partition. Meanwhile, the random experiments enable us to analyze the classification performance from the viewpoint of statistics.

Table 2 Deterministic partition of training samples and testing samples on ORL face dataset

	ORL1	ORL2	ORL3	ORL4
Training set	1, 2, 3, 4, 5	6, 7, 8, 9, 10	1, 3, 5, 7, 9	2, 4, 6, 8, 10
Testing set	6, 7, 8, 9, 10	1, 2, 3, 4, 5	2, 4, 6, 8, 10	1, 3, 5, 7, 9

Table 3 Deterministic partition of training samples and testing samples on YALE face dataset

	YALE1	YALE2	YALE3	YALE4
Training set	1, 2, 3, 4, 5, 6	6, 7, 8, 9, 10, 11	1, 3, 5, 7, 9, 11	2, 4, 6, 8, 10, 11
Testing set	7, 8, 9, 10, 11	1, 2, 3, 4, 5	2, 4, 6, 8, 10	1, 3, 5, 7, 9

Table 4 Deterministic partition of training samples and testing samples on AR face dataset

	AR1	AR2	AR3	AR4
Training set	a, b, c, d, e, f, g	n, o, p, q, r, s, t	a, b, c, d, n, o, p, q	a, e, f, g, n, r, s, t
Testing set	n, o, p, q, r, s, t	a, b, c, d, e, f, g	e, f, g, r, s, t	b, c, d, o, p, q

2) The number of projection vectors in each dimensionality reduction method is set to C-1 in all our experiments;

3) When performing experiments on NCA, we try four initializations for the projection matrices: 1) P_{DCV}; 2) P_S, which is defined in (64); 3) P_{PCA}; 4) P_{LDA}, which is the projection matrix by PCA+LDA method [4-6]. Accordingly, the obtained projection matrices (through optimizing (14)) are denoted as A_1, A_2, A_3 and A_4 respectively.

5) In LAP and ULAP, the value of k in (17) and (18) is set to the number of training samples in each class subtracted by 1.

Table 5 Values of α_1, α_2, β_1, β_2, and those of the corresponding objective function (14) by P_{DCV}, P_S, $\beta_1 P_{DCV}$ and $\beta_2 P_S$.

	P_{DCV}				P_S			
	v_1	α_1	β_1	v_2	v_3	α_2	β_2	v_4
ORL1	199.98	7.001	11.951	200	197.69	3.0715	18.044	200
ORL2	199.96	6.4200	12.480	200	194.92	1.7008	24.248	200
ORL3	199.97	7.0760	11.888	200	195.72	2.0082	22.315	200
ORL4	199.94	6.0884	12.816	200	195.60	2.4045	20.393	200
YALE1	89.897	6.6540	12.259	90	83.350	1.8923	22.988	90
YALE2	89.663	5.2458	13.807	90	82.297	0.69782	37.855	90
YALE3	89.465	4.4365	15.013	90	77.583	0.58613	41.305	90
YALE4	89.824	5.5749	13.393	90	84.591	2.1525	21.554	90
AR1	487.57	1.9022	22.928	700	409.99	0.65064	39.204	700
AR2	480.68	1.8758	23.089	700	408.19	0.50193	44.635	700
AR3	366.24	1.2815	27.934	700	389.18	0.42745	48.368	700
AR4	258.48	1.5978	25.017	700	308.21	0.41230	49.248	700

4.2 P_{DCV} as an optimal solution to the objective function of both NCA and LAP

We first verify our argument that βP_{DCV} (β is a relatively large number) is the optimal solution to the objective function of NCA and report experimental results in Table 5. From this table, we can observe that $v_1=f(P_{DCV})<M$, namely, P_{DCV} can not directly attain a optimum of (14). However, the corresponding α_1s defined through (55) are all positive, and we can calculate β_1s according to (61). From $v_2=f(\beta_1 P_{DCV})$ reported in Table 5, we can easily get that $v_2=M$, and as a result our argument mentioned in the beginning of this paragraph is experimentally proven. In addition, from the same table, we can see that $v_3=f(P_S)<M$ while $v_4=f(\beta_2 P_S)=M$, and this verifies Corollary 4 experimentally, namely, there is a category of projection matrices derived from P_{DCV} which will achieve the optimal value of (14). Note that, as described in section 4.1, the gray level values are rescaled to [0 1] in all the experimental data. However, if such gray level values are in the range of [0 255], both $f(P_{DCV})$ and $f(P_S)$ will naturally equal M, since the calculated β_1 and β_2 in Table 5 are all less than 255. In NCA, we try four initialization matrices P_{DCV}, P_S, P_{PCA} and P_{LDA}, and obtain corresponding projection matrices A_1, A_2, A_3 and A_4 by a conjugate gradient optimizer. For better understanding of such optimizing procedure, we plot the values of the objective function (14) during each iteration step on ORL1 in Fig. 4, from which we can see that: 1) the initializations using P_{DCV}, P_S, P_{PCA} and P_{LDA} respectively generally lead to local optima of (14), not the global

optima; 2) the initialization with P_{PCA} yields lower objective function value than P_{DCV}; P_S and P_{LDA}. Meanwhile, the values of (14) yielded respectively by A_1, A_2, A_3 and A_4 on ORL1-ORL4 and YALE1-YALE4 reported in Table 6 confirm such conclusions.

Table 6 Values of the objective function (14) by A_1, A_2, A_3 and A_4

Dataset	A_1	A_2	A_3	A_4
ORL1	199.99	199.98	195.99	199.98
ORL2	199.98	199.98	194.00	199.98
ORL3	199.98	199.98	193.00	199.98
ORL4	199.98	199.98	190.65	199.98
YALE1	89.988	89.984	66.995	89.989
YALE2	89.985	88.417	72.999	89.991
YALE3	89.985	89.377	62.997	89.996
YALE4	89.986	89.990	72.000	89.692

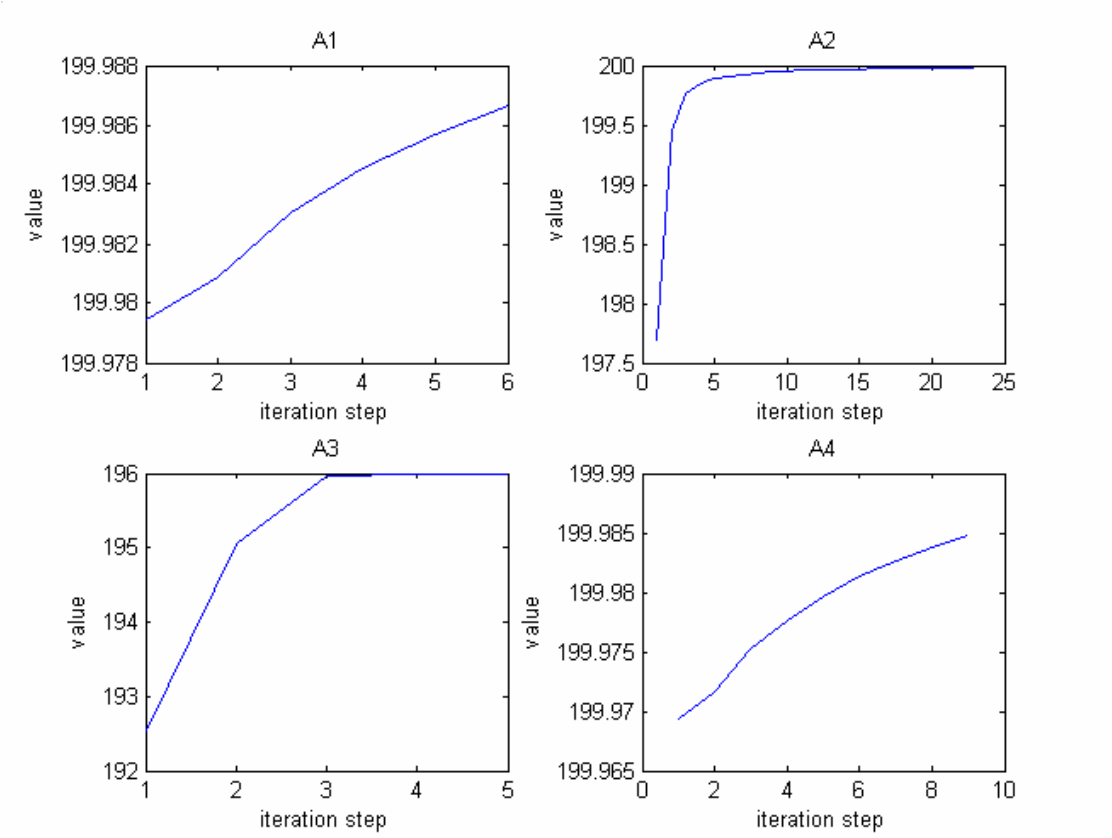

Fig. 4 Illustration of NCA to obtain A_1, A_2, A_3 and A_4 with corresponding initializations on ORL1.

Secondly, we verify our argument that P_{DCV} is the optimal solution to (16). Our experiments on ORL, YALE and AR show that trace($P_{DCV}{}^T YLY^T P_{DCV}$) is less than 1e-10 and consequently can be

considered to be zero numerically, then we can say that P_{DCV} is the optimal solution to (16). Now, we turn to the values of objective function (16) by LAP and ULAP, or equivalently the values of trace($P_{LAP}{}^{T}YLY^{T}P_{LAP}$) and trace($P_{ULAP}{}^{T}YLY^{T}P_{ULAP}$), and report them in Table 7, from which we can obviously observe that neither P_{LAP} nor P_{ULAP} is the optimal solution to the objective function (16) since its optimal value is 0. In addition, the values of trace($P_{ULAP}{}^{T}YLY^{T}P_{ULAP}$) are greatly larger than those of trace($P_{LAP}{}^{T}YLY^{T}P_{LAP}$), which signifies that ULAP is inferior to LAP in classification performance. We will verify this in the following report of classification performance.

Table 7 Values of objective function (16) by P_{LAP} and P_{ULAP}

	ORL1	ORL2	ORL3	ORL4
P_{LAP}	183.34	183.53	209.17	209.14
P_{ULAP}	1145.8	1226.7	1316.5	1321.6
	YALE1	YALE2	YALE3	YALE4
P_{LAP}	92.682	114.15	119.16	79.236
P_{ULAP}	1458.6	1093.0	1463.4	1141.5
	AR1	AR2	AR3	AR4
P_{LAP}	5639.3	5553.8	5628.2	7512.7
P_{ULAP}	25367	25006	20677	39240

4.3 Classification performance

Now, we report the classification accuracies of DCV, NCA, and LAP on the three datasets. The classification accuracies reported here follow the subsequent two procedures: 1) we first utilize these methods respectively to extract the (C-1)-dimensional features, and 2) a nearest neighbor classifier with Euclidean distance is employed for classification based on the extracted features.

Firstly, let us look at the classification performance of the NCA method reported in Table 8, from which we can see that the initial transformation matrix plays an important role in the classification performance of NCA. More specifically, although the initialization with both P_S and P_{LDA} lead to comparable objective function values to P_{DCV}, the classification performances by $A_2$and A_4 are generally inferior to those by A_1. Thus, one important problem in NCA is to choose the proper initialization matrix. In addition, we have reported in section 4.2 that $\beta_2 P_S$ can achieve the optimal value in the sense of objective function (14), and we have also discussed in section 3.2 that compared to P_{DCV}, there is a loss of discriminative information in P_S. Our experimental results reported in Table 8 show that P_S works significantly more poorly than P_{DCV}. So the other

important problem in NCA is to cope with such fact that there exist certain projection matrices which yield the optimum of (14) and meanwhile inherently have a loss in discriminant information compared to the projection matrix of DCV. (Note that, for NCA, we report the experiments on ORL and YALE in a deterministic manner, and the rest experiments are omitted due to the following two reasons: 1) NCA is time consuming for relative large dataset; and 2) the classification performance of NCA can be read from that of DCV, since the projection matrix (multiplied by a factor) of the latter is the optimal solution to the former)

Table 8 Classification accuracies (%) on three datasets in deterministic partition

Dataset	DCV	NCA				P_S	LAP	ULAP
		A_1	A_2	A_3	A_4			
ORL1	91.5	91.5	86.5	90.0	87.5	84.0	85.0	72.5
ORL2	92.5	93	92.0	89.5	92.0	91.0	91.0	77.0
ORL3	97.5	97.5	92.0	95.5	92.0	92.0	91.0	82.0
ORL4	98.0	98.0	97.0	93.0	94.0	96.0	93.5	85.5
avg	94.9	95.0	91.9	92	91.4	90.8	90.1	79.3
YALE1	89.3	89.3	77.3	72.0	81.3	69.3	82.7	74.7
YALE2	81.3	82.7	62.7	70.7	84.0	76.0	82.7	72.0
YALE3	92.0	93.3	65.3	78.7	86.7	76.0	92.0	76.0
YALE4	86.7	85.3	72.0	64.0	77.3	66.7	85.3	64.0
avg	87.3	87.7	69.3	71.3	82.3	72.0	85.7	71.7
AR1	83.6	/	/	/	/	69.9	85.1	67.4
AR2	83.3	/	/	/	/	72.3	84.7	68.7
AR3	84.8	/	/	/	/	51.2	77.2	72.0
AR4	78.3	/	/	/	/	63.8	79.3	67.2
avg	82.5	/	/	/	/	64.3	81.6	68.8

/: the corresponding experiments are not conducted

avg: the average classification accuracy of the corresponding four independent experiments

Secondly, we look at the classification performances of LAP and ULAP. Our experiments on the three dataset in deterministic partitions consistently show that LAP yields significantly better results than ULAP. Such experimental results are in accord with: 1) that trace($P_{ULAP}{}^T YLY^T P_{ULAP}$) are greatly larger than those of trace($P_{LAP}{}^T YLY^T P_{LAP}$), as is revealed in section 4.2; and 2) what we argue in section 3.3, namely in the SSS problem, ULAP is not good for classification.

Thirdly, we make a comparison of all the methods present to draw the following conclusions:

1) DCV achieves comparable classification accuracies to NCA initialized with P_{DCV}, and

meanwhile significantly higher classification accuracies than NCA with other initialization matrices, which can clearly be observed from Table 8. This result is in accord with the fact that: 1) βP_{DCV} is the optimal solution to NCA; and 2) the initialization matrix plays an important role in NCA.

2) DCV achieves higher classification accuracies than the LAP method on ORL1-ORL4, YALE1, YALE3, AR3, and ORL and YALE in a random manner, whereas comparable or even inferior classification accuracies to the LAP method on the rest, as can be observed from Table 8 and 9. We attribute this phenomenon to: 1) the projection matrix of DCV is the optimal solution to the objective function of LAP; 2) DCV works well in the case that the samples belonging to the same class have relatively small variance while poorly when they have relatively large variance. The first has already been verified, and we will detail the second in the following.

Table 9 Average classification accuracies (%) on three datasets in 20 runs

	DCV		LAP	
	CA	STD	CA	STD
ORL	**96.2**	**0.016812**	91.4	0.023736
YALE	**83.0**	0.10900	81.9	**0.10020**
AR	94.0	**0.06135**	**94.2**	0.064372

CA: classification accuracy

STD: the standard derivation of the classification accuracies in 20 runs

Table 10 Values of *MSV* respectively on ORL, YALE and AR in a deterministic manner

dataset	ORL1	ORL2	ORL3	ORL4
MSV	0.0837	0.0852	0.0881	0.0888
dataset	YALE1	YALE2	YALE3	YALE4
MSV	0.1207	0.1187	0.1305	0.1150
dataset	AR1	AR2	AR3	AR4
MSV	0.1568	0.1564	0.1032	0.1690

In order to partially explain the reason why DCV works better on some datasets, and meanwhile yields comparable or even inferior classification accuracies on the rest, we define the mean standard variance (*MSV*) as:

$$MSV = \frac{1}{C} SV_i \tag{68}$$

where SV_i is the standard variance of the i-th class defined as:

$$SV_i = \frac{1}{d}\sum_{k=1}^{d}\sqrt{\frac{1}{N_i - 1}\sum_{j=1}^{N_i}(x_{jk}^i - m_{ik})^2} \tag{69}$$

where x_{jk}^i denotes the k-th element of the d-dimensional sample x_j^i, and similarly, m_{ik} denotes the k-th element of the mean sample of the i-th class m_i:

$$m_i = \frac{1}{N_i}\sum_{j=1}^{N_i} x_j^i \tag{70}$$

We report the values of *MSV* on ORL, YALE and AR data in a deterministic manner in Table 10, from which we can see that when DCV achieves significantly better classification performance than LAP, the value of *MSV* is relatively small (e.g., on ORL1-ORL4, AR3, the values of *MSV* are all below 0.11), while on the contrary, when the value of *MSV* is relatively large (e.g., on AR1, AR2 and AR4, the values of *MSV* are all above 0.15), DCV yields inferior classification accuracy compared to LAP. However, this is the preliminary experiments, and is worthy of further study.

5 Conclusion and future work

In this paper, we have a comparative study among DCV, NCA and LAP to find that in SSS problem, the projection matrix of DCV is the optimal solution to both NCA and LAP in case of their respective objective functions, whereas neither NCA nor LAP may achieve their optimal objective function value. Both theoretical analysis and experimental simulations are presented to verify our arguments. In addition, we show that DCV is much more efficient than both NCA and LAP in both calculating the projection matrix and the classification of a given unknown sample, and reveal the essence of DCV, i.e., calculating the projection matrix is equivalent to solving a thin QR decomposition problem. The revealed essence makes DCV easier to be understood and to be extended to its nonlinear version through kernel QR decomposition [25]. Finally, we experimentally show that DCV is not definitively superior to LAP although the former achieves the optimal solution to the latter, and give possible explanations, namely, when the mean standard variance (*MSV*) is relatively small, DCV works significantly better than the other methods whereas when *MSV* is relatively high, DCV works poorly.

In our point of views, future study should be carried out in the following aspects:

1) Clarify the relationships among different methods, which brings convenience to practitioner

in selecting specific method from a large number of similar methods present.

2) NCA is a very good method, since it relates the feature extraction procedure with the classification performance. However, in the SSS problem, the following question should be tackled: 1) how to choose the proper initialization matrix; and 2) how to deal with such case that certain projection matrices yield the optima of its objective function whereas it is obvious that there is a loss of information.
3) DCV is a good method for solving SSS problem in (1), and achieves the optimum of the criterion (1), (14) and (16). However, as revealed in the experimental parts, it does not definitively yield significantly better classification performance over LAP. Our preliminary result shows that DCV works well when *MSV* is relatively small and poorly when *MSV* is relatively high. Thus study should be carried out to improve the classification performance of DCV when *MSV* is relatively high.
4) Based on the revealed essence, the DCV method can be easily extended to its nonlinear version through kernel QR decomposition [25]. Furthermore, by using specific kernel (such positive kernel as Gaussian kernel), the kernelized DCV can be applied to non-SSS problem, while the original DCV can not.

Acknowledgement

Thank Xiaofei He for valuable discussion, and Jacob Goldberger for supplying the NCA code. And we thank Natural Science Foundation of China under Grant Nos. 60473035 for support.

Reference

[1] M. A. Turk, and A. P. Pentland, Face Recognition Using Eigenfaces, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.

[2] H. Murase and S.K. Nayar, Visual Learning and Recognition of 3-D Objects from Appearance, Int'l J. Computer Vision, vol. 14, pp. 5-24, 1995.

[3] A. M. Martinez and A.C. Kak, PCA versus LDA, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no.2, pp. 228-233, 2001.

[4] D. L. Swets and J. Weng, Using discriminant eigenfeatures for image retrieval, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pp. 831-836, August 1996.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp.711-720, 1997.

[6] C. Liu, and H. Wechsler, Robust Coding Schemes for Indexing and Retrieval from Large Face Databases, IEEE Transactions on Image Processing, vol.9, no.1, pp. 132-136, 2000.

[7] H. Cevikalp and M. Wilke, Face recognition by using discriminative common vectors, Proceedings of 17th International Conference on Pattern Recognition, vol. 1, pp.326-329, August 2004.

[8] H. Cevikalp, M. Neamtu, M. Wilkes, A. Barkana, Discriminative common vectors for face recognition, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 27, no. 1, pp. 4-13, January 2005.

[9] J. Liu and S. Chen, Equivalences among different Null Space Based feature extraction Methods for Small Sample Size Problem, Submitted

[10] L. F. Chen, H. Y. M. Liao, M. T. Ko, J-C Lin and G. J. Yu, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognition, vol. 33, pp. 1713-1726, 2000.

[11] J. Yang, D. Zhang and J. Y Yang, A generalised K-L expansion method which can deal with small sample size and high-dimensional problems, Pattern Analysis & Applications, vol. 6, pp. 47-54, April 2003.

[12] R. Huang, Q. Liu, H. Lu, and S. Ma, Solving the small size problem of LDA, Proceedings of 16th International Conference on Pattern Recognition, vol. 3, pp. 29-32, August 2002.

[13] Y. Chang, C. Hu, M. Turk, "Manifold of Facial Expression," *Pro. IEEE Int'l Workshop Analysis and Modeling of Faces and Gestures*, Oct. 2003.

[14] S.T. Roweis and L.K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embeddings, Science, vol. 290, 2000.

[15] M. Belkin and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, *Proc. Conf. Advances in Neural Information Processing Systems*, 2001.

[16] X. He and P. Niyogi, Locality preserving projections, *Proc. Conf. Advances in Neural Information Processing Systems*, 2003.

[17] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using Laplacianfaces, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, 2005.

[18] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, Neighbourhood Component Analysis, *Proc. Conf. Advances in Neural Information Processing Systems*, 2004.

[19] M. B. Gulmezoglu, V. Dzhafarov, M. Keskin, and A. Barkana, A novel approach to isolated word recognition, IEEE Transactions on Speech and Audio Processing, vol. 7, no. 6, November 1999.
[20] M. B. Gulmezoglu, V. Dzhafarov, and A. Barkana, The common vector approach and its relation to principal component analysis, IEEE Transactions on Speech and Audio Processing, vol. 9, no. 6, September 2001.
[21] G. H. Golub, C. F. V. Loan, Matrix Computations, 3rd Edition, The Johns Hopkins University Press, Baltimore, MD, USA, 1996.
[22] B.E. Boser, I.M. Guyon, and V.N. Vapnik, "A training algorithm for optimal margin classifiers," In D. Haussler, editor, 5th annual ACM Workshop on COLT, pp. 144-152, 1992.
[23] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.
[24] C. Cortes, V.N. Vapnik, Support-vector networks, Machine Learning, vol. 20, no. 3, 273–297. 1995.
[25] S.T. John, C. Nello, Kernel methods for pattern analysis, Cambridge University Press, 2004.
[26] A.M. Martinez and R. Benavente, "The AR Face Database," CVC Technical Report #24, June 1998.
[27] J., Ye, Q., Li, A two-stage Linear Discriminant Analysis via QR-Decomposition, IEEE Trans. Pattern Analysis and Machine Intelligence, 27(6): 929-941, 2005.

Table captions:

Figure captions: