Computing and Informatics, Vol. 35, 2016, 1231–1248

INCREMENTAL AND DECREMENTAL NONPARAMETRIC DISCRIMINANT ANALYSIS FOR FACE RECOGNITION

Nitin Kumar

Department of Computer Science and Engineering
National Institute of Technology Uttarakhand, India
e-mail: nitin2689@gmail.com

Ramesh Kumar Agrawal

School of Computer Science and Systems Sciences
Jawaharlal Nehru University New Delhi, India
e-mail: rkajnu@gmail.com

Ajay Jaiswal

S. S. College of Business Studies
University of Delhi New Delhi, India
e-mail: a_ajayjaiswal@yahoo.com

Abstract. Nonparametric Discriminant Analysis (NDA) possesses inherent advantages over Linear Discriminant Analysis (LDA) such as capturing the boundary structure of samples and avoiding matrix inversion. In this paper, we present a novel method for constructing an updated Nonparametric Discriminant Analysis (NDA) model for face recognition. The proposed method is applicable to scenarios where bursts of data samples are added to the existing model in random chunks. Also, the samples which degrade the performance of the model need to be removed. For both of these problems, we propose incremental NDA (INDA) and decremental NDA (DNDA) respectively. Experimental results on four publicly available datasets viz. AR, PIE, ORL and Yale show the efficacy of the proposed method. Also, the proposed method requires less computation time in comparison to batch NDA which makes it suitable for real time applications.

Keywords: Small sample size, linear discriminant analysis, nonparametric discriminant analysis, scatter matrix, face recognition

Mathematics Subject Classification 2010: 62-H30

1 INTRODUCTION

Automatic face recognition has received significant attention from pattern recognition and machine learning research community in the past few decades. It has diverse applications [1] such as access control, surveillance, ATM, corpse identification, etc. It has achieved good success in controlled environment, but still it faces certain challenges such as illumination changes, variation in pose, expression variation, occlusion, etc. [2, 3] which degrade its performance considerably. Several methods have been proposed in literature to overcome this problem [4]. Appearance based methods for face recognition [5, 6] were widely used for feature extraction in the past. In these methods, the face image of a person with size $l \times w$ pixels is represented as a vector in n-dimensional space where $n = l \times w$. However, working in such high dimensional space will cause large computational and storage requirements. Therefore, dimensionality reduction techniques are employed to reduce the computational constraints. Parametric Discriminant Analysis (PDA), also called Linear Discriminant Analysis (LDA) [7, 8] is one of the supervised learning methods used for feature extraction in various pattern recognition applications including face recognition. In this method, classification of data samples is done by transforming the input sample to another space where between-class scatter $\mathbf{S}_b$ is maximized and within-class scatter $\mathbf{S}_w$ is minimized simultaneously.

The performance of LDA is satisfactory when there are enough samples available but degrades when only few samples are available. This degradation is due to large dimensionality and availability of few samples, a well-known problem in literature called curse-of-dimensionality or small-sample-size (SSS) [9]. Various subspace methods and extensions of PDA are proposed in literature to counter SSS problem [10, 11, 12, 13, 14, 15]. A comparative study of LDA and its variants is conducted by Kumar et al. [16] which solve SSS problems to a certain extent. Li et al. [17] pointed out that there are some inherent limitations in LDA which affect its performance dramatically. Firstly, the computation of scatter matrices involved in LDA presumes that data samples are normally distributed. Secondly, the maximum features that can be extracted using LDA are $c-1$ (where c is the number of classes). Thirdly, the between-class scatter takes into consideration only the class means and thus is unable to capture the boundary structure of the classes effectively. Fourthly, the transformation matrix $\mathbf{W}$ to be found involves the inverse of $\mathbf{S}_w$ which becomes singular when few data samples are available.

In order to address these limitations, Nonparametric Discriminant Analysis (NDA) is proposed by Fukunaga and Mantock [18]. Originally, the method was

suggested for two class problem. In this technique, the between-class scatter was defined in a non-parametric manner based on k-nearest neighbors (k-NN) perspective. Li et al. [17] extended the original NDA to multi-class NDA. In their research work, the within-class scatter is modified in a non-parametric way and called this approach non-parametric feature analysis (NFA). Bressan and Vitria [19] have investigated the effect of k on the performance of NDA. They have pointed out that the choice of k does not affect the results significantly. Also, when the value of k equals the number of samples available, the features extracted will be same as LDA. They have modified the definition of between-class scatter and within-class scatter based on nearest-neighbor (NN), i.e. $k = 1$ and showed that the proposed method outperformed the original NDA and LDA in terms of classification accuracy.

All the methods discussed above assume that all the training samples are available in advance. However, this assumption may not hold in many circumstances in which data is not collected at once but gathered as stream of chunks of size one or more. If the frequency with which the system is updated is relatively high, then it becomes of utmost importance that the developed system should have the capability of dynamically updating its model. Further, if certain data samples are found which degrade the performance, the system should be capable enough to remove such samples from its training model. For handling such scenarios, Hall et al. [20] proposed PCA merging and splitting in which two different eigenspace models can be merged or split. Similarly, Shaoning et al. [21, 22] have proposed incremental LDA (ILDA) for classification of data streams and decremental LDA (DLDA) in which the system is dynamically updated. Besides these two models, there has been several research works for dynamically updating the discriminant models. Zhao et al. [23] have also proposed an incremental model for updating Linear Discriminant Analysis using generalized singular value decomposition (GSVD) and call their method GSVD-ILDA. Lamba et al. [24] have suggested an incremental subclass discriminant analysis in which the incremental model is based on updating each individual class. Raducanu and Vitria [25] have proposed an approach for online learning in Nonparametric Discriminant Analysis. Pang et al. [26] have proposed Active Mode Incremental NDA approach which is based on residues but requires more computation time and achieves comparable classification accuracy as the method given in [25].

Motivated by the fact that NDA outperforms LDA and the approach of Hall et al. and Shaoning et al., in this paper, we propose Incremental NDA (INDA) and Decremental NDA (DNDA). The proposed algorithms update the existing NDA model by incorporating the new data samples and forgetting the unwanted samples respectively. The proposed algorithm outperforms ILDA and DLDA methods in terms of recognition accuracy and also requires less training time in comparison to the batch method. The rest of the paper is organized as follows: Section 2 briefly discusses batch PDA and batch NDA. The proposed algorithms are given in Section 3. The experimental setup and results on four publicly available face datasets are presented and analyzed in Section 4. Some concluding remarks and future work are included in Section 5.

2 BATCH PDA AND BATCH NDA

Let us assume that $\mathbf{X}$ represents the dataset containing a total of N data samples $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N\}$ each having dimensionality n. Each sample belongs to one of c classes 1, 2, ..., C. Assuming i^{th} class consists of N_i samples, then the total samples are $N = \sum N_i$.

2.1 Parametric Discriminant Analysis (PDA)

Linear Discriminant Analysis (LDA) also called PDA [7], is based on Gaussian distribution of data samples. The between-class scatter $\mathbf{S}_b$ and within-class scatter $\mathbf{S}_w$ are mathematically defined as:

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

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

where $\mathbf{m} = (1/N)\sum_{i=1}^{N} \mathbf{x}_i$ is the mean of the whole dataset and $\mathbf{m}_i$ represents the mean of i^{th} class. For batch LDA, Fisher criterion to find the optimal transformation $\mathbf{W}$ is given by [8]:

$$J(\mathbf{W}) = \underset{\mathbf{W}}{^{arg\,max}} \frac{|\mathbf{W}^T \mathbf{S}_b \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_w \mathbf{W}|}. \tag{3}$$

The optimal transformation matrix $\mathbf{W}$ can be found by solving the following generalized eigenvalue decomposition problem:

$$\mathbf{S}_b \mathbf{W} = \lambda \mathbf{S}_w \mathbf{W} \quad \text{or} \quad \mathbf{S}_w^{-1} \mathbf{S}_b = \lambda \mathbf{W}. \tag{4}$$

2.2 Nonparametric Discriminant Analysis (NDA)

Originally, NDA for two-class problem was suggested by Fukunaga and Mantock [18]. This was extended to multi-class problems by Li et al. [17]. The optimal transformation $\mathbf{W}_{NDA}$ in original NDA is found by solving the following criterion:

$$\mathbf{W}_{NDA} = \underset{\mathbf{W}_{NDA}^T \mathbf{S}_w \mathbf{W}_{NDA} = \mathbf{I}}{^{arg\,max}} trace(\mathbf{W}_{NDA}^T \hat{\mathbf{S}}_b \mathbf{W}_{NDA}) \tag{5}$$

where $\hat{\mathbf{S}}_b = \sum_{i=1}^{c} \sum_{j=1, j\neq i}^{c} \sum_{r=1}^{N_i} w(i,j,r)(\mathbf{x}_r^i - \mathbf{m}_j(\mathbf{x}_r^i))(\mathbf{x}_r^i - \mathbf{m}_j(\mathbf{x}_r^i))^T$ and $\mathbf{m}_j(\mathbf{x}_r^i) = \frac{1}{k}\sum_{p=1}^{k} NN_p(\mathbf{x}_r^i, j)$

Here, $\text{NN}_p(\mathbf{x}_r^i, j)$ is the p^{th} nearest neighbor from class j to $\mathbf{x}_r^i$. For computational efficiency, we have assumed the weights $w(i,j,r)$ equal to unity. The steps to obtain $\mathbf{W}_{NDA}$ are listed in Table 1.

In the original formulation, the between-class scatter was redefined in nonparametric manner while the within-class scatter retained the same form as PDA. Thus,

1. Given N data items as n-dimensional columns in data matrix $\mathbf{X}$, compute $\mathbf{S}_w$ as given in Equation (2).
2. Find the eigenvectors (represented as columns of Θ) and eigenvalues (represented as columns of Λ) of $\mathbf{S}_w$.
3. Whiten the data as $\mathbf{M} = \Lambda^{1/2}\Theta^T\mathbf{X}$ to obtain K_w – dimensional feature vectors (where K_w is the number of non-zero eigenvalues).
4. Compute $\hat{\mathbf{S}}_b$ on the whitened data.
5. Find the eigenvectors of $\hat{\mathbf{S}}_b$ and place them as columns in Ψ sorted by decreasing eigenvalues.
6. Select the first $K_b = \min(K_w, n, rank(Sb))$ dominant eigenvectors such that $\Psi_K = [\Psi_1, \Psi_2, \ldots, \Psi_{K_b}]$.
7. The resulting transformation is $\mathbf{W}_{NDA} = \Psi_K^T\Lambda^{1/2}\Theta^T$ and projected data, $\mathbf{Z} = \mathbf{W}_{NDA}\mathbf{X} = \Psi_K^T\mathbf{M}$.

Table 1. Algorithm to find optimal transformation in Equation (5)

it still considers Gaussian assumption in the data samples. To remove this assumption also, Bressan and Vitria [19] defined both $\mathbf{S}_b$ and $\mathbf{S}_w$ based on the nearest neighbor. The between-class scatter $\mathbf{S}_b$ is constructed from the nearest neighbor pointing to other class while within-class scatter is defined from nearest neighbor within the same class. The between-class and within-class nearest neighbors for a data sample $\mathbf{x} \in c$ are defined as:

$$\mathbf{x}_b = \{\mathbf{x}' \in \bar{c} \,||\mathbf{x}' - \mathbf{x}|| \leq ||\mathbf{z} - \mathbf{x}||, \forall \mathbf{z} \in \bar{c}\}, \tag{6}$$

$$\mathbf{x}_w = \{\mathbf{x}' \in c \,||\mathbf{x}' - \mathbf{x}|| \leq ||\mathbf{z} - \mathbf{x}||, \forall \mathbf{z} \in c\}. \tag{7}$$

Though the above definitions are for 1-NN, these can be extended to k-NN by defining $\mathbf{x}_b$ and $\mathbf{x}_w$ as the mean of k-nearest between or within-class neighbors. The between-class differences Φ_b^i and within-class differences Φ_w^i of i^{th} tuple are defined as $\Phi_b^i = \mathbf{x}^i - \mathbf{x}_b$ and $\Phi_w^i = \mathbf{x}^i - \mathbf{x}_w$, respectively. Thus, $\mathbf{S}_b$ and $\mathbf{S}_w$ are defined as [19]:

$$\mathbf{S}_b = \frac{1}{N}\sum_{i=1}^{N} w_i(\Phi_b^i)(\Phi_b^i)^T, \tag{8}$$

$$\mathbf{S}_w = \frac{1}{N}\sum_{i=1}^{N} w_i(\Phi_w^i)(\Phi_w^i)^T. \tag{9}$$

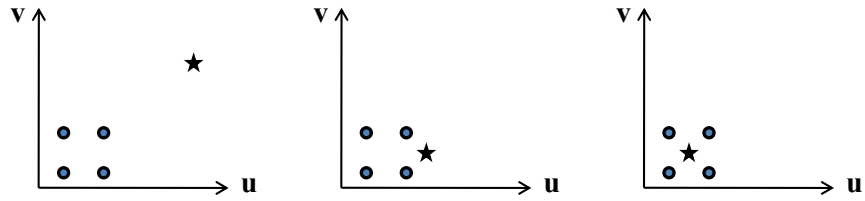
The optimal transformation $\mathbf{W}_{NDA2}$ is found by the following criterion [19]:

$$\mathbf{W}_{NDA2} = {}^{arg\,max}_{E\{||\mathbf{S}_w||\}=1} E\{||\mathbf{S_b}||\}. \tag{10}$$

We have formulated the proposed algorithms based on Bressan and Vitria [19] algorithm described in the next section.

3 PROPOSED ALGORITHMS

As discussed in Section 1, new data samples may arrive at different points of time in various chunks. Thus, with the new samples arriving, it is interesting to note for which samples the nearest neighbors (NNs) get changed. The new sample belonging to a particular class i may or may not affect the NN of the data samples already in that class. It is illustrated with 2-D data samples shown in Figure 1:



a) b) c)

Figure 1. Effect of new data samples on nearest neighbor of existing data samples (dots represent the data items already in the dataset while star represents the new data sample)

Figure 1 a) shows the best case when none of the NN of existing data samples is changed, while in Figure 1 b) *NNs* of some samples are affected which is common in practice. Figure 1 c) represents the worst case where *NNs* of all the samples are affected. The worst case is rare to happen practically because it requires all the data samples at equal distance and the new sample to be exactly at equal distance from all the samples. Now, suppose the batch NDA is represented as 3-tuple $(\mathbf{X}, \mathbf{S}_w, \mathbf{D})$ where $\mathbf{X}$ represents the data samples arranged as column vectors, $\mathbf{S}_w$ represents the within-class scatter and $\mathbf{D} = [d_{ij}]$ represents distance matrix whose elements are the Euclidean distances between i^{th} and j^{th} data sample. The mathematical formulation of the proposed algorithms is discussed next.

3.1 Sequential Incremental NDA (SINDA)

Suppose the new data items are collected sequentially, $\mathbf{x}_{N+1}$, $\mathbf{x}_{N+2}$, ... and so on. Let the $(N+1)^{\text{th}}$ data item $\mathbf{y}$ acquired has class label l, then the updated NDA model $(\mathbf{X}', \mathbf{S}_w{}', \mathbf{D}')$ can be computed from batch model such that:

$$\mathbf{X}' = [\mathbf{X} \quad \mathbf{y}],$$
$$\mathbf{D}' = \begin{bmatrix} \mathbf{D} & \mathbf{A}; & \mathbf{A}^T & \mathbf{0} \end{bmatrix}$$

where $\mathbf{A}$ represents the column vector of the Euclidean distances of the new sample from the already available samples. Similarly, the updated within-class scatter can be found as explained next. If $l = c + 1$ represents a newly introduced class then the within-class scatter $\mathbf{S}_w$ is updated by multiplying by a constant factor as below:

$$\mathbf{S}_w{}' = \frac{N}{N+1}\mathbf{S}_w. \tag{11}$$

On the other hand, if the new sample is from an existing class, then it will affect the NNs of the samples of that class. Based on $\mathbf{D}'$, we can find the data samples (say $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t$) for which the NN is changed due to the new sample. Let Δ^i_{old} denote the old within-class differences and Δ^i_{new} denote the new within-class differences as defined below:

$$\Delta^i_{old} = (\mathbf{x}_i - \mathbf{x}_w) \quad \text{and} \quad \Delta^i_{new} = (\mathbf{x}_i - \mathbf{x}_w{}')$$

where $\mathbf{x}_w{}'$ is the new NN of i^{th} data sample. Hence, the within-class scatter matrix is updated as follows:

$$\mathbf{S}_w{}' = \frac{1}{N+1}\left(N \times \mathbf{S}_w - \sum_{i=1}^{t} \Delta^i_{old}(\Delta^i_{old})^T + \sum_{i=1}^{t} \Delta^i_{new}(\Delta^i_{new})^T + (\mathbf{y} - \mathbf{y}_w)(\mathbf{y} - \mathbf{y}_w)^T\right) \tag{12}$$

where $\mathbf{y}_w$ represents the NN of the new data sample belonging to class l.

3.2 Chunk Incremental NDA (CINDA)

When the new data items are acquired in chunks $\mathbf{Y}_0, \mathbf{Y}_1, \ldots$ (at times $t = 0, 1, \ldots$, respectively) and so on such that $\mathbf{Y} = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$ represents k new samples, the NN of the data samples of more than one class get changed. Therefore, in this case, the updated NDA model $(\mathbf{X}', \mathbf{S}_w{}', \mathbf{D}')$ can be computed as given below:

$$\mathbf{X}' = [\mathbf{X} \quad \mathbf{y}],$$
$$\mathbf{D}' = \begin{bmatrix} \mathbf{D} & \mathbf{A}; & \mathbf{A}^T & \mathbf{B} \end{bmatrix}$$

where $\mathbf{A}$ represents the $N \times k$ matrix of the Euclidean distances of the new sample from the already available samples and $\mathbf{B}$ represents the $k \times k$ matrix of Euclidean distances between the new samples. Based on $\mathbf{D}$, we can find samples (say $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t$) for which the NN are changed. For such samples, let Δ^{ij}_{old} denote the old within-class differences of the original dataset belonging to i^{th} class and $1 \le j \le t_i$. Similarly Δ^{ij}_{new} denotes the within-class differences of the data samples of i^{th} class new after incorporating the new data samples and is given below:

$$\Delta^{ij}_{old} = (\mathbf{x}_i - \mathbf{x}_w) \quad \text{and} \quad \Delta^{ij}_{new} = (\mathbf{x}_i - \mathbf{x}_w{}')$$

where $\mathbf{x}_w{}'$ is the new NN of i^{th} data sample. Based on these differences, we define two matrices $\mathbf{P} = \sum_i \sum_{j=1}^{t_i} (\Delta^{ij}_{old})(\Delta^{ij}_{old})^T$ and $\mathbf{R} = \sum_i \sum_{j=1}^{t_i} (\Delta^{ij}_{new})(\Delta^{ij}_{new})^T$ which denote the within-class scatter in the original data samples and after new data samples added, respectively. Thus, the updated within-class scatter can be computed as follows:

$$\mathbf{S}_w{}' = \frac{1}{N+k}\left(N \times \mathbf{S}_w - \mathbf{P} + \mathbf{R} + (\mathbf{Y} - \mathbf{Y}_w)(\mathbf{Y} - \mathbf{Y}_w)^T\right) \tag{13}$$

where $\mathbf{Y}_w$ represents the NNs of the new samples corresponding to their respective classes. Once the updated within-class scatter $\mathbf{S}_w{}'$ is computed as above, the optimal transformation $\mathbf{W}_{NDA2}$ can be found according to the algorithm given in Table 1.

3.3 Sequential Decremental NDA (SDNDA)

As discussed earlier, certain data samples which may degrade the performance of the model need to be removed. For handling such situations, the unwanted samples can be removed from the existing model as described next. Let the data items be removed sequentially. Let the j^{th} data item $\mathbf{y}$ have class label l, then the updated NDA model $(\mathbf{X}', \mathbf{S}_w{}', \mathbf{D}')$ can be computed from batch model such that:

$$\mathbf{X}' = \{\mathbf{X}\} - \mathbf{y},$$
$$\mathbf{D}' = \mathbf{D}_{((1:j-1):(j+1:N),(1:j-1):(j+1:N))}$$

where $\mathbf{D}'$ is obtained by removing the j^{th} row and j^{th} column from $\mathbf{D}$. The removal of the sample will affect the NNs of the samples of that class. Based on $\mathbf{D}$, we can find the data samples for which the NN is changed due to the removed sample. Thus let Δ^i_{old} denote the old within-class differences and Δ^i_{new} denote the new within-class differences defined in same way as Section 3.1. Thus, the within-class scatter matrix is updated as follows:

$$\mathbf{S}_w{}' = \frac{1}{N-1}\left(N \times \mathbf{S}_w - \sum_{i=1}^{t} \Delta^i_{old}(\Delta^i_{old})^T + \sum_{i=1}^{t} \Delta^i_{new}(\Delta^i_{new})^T + (\mathbf{y} - \mathbf{y}_w)(\mathbf{y} - \mathbf{y}_w)^T\right) \tag{14}$$

where $\mathbf{y}_w$ represents the NN of the old data sample belonging to class l.

3.4 Chunk Decremental NDA (CDNDA)

When the new data items are removed in chunks $\mathbf{Y}_0$, $\mathbf{Y}_1$, ... (at time $t = 0, 1, \ldots$, respectively) and so on such that $\mathbf{Y} = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$ represents k old samples, the NN of the data samples of more than one class get changed. Therefore, in this case the updated NDA model $(\mathbf{X}', \mathbf{S}_w{}', \mathbf{D}')$ can be computed as given below:

$$\mathbf{X}' = \{\mathbf{X}\} - \mathbf{Y},$$
$$\mathbf{D}' = \mathbf{D}_{\{1:N\}-\{1:k\},\{1:N\}-\{1:k\}}$$

where $\mathbf{D}$ represents the $(N-k) \times (N-k)$ matrix of the Euclidean distances of the remaining samples in the database. Based on $\mathbf{D}$, we can find samples for which the NN are changed. For such samples, let Δ^i_{old} denote the within-class differences of the existing dataset belonging to i^{th} class and Δ^i_{new} denote the within-class

differences of the data samples of i^{th} class after removing the old data samples. Based on these differences, we define two matrices $\mathbf{P} = \sum_i \sum_{j=1}^{t_i} (\Delta_{old}^{ij})(\Delta_{old}^{ij})^T$ and $\mathbf{R} = \sum_i \sum_{j=1}^{t_i} (\Delta_{new}^{ij})(\Delta_{new}^{ij})^T$ which denote the within-class scatter in the original data samples and after data samples removed, respectively. The definition of Δ_{old}^{ij} and Δ_{new}^{ij} is the same as given in Section 3.2. Thus, the updated within-class scatter can be computed as follows:

$$\mathbf{S}_w{}' = \frac{1}{N-k}\left(N \times \mathbf{S}_w - \mathbf{P} + \mathbf{R} + (\mathbf{Y} - \mathbf{Y}_w)(\mathbf{Y} - \mathbf{Y}_w)^T\right) \tag{15}$$

where $\mathbf{Y}_w$ represents the NNs of the new samples corresponding to their respective classes. Once the updated within-class scatter $\mathbf{S}_w{}'$ is computed as above, the optimal transformation $\mathbf{W}_{NDA2}$ can be found according to the algorithm given in Table 1. The training and the test images are transformed using $\mathbf{W}_{NDA2}$ and recognition is performed with nearest neighbor classifier. As no approximation is used in updating scatter matrix, the performance of the Batch NDA is the same as the proposed algorithms.

4 EXPERIMENTAL SET-UP AND RESULTS

To check the efficacy of the proposed algorithms, we evaluate the performance of the proposed algorithm on four publicly available face datasets viz. AR [27, 28], CMU-PIE [29], ORL [30] and Yale [31]. The performance of the proposed algorithms is evaluated in terms of average classification accuracy and the training computation time. A brief review of the face datasets used for experiments is given as follows:

4.1 Datasets Used

The AR face dataset [27, 28] consists of a total of 2 600 images of 100 identities captured in two sessions. The facial images have varying illumination, expression and occlusion. The illumination subset was selected for experiments and original cropped images of size 120×100 were resized to 34×25. The CMU-PIE face dataset [29] comprises 41 368 images of 68 identities with 13 different poses, 43 illumination conditions and 4 expressions. We selected the images with varying illumination variation and background light off with 21 images of each person except one person for which only 18 images are available. Hence, the total number of face images was 1 425 ($= 67 \times 21 + 18$). The original images were first cropped manually and then resized to 28×25. The ORL face dataset [30] consists of 400 images of 40 subjects, each with 10 images per person. The facial images contain slight variations in pose and illumination. We have used whole ORL dataset for our experiments. The original images of 112×92 were resized to 30×25. The Yale face dataset [27] has 165 grayscale images of 15 persons with 11 images for each person. These 11 images are captured for each different facial expression or config-

urations: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. Each image was first cropped and then rescaled to 28×25. Sample images from each of the four face datasets are shown in Figure 2.

The images in all the datasets used for experiments are resized such that the aspect ratio remains unchanged. For each dataset, we randomly partition the data into training set and test set. This partitioning process is repeated 10 times to obtain average classification accuracy. The classification accuracy of INDA and DNDA is compared with the existing algorithm ILDA and DLDA, respectively. The summary of datasets used for experiments are given in Table 2. All the experiments are performed on Intel PC Core 2 Duo 3.0 GHz with 8 GB RAM running MATLAB 7.8.

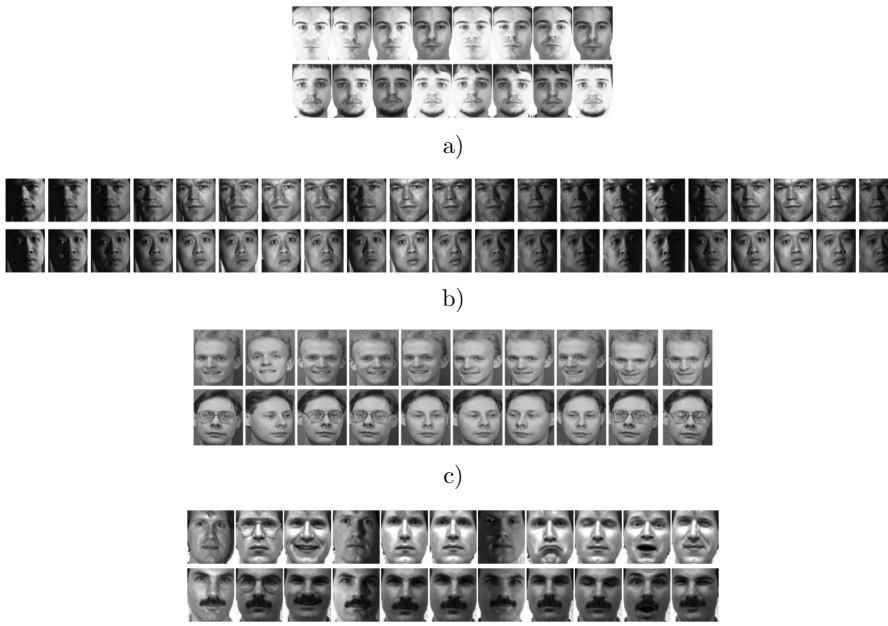


Figure 2. Some randomly chosen face images from the three face datasets: a) AR, b) PIE, c) ORL, d) Yale

4.2 Classification Accuracy

4.2.1 Incremental NDA

The classification accuracy of the proposed INDA algorithm averaged over 10 random splits for the four face datasets and at different intervals is shown in Fi-

Database	Classes	Total Images	% Used for Initial Model
AR	100	800	25
CMU-PIE	68	1 425	20
ORL	40	400	25
Yale	15	165	20

Table 2. Summary of the datasets used in experiments

gure 3 a)–d). We can observe that the proposed algorithm outperforms the earlier ILDA method on all the datasets. In addition, the performance of the proposed approach improves consistently with increase in the number of samples while the performance of earlier ILDA approach may improve or deteriorate with the same scenario. The results of ILDA are also consistent with that of [21]. The reason for poor performance of ILDA is that the number of samples available per class is small resulting into poor estimation of scatter matrices. Further, as the new samples from new classes are added, ILDA performance does not increase consistently.

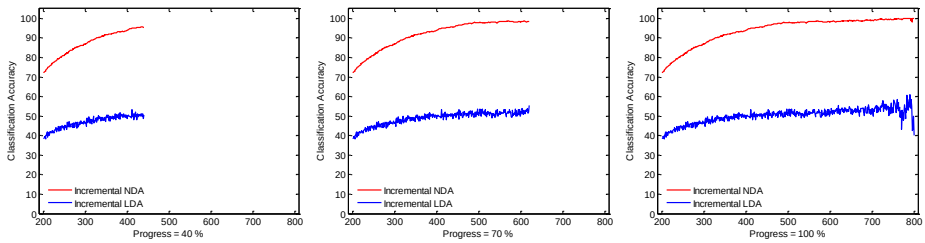
4.2.2 Decremental NDA

The classification accuracy for decremental NDA is depicted in Figure 4. It is readily observed that the proposed DNDA algorithm outperforms DLDA on all the datasets. As DLDA cannot be implemented with single data sample because no model can be developed with single image, we have used larger chunk size for DLDA. In contrast, DNDA provides this flexibility also thus allowing removal of single sample or chunks from the model.

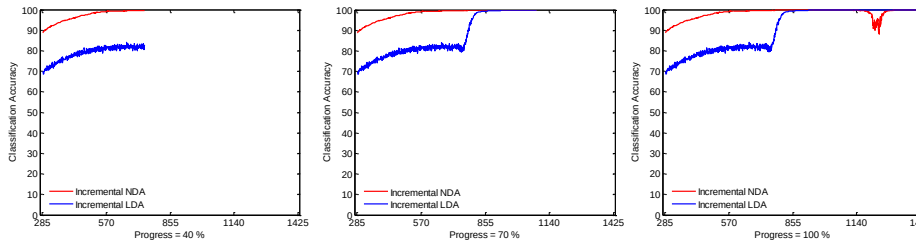
4.2.3 Computational Complexity Analysis

The time complexity of Batch NDA, SINDA, CINDA, SDNDA and CDNDA are given in Table 3 with N existing data samples and k new samples. The computational complexity of the method depends mainly on computation of distance matrix and the within-class scatter. It is easy to observe that for the proposed methods, the complexity is significantly reduced from $O(N^2)$ to $O(Nk)$ where usually $k << N$. Further, t is the number of data samples whose NN is changed due to the new sample(s) and again $t << N$ as discussed in Section 3.

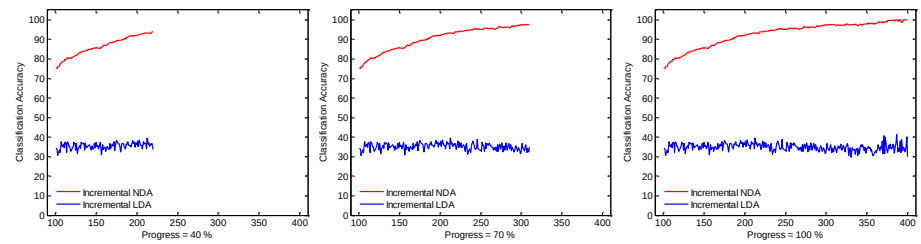
The training time required for the proposed INDA algorithm in comparison to batch NDA is shown in Figure 5 for the four datasets. The initial model for all the datasets was developed with some percentage of total images as discussed in Section 4.1. We can observe from the graphs shown in Figure 5 that the time taken by the INDA model is less in comparison to batch NDA for all the datasets. For the AR dataset, the complete model development requires 1.9 s approximately while the proposed approach takes less than 0.8 s which is a significant improvement. Similar improvements are also observed for ORL and Yale datasets. In CMU-PIE, though



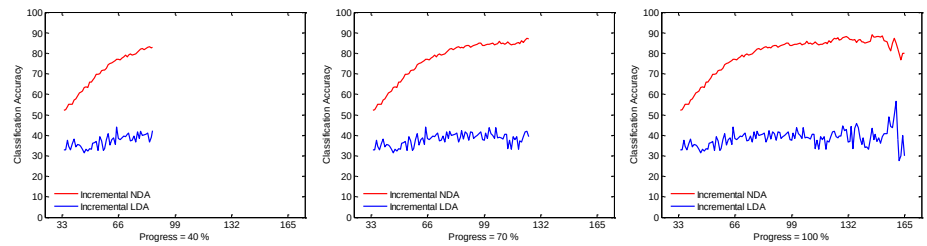
a)



b)



c)



d)

Figure 3. Comparison of average classification accuracy of INDA and ILDA at various intervals on the four datasets: a) AR, b) PIE, c) ORL, d) Yale

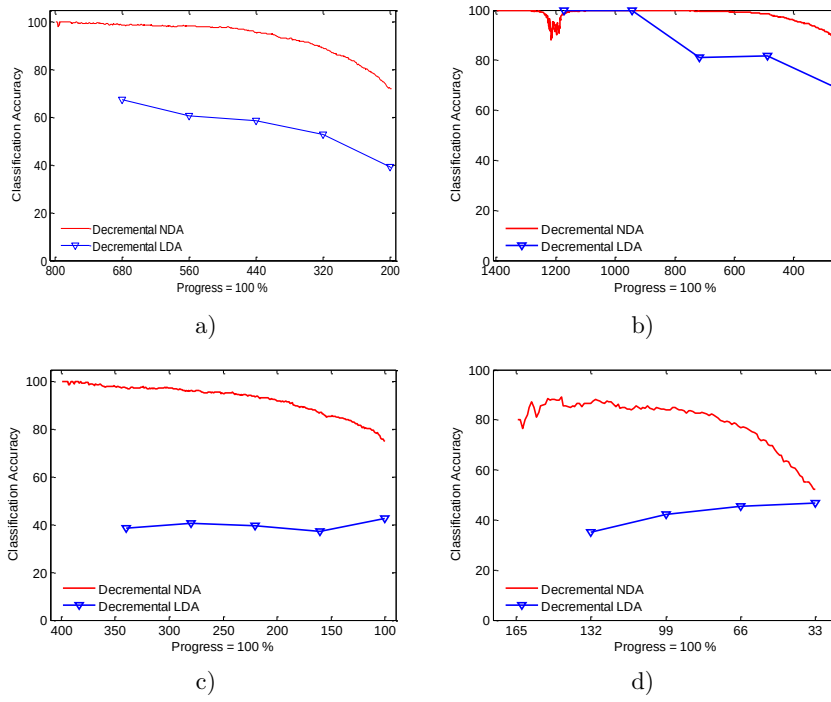


Figure 4. Comparison of average classification accuracy of DNDA and DLDA on the four datasets: a) AR, b) PIE, c) ORL, d) Yale

Method	**Time Complexity**
Batch NDA	$(N+k)(N+k-1)n + (1/2)(N+k)n^2$
SINDA/SDNDA*	$2Nn + (2t+1)n^2$
CINDA/CDNDA*	$2Nk + k(k-1)n + (2t+k)n^2$

* Complexity of SINDA is the same as SDNDA and that of CINDA is the same as CDNDA

Table 3. Complexity comparison of algorithms

the time taken increases consistently with the number of images, still INDA requires less time than its batch counterpart.

5 CONCLUSION AND FUTURE WORK

Parametric Discriminant Analysis is one of the benchmark techniques for feature extraction in pattern recognition literature. But this technique suffers from two major problems: requirement of enough samples for estimation of scatter matrices

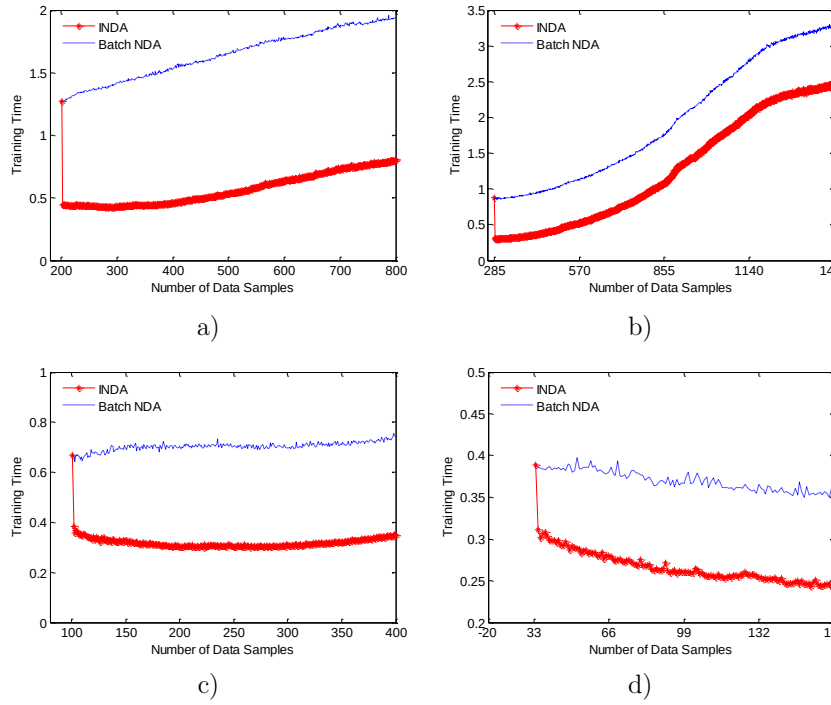


Figure 5. Training Time (sec) of INDA algorithm compared with Batch NDA: a) AR, b) PIE, c) ORL, d) Yale dataset

and assumption of the samples to be drawn from Gaussian distribution. For practical problems such as face recognition, both the conditions need not to be satisfied. Further, all the samples required for the model development may not be available in advance. To counter both these challenges simultaneously, in this paper, we proposed INDA and DNDA algorithms with improved efficiency. The INDA algorithm is further divided into two categories, i.e. sequential INDA and chunk INDA, depending upon the number of new data samples being one or more than one, respectively. Similarly DNDA is divided into two categories, i.e. sequential DNDA and chunk DNDA. The experimental results on four face datasets, i.e. AR, CMU-PIE, ORL and Yale, show that the proposed algorithm is robust against small sample size (SSS) problem. Based on the results, we can summarize the characteristics of the proposed method as follows:

1. The classification accuracy of INDA and DNDA outperforms ILDA and DLDA, respectively, when only few data samples are available.

2. INDA requires less computation training time than batch NDA exploiting advantage of the already developed model.

3. Computational complexity of the proposed algorithm is notably less in comparison to the batch mode.
4. Decremental learning with only one sample is possible with the proposed algorithm in contrast to DLDA which requires large chunks.
5. The performance of INDA and DNDA is same as batch NDA in terms of classification accuracy as no approximation is used.

Although we have performed experiments on the face datasets, it cannot be overlooked that the proposed algorithm may prove useful in other domains such as document classification, handwriting recognition and target recognition, etc. In future work, we would like to investigate the possibility of incremental formulation of other popular feature extraction methods.

REFERENCES

[1] Prabhakar, S.—Pankati, S.—Jain, A. K.: Biometric Recognition: Security and Privacy Concerns. IEEE Security and Privacy, Vol. 1, 2003, pp. 33–42.

[2] Zhao, W.—Chellappa, R.—Rosenfield, A.—Philips, P. J.: Face Recognition: A Literature Survey. ACM Computing Surveys (CSUR), Vol. 35, 2003, No. 4, pp. 399–458.

[3] Cordiner, A.: Illumination Invariant Face Detection. University of Wollongong, MComSc thesis, 2009.

[4] Zou, X.—Kittler, J.—Messer, K.: Illumination Invariant Face Recognition: A Survey. Proceedings of the First IEEE International Conference on Biometrics: Theory, Applications, and Systems, Virginia, USA, September 2007, pp. 1–8.

[5] Turk, M.—Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience, Vol. 3, 1991, pp. 71–86.

[6] Murase, H.—Nayar, S. K.: Visual Learning and Recognition of 3-D Objects from Appearance. International Journal of Computer Vision, Vol. 14, 1995, No. 1, pp. 5–24.

[7] Fukunaga, K.: Introduction to Statistical Pattern Recognition. San Diego, Academic Press, USA, 1990.

[8] Duda, R. O.— Hart, P. E.—Stork, D.: Pattern Classification. 2nd Edition. John Wiley and Sons, NY, 2000.

[9] Bellman, R. E.: Adaptive Control Processes. Princeton University Press, Princeton, 1961.

[10] Belhumeur, P. N.—Hespanha, J. P.—Kriegman, D. J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, 1997, No. 7, pp. 711–720.

[11] Raudys, S.—Duin, R. P. W.: Expected Classification Error of the Fisher Linear Classifier with Pseudoinverse Covariance Matrix. Pattern Recognition Letters, Vol. 19, 1998, No. 5-6, pp. 385–392.

[12] Chen, L. F.—Liao, H. M.—Ko, M.—Lin, J.—Yu, G.: A New LDA-Based Face Recognition System which Can Solve the Small Sample Size Problem. Pattern Recognition, Vol. 33, 2000, No. 10, pp. 1713–1726.

[13] Wang, X.—Tang, X.: Dual-Space Linear Discriminant Analysis for Face Recognition. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR04), Washington, Vol. 2, 2004, pp. 564–569.

[14] Ye, J.—Li, Q.: A Two-Stage Linear Discriminant Analysis via QR-Decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, 2005, No. 6, pp. 929–941.

[15] Jiang, X.—Mandal, B.—Kot, A.: Eigenfeature Regularization and Extraction in Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, 2008, No. 3, pp. 383–394.

[16] Kumar, N.—Jaiswal, A.—Agrawal, R. K.: Performance Evaluation of Subspace Methods to Tackle Small Sample Size Problem in Face Recognition. Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI '12), ACM, New York, NY, USA, 2012, pp. 938–944.

[17] Li, Z.—Lin, D.—Tang, X.: Nonparametric Discriminant Analysis for Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, 2009, No. 4, pp. 755–761.

[18] Fukunaga, K.—Mantock, J. M.: Nonparametric Discriminant Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 5, 1983, No. 6, pp. 671–678.

[19] Bressan, M.—Vitri, J.: Nonparametric Discriminant Analysis and Nearest Neighbor Classification. Pattern Recognition Letters, Vol. 24, 2003, No. 15, pp. 2743–2749.

[20] Hall, P.—Marshall, D.—Martin, R.: Merging and Splitting Eigenspace Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, 2000, No. 9, pp. 1042–1049.

[21] Shaoning, P.—Ozawa, S.—Kasabov, N.: Incremental Linear Discriminant Analysis for Classification of Data Streams. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 35, 2005, No. 5, pp. 905–914.

[22] Shaoning, P.—Ban, T.—Kadobayashi, Y.—Kasabov, N.: Incremental and Decremental LDA Learning with Applications. Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, July 2010, pp. 1–8.

[23] Zhao, H.—Yuen, P. C.: Incremental Linear Discriminant Analysis for Face Recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. Vol. 38, 2008, No. 1, pp. 210–221.

[24] Lamba, H.—Dhamecha, T. I.—Vatsa, M.—Singh, R.: Incremental Subclass Discriminant Analysis: A case study in Face Recognition. Proceedings of the 19th IEEE International Conference on Image Processing (ICIP), Florida, USA, September 2012, pp. 593–596.

[25] Raducanu, B.—Vitria, J.: Online Nonparametric Discriminant Analysis for Incremental Subspace Learning and Recognition. Pattern Analysis and Applications, Vol. 11, 2008, Vol. 3-4, pp. 259–268.

[26] Pang, S.—Dhoble, K.—Chen, Y.—Kasabov, N.—Ban, T.—Kadobayashi, Y.: Active Mode Incremental Nonparametric Discriminant Analysis Learning. Proceedings of the Eighth International Conference on Information and Management Sciences, Kunming, China, July 2009, pp. 407–412.

[27] Martinez, A. M.—Benavente, R.: The AR Face Database. CVC Technical Report No. 24, 1998.

[28] Martinez, A. M.—Kak, A. C.: PCA Versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, 2001, No. 2, pp. 228–233.

[29] Sim, T.—Baker, S.—Bsat, M.: The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces. Technical Report CMU-RITR-01-02. The Robotics Institute, Carnegie Mellon University, 2001.

[30] Samaria, F. S.—Harter, A. C.: Parameterisation of a Stochastic Model for Human Face Identification. Proceedings of the Second IEEE Workshop on Applications of Computer Vision, Florida, USA, 1994, pp. 138–142.

[31] Yale Face Database: Available on: `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`.

Nitin Kumar received his Ph.D. and M.Tech. degrees in computer science and technology from Jawaharlal Nehru University, New Delhi. Currently, he is working as Assistant Professor in Department of Computer Science and Engineering at National Institute of Technology Uttarakhand, India. His current research areas include face recognition, pattern classification, image processing, visual attention modeling.

Ramesh Kumar Agrawal received M.Tech. from Indian Institute of Technology Delhi, New Delhi, and Ph.D. (computational physics) from University of Delhi, Delhi. Currently he is working as Professor at the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. His current research areas are: classification, feature extraction and selection problems in domains of image processing, security, and bioinformatics.

Ajay Jaiswal did his graduation from Dayalbagh Educational Institute, Agra, and post-graduation from IIT Roorkee. He received his Ph.D. degree from Jawaharlal Nehru University, Delhi, in the area of pattern recognition. He is working as Assistant Professor in S. S. College of Business Studies, University of Delhi, Delhi. He has more than 14 years of teaching experience.