

A New Incremental Optimal Feature Extraction Method for On-line Applications

Youness Aliyari Ghassabeh, Hamid Abrishami Moghaddam

Electrical Engineering Department, K. N. Toosi University of
Technology, Tehran, Iran
y_aliyari@ee.kntu.ac.ir, moghadam@saba.kntu.ac.ir

Abstract. In this paper, we introduced new adaptive learning algorithms to extract linear discriminant analysis (LDA) features from multidimensional data in order to reduce the data dimension space. For this purpose, new adaptive algorithms for the computation of the square root of the inverse covariance matrix $\Sigma^{-1/2}$ are introduced. The proof for the convergence of the new adaptive algorithm is given by presenting the related cost function and discussing about its initial conditions. The new adaptive algorithms are used before an adaptive principal component analysis algorithm in order to construct an adaptive multivariate multi-class LDA algorithm. Adaptive nature of the new optimal feature extraction method makes it appropriate for on-line pattern recognition applications. Both adaptive algorithms in the proposed structure are trained simultaneously, using a stream of input data. Experimental results using synthetic and real multi-class multi-dimensional sequence of data, demonstrated the effectiveness of the new adaptive feature extraction algorithm.

Keywords: Adaptive Learning Algorithm, Adaptive Linear Discriminant Analysis, Feature Extraction.

1 Introduction

Feature extraction is generally considered as a process of mapping the original measurements into a more effective feature space. When we have two or more classes, feature extraction consists of choosing those features which are most effective for preserving class separability in addition to dimension reduction [1]. Linear discriminant analysis (LDA) has been widely used in pattern recognition applications, such as feature extraction, face and gesture recognition [2-4]. LDA also known as fisher discriminant analysis (FDA) seeks directions for efficient discrimination during dimension reduction [1].

Typical implementation of this technique assumes that a complete dataset for training is available, and learning is carried out in one batch. However, when we conduct LDA learning over datasets in real-world applications, we often confront difficult situations where a complete set of training samples is not given in advance. Actually, in most cases such as on-line face recognition and mobile robotics, data are presented as a stream. Therefore, the need for dimensionality reduction, in real time

applications motivated researchers to introduce adaptive versions LDA. Mao and Jain [5] proposed a two layer network, each of which was an adaptive principal component analysis (APCA) network. Chatterjee and Roychowdhury [6] presented adaptive algorithms and a self-organized LDA network for feature extraction from Gaussian data using gradient descent optimization technique. They described algorithms and networks for (i) feature extraction from unimodal and multi-cluster Gaussian data in the multi-class case and (ii) multivariate linear discriminant analysis in multi-class case. Approach presented in [7] suffers from low convergence rate. To solve this drawback, Abrishami Moghaddam et al. [7] derived accelerated convergence algorithms for adaptive LDA (ALDA), based on steepest descent, conjugate direction and Newton-Raphson methods.

In this study, we present new adaptive learning algorithms for the computation of $\Sigma^{-1/2}$. Furthermore, we introduce a cost function related to these algorithms and prove their convergence by discussing about its properties and initial conditions. Finally, we combine our $\Sigma^{-1/2}$ algorithm with an APCA algorithm for ALDA. Each algorithm discussed in this paper considers a flow or sequence of inputs for training; therefore there is no need to a large set of sample data. Memory size and complexity reduction provided by the new ALDA algorithm make it appropriate for on-line pattern recognition and machine vision applications [8, 9]. We will show the effectiveness of these new adaptive algorithms for extracting LDA features using different on-line experiments.

The organization of the paper is as follows. The next section describes the fundamentals of LDA. Section 3, presents the new adaptive algorithms for estimation of the square root of the inverse covariance matrix $\Sigma^{-1/2}$ and analyzes its convergence. Then, by combination of this algorithm with an APCA algorithm in cascade, we implemented an ALDA feature extraction algorithm. Section 4 is devoted to simulations and experimental results. Finally, conclusion remarks are given in section 5.

2 Linear Discriminant Analysis Fundamentals

Let $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, $\mathbf{x} \in \mathfrak{R}^n$ be N samples from L classes $\{\omega_1, \omega_2, \dots, \omega_L\}$. Consider \mathbf{m} and Σ denote the mean vector and covariance matrix of samples, respectively. LDA searches the directions for maximum discrimination of classes in addition to dimensionality reduction. To achieve this goal, within-class and between-class matrices are defined [1]. A within-class scatter matrix is the scatter of the samples around their respective class means \mathbf{m}_i and denoted by Σ_w . The between-class scatter matrix is the scatter of class means \mathbf{m}_i around the mixture mean \mathbf{m} , and denoted by Σ_b . Finally, the mixture scatter matrix is the covariance of all samples regardless of class assignments, and represented by Σ . In LDA, the optimum linear transform is composed of $p(\leq n)$ eigenvectors of $\Sigma_w^{-1}\Sigma_b$ corresponding to its p largest eigenvalues. Alternatively, $\Sigma_w^{-1}\Sigma$ can be used for LDA. A simple analysis

shows that both $\Sigma_w^{-1}\Sigma_b$ and $\Sigma_w^{-1}\Sigma$ has the same eigenvector matrix. In general, Σ_b is not a full rank matrix, hence we shall use Σ in place of Σ_b . The computation of the eigenvector matrix Φ_{LDA} of $\Sigma_w^{-1}\Sigma$ is equivalent to the solution of the generalized eigenvalue problem $\Sigma\Phi_{LDA} = \Sigma_w\Phi_{LDA}\Lambda$, where Λ is the generalized eigenvalue matrix. Under assumption of Σ_w being a positive definite matrix, if we consider $\Psi = \Sigma_w^{1/2}\Phi_{LDA}$, there exists a symmetric $\Sigma_w^{-1/2}$ such that the problem can be reduced to a symmetric eigenvalue problem [1] :

$$\Sigma_w^{-1/2}\Sigma\Sigma_w^{-1/2}\Psi = \Psi\Lambda. \quad (1)$$

3 New Adaptive Learning Algorithms for the LDA Feature Extraction

We use two adaptive training algorithms in cascade for extracting optimal LDA features. The first algorithm called $\Sigma^{-1/2}$ algorithm is for the computation of the square root of the inverse covariance matrix. We prove the convergence of the new adaptive $\Sigma^{-1/2}$ algorithms by introducing a cost function related to them. By minimization of the cost function using gradient descent method, we present our new adaptive $\Sigma^{-1/2}$ algorithms. The second algorithm is an APCA algorithm introduced by Sanger [10] and is used for the computation of the eigenvectors of the covariance matrix. We prove the convergence of the cascade architecture as an ALDA feature selection.

3.1 New Adaptive $\Sigma^{-1/2}$ Algorithm and Convergence Proof

We define the cost function $J(\mathbf{w})$ with parameter \mathbf{w} , $J: \Re^{n \times n} \rightarrow \Re$ as follows:

$$J(\mathbf{W}) = \frac{tr(\mathbf{W}^3 \mathbf{X}\mathbf{X}^t)}{3} - tr(\mathbf{W}). \quad (2)$$

The cost function $J(\mathbf{w})$ is a continuous function with respect to \mathbf{w} . If the sample vectors have zero mean value, the expected value of J will be given by:

$$E(J(\mathbf{W})) = \frac{tr(\mathbf{W}^3 \Sigma)}{3} - tr(\mathbf{W}). \quad (3)$$

Where Σ is the covariance matrix. The first derivative of (3) is computed as follows [11]:

$$\frac{\partial E(J(\mathbf{W}))}{\partial \mathbf{W}} = (\mathbf{W}^2 \Sigma + \Sigma \mathbf{W}^2 + \mathbf{W} \Sigma \mathbf{W}) / 3 - \mathbf{I}. \quad (4)$$

If \mathbf{W} is selected such that $\mathbf{W} \Sigma = \Sigma \mathbf{W}$, equating (4) to zero will result in $\mathbf{W} = \Sigma^{-1/2}$. Therefore, $\Sigma^{-1/2}$ is a critical point (matrix) of (4). The second derivative of $E(J)$ with respect to \mathbf{W} is [11]:

$$\frac{\partial^2 E(J(\mathbf{W}))}{\partial^2 \mathbf{W}} = 2(\mathbf{I} \otimes \Sigma \mathbf{W}) + 2(\Sigma \mathbf{W} \otimes \mathbf{I}) + \mathbf{W} \otimes \Sigma + \Sigma \otimes \mathbf{W}. \quad (5)$$

where it is assumed that \mathbf{W} is symmetric and $\mathbf{W} \Sigma = \Sigma \mathbf{W}$. Substituting $\mathbf{W} = \Sigma^{-1/2}$ in (5) will result in a positive definite matrix. The above analysis implies that if \mathbf{W} is a symmetric matrix satisfying $\mathbf{W} \Sigma = \Sigma \mathbf{W}$, the cost function $J(\mathbf{w})$ will have a minimum that occurs at $\mathbf{W} = \Sigma^{-1/2}$ [11].

Using the gradient descent optimization method [12] we obtained the following adaptive equation for the computation of $\Sigma^{-1/2}$:

$$\begin{aligned} \mathbf{W}_{k+1} &= \mathbf{W}_k + \eta \left(-\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} \right) \\ &= \mathbf{W}_k + \eta (I - (\mathbf{W}_k^2 \mathbf{x}_{k+1} \mathbf{x}_{k+1}^t + \mathbf{x}_{k+1} \mathbf{x}_{k+1}^t \mathbf{W}_k^2 + \mathbf{W}_k \mathbf{x}_{k+1} \mathbf{x}_{k+1}^t \mathbf{W}_k) / 3) \end{aligned} \quad (6)$$

where \mathbf{W}_{k+1} is the estimation of $\Sigma^{-1/2}$ in $k+1$ -th iteration, η is the step size and \mathbf{x}_{k+1} is the input vector at iteration $k+1$. The only constraint on (6) is its initial conditions, that is \mathbf{W}_0 must be a symmetric and positive definite matrix satisfying $\mathbf{W}_0 \Sigma = \Sigma \mathbf{W}_0$. It is quite easy to prove that if \mathbf{W}_0 is a symmetric and positive definite matrix, then all values of \mathbf{W}_k ($k=1, 2, \dots$) will be symmetric and positive definite. Therefore, the final estimation also will have these properties. To avoid confusion for choosing the initial value \mathbf{W}_0 , we consider \mathbf{W}_0 equal to identity matrix multiplied by a positive constant α ($\mathbf{W}_0 = \alpha \mathbf{I}$).

3.2 Reduction of Computational Cost

As mentioned above, we consider the initial condition equal to identity matrix multiplied by a constant. It is clear that for this initial condition we have $\mathbf{W}_0 \Sigma = \Sigma \mathbf{W}_0$; hence the expected value of (6) is equal to:

$$E(\mathbf{W}_{k+1}) = \mathbf{W}_k + \eta_k (\mathbf{I} - (\mathbf{W}_k^2 \boldsymbol{\Sigma} + \mathbf{W}_k \boldsymbol{\Sigma} \mathbf{W}_k + \boldsymbol{\Sigma} \mathbf{W}_k^2) / 3) \quad (7)$$

It is quite easy to prove that if $\mathbf{W}_0 \boldsymbol{\Sigma} = \boldsymbol{\Sigma} \mathbf{W}_0$, then we will obtain:

$$E(\mathbf{W}_{k+1}) = \mathbf{W}_k + \eta_k (\mathbf{I} - \mathbf{W}_k^2 \boldsymbol{\Sigma}) = \mathbf{W}_k + \eta_k (\mathbf{I} - \mathbf{W}_k \boldsymbol{\Sigma} \mathbf{W}_k) = \mathbf{W}_k + \eta_k (\mathbf{I} - \boldsymbol{\Sigma} \mathbf{W}_k^2) \quad (8)$$

Therefore (6) is simplified to three more efficient forms as follows:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k (\mathbf{I} - \mathbf{W}_k^2 \mathbf{x}_{k+1} \mathbf{x}_{k+1}^t) \quad (9)$$

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k (\mathbf{I} - \mathbf{W}_k \mathbf{x}_{k+1} \mathbf{x}_{k+1}^t \mathbf{W}_k) \quad (10)$$

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k (\mathbf{I} - \mathbf{x}_{k+1} \mathbf{x}_{k+1}^t \mathbf{W}_k^2) \quad (11)$$

Equations (9-11) have less computational cost with respect to (6). Obviously, the expected values of \mathbf{W}_k as $k \rightarrow \infty$ in (6) and (9-11) are equal to $\boldsymbol{\Sigma}^{-1/2}$, provided that $\mathbf{W}_0 \boldsymbol{\Sigma} = \boldsymbol{\Sigma} \mathbf{W}_0$.

3.3 Adaptive Computation of Eigenvectors

We use the following algorithm for the computation of eigenvectors:

$$\mathbf{T}_{k+1} = \mathbf{T}_k + \gamma_k (\mathbf{y}_k \mathbf{x}_k^t - LT[\mathbf{y}_k \mathbf{y}_k^t] \mathbf{T}_k) \quad (12)$$

where $\mathbf{y}_k = \mathbf{T}_k \mathbf{x}_k$ and \mathbf{T}_k is a $p \times n$ matrix that converges to a matrix \mathbf{T} whose rows are the first p eigenvectors of $\boldsymbol{\Sigma}$. $LT[.]$ sets all entries of its matrix argument which are above the diagonal to zero and γ_k is learning rate which meets Ljung's conditions [13]. The convergence of this algorithm has been proved by Sanger [10] using stochastic approximation theory. It has been shown that algorithm (12) computes the eigenvectors of the covariance matrix corresponding to its eigenvalues in descending order. Therefore, choosing initial value as a random $p \times n$ matrix, algorithm (12) will converge to a matrix \mathbf{T} whose rows are the first p eigenvectors of covariance matrix, ordered by decreasing eigenvalues.

There are different adaptive estimations of the mean vector. The following equation was used in [6, 7]:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \eta_{k+1} (\mathbf{x}_{k+1} - \mathbf{m}_k) \quad (13)$$

where η_{k+1} satisfies Ljung assumptions [13].

3.4 New Adaptive LDA Algorithm

As discussed in section 2, the LDA features are significant eigenvectors of $\Sigma_w^{-1}\Sigma$. For adaptive computation of them, we combine two algorithms discussed in the previous sub-sections in cascade and show that this architecture asymptotically computes LDA features. Consider the training sequence described at the beginning of section 2. Furthermore, let \mathbf{m}_k^i denote the estimated mean vector of class $i(i=1,2,\dots,L)$ at k -th iteration and $\omega(\mathbf{x}_k)$ denote the class of \mathbf{x}_k . The training sequence $\{\mathbf{y}_k\}$ for $\Sigma^{-1/2}$ algorithm is defined by $\mathbf{y}_k = \mathbf{x}_k - \mathbf{m}_k^{\omega(\mathbf{x}_k)}$. With the arrival of every training sample \mathbf{x}_k , \mathbf{m}_k^i is updated according to its class using (13). It is easy to show that the correlation of the sequence $\{\mathbf{y}_k\}$ is the within-class scatter matrix Σ_w . Therefore, we have the following equation:

$$\lim_{k \rightarrow \infty} E[(\mathbf{x}_k - \mathbf{m}_k^{\omega(\mathbf{x}_k)})(\mathbf{x}_k - \mathbf{m}_k^{\omega(\mathbf{x}_k)})^t] = \lim_{k \rightarrow \infty} E[\mathbf{y}_k \mathbf{y}_k^t] = \Sigma_w. \quad (14)$$

Suppose the sequence $\{\mathbf{z}_k\}$ is defined by, $\mathbf{z}_k = \mathbf{x}_k - \mathbf{m}_k$. Where \mathbf{m}_k is the estimated mixture mean value in k -th iteration. We train the $\Sigma^{-1/2}$ algorithm by the sequence $\{\mathbf{y}_k\}$ and use \mathbf{W}_k in (8-11) to create the new sequence $\{\mathbf{u}_k\}$ as follows, $\mathbf{u}_k = \mathbf{W}_k \mathbf{z}_k$. The sequence $\{\mathbf{u}_k\}$ is used to train the algorithm (12). As mentioned before, the matrix \mathbf{T} in the algorithm (12) converges to the eigenvectors of the covariance matrix of the input vectors, ordered by decreasing eigenvalues. Hence, (12) will converge to the eigenvectors of $E(\mathbf{u}_k \mathbf{u}_k^t)$. It is quite easy to show:

$$\lim_{k \rightarrow \infty} E(\mathbf{u}_k \mathbf{u}_k^t) = \Sigma_w^{-1/2} \Sigma \Sigma_w^{-1/2}. \quad (15)$$

Our aim is to estimate the eigenvectors of $\Sigma_w^{-1}\Sigma$. Suppose Φ and Λ denote the eigenvector and eigenvalue matrices corresponding to $\Sigma_w^{-1}\Sigma$. Following equations are held [1]:

$$\Sigma_w^{-1}\Sigma\Phi = \Phi\Lambda, \quad \Sigma_w^{-1/2}\Sigma\Sigma_w^{-1/2}\Psi = \Psi\Lambda. \quad (16)$$

where $\Psi = \Sigma_w^{1/2}\Phi$. From (16), it is concluded that the eigenvector matrix of $\Sigma_w^{-1/2}\Sigma\Sigma_w^{-1/2}$ is equal to Ψ . In the other words, the matrix \mathbf{T}^t in the second algorithm converges to Ψ and the following equation is held:

$$\lim_{k \rightarrow \infty} \mathbf{T}_k^t = \Psi = \Sigma_w^{1/2}\Phi^t. \quad (17)$$

By multiplying the outputs of the first and second algorithms as $k \rightarrow \infty$, we will have:

and actual $\Sigma^{-1/2}$ matrices. Fig.2 shows values of the error during iterations for each covariance matrix. Final values of the error after 500 samples are 0.1755 for $d=10$, 0.1183 for $d=8$, 0.1045 for $d=6$ and 0.0560 for $d=4$. As expected, the simulation results confirmed the convergence of (6) toward $\Sigma^{-1/2}$. We repeated the same experiment for (9-11) and in all of experiments and get the same results.

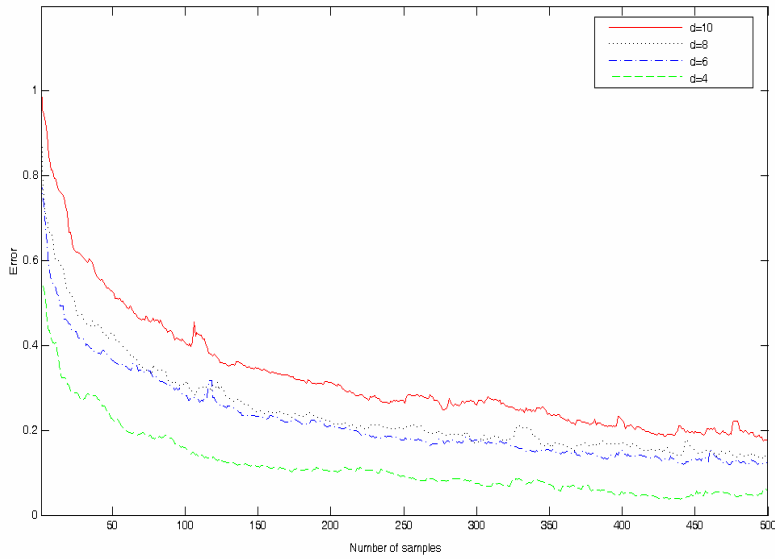


Fig.2 Convergence of $\Sigma^{-1/2}$ algorithm toward its final value, for different covariance matrices.

4.2 Experiments on Adaptive LDA Algorithm

We tested the performance of the new ALDA using *i*) ten dimensional five class Gaussian data and *ii*) PIE database.

4.2.1 Experiment with Ten Dimensional Data

For this purpose, we generated 500 samples of 10-D Gaussian data, each from five classes with different mean vectors and covariance matrices. The means and covariances were obtained from [14] with the covariance matrices multiplied by 20. The eigenvalues of $\Sigma_w^{-1}\Sigma_b$ are 10.84, 7.01, 0.98, 0.34, 0, 0, 0, 0, 0, 0. Thus, the data has intrinsic dimensionality of four for classification, of which only two features corresponding to the eigenvalues 10.84 and 7.01 are significant. We used the proposed ALDA to extract relevant features for classification and compared these

features with their actual values computed from samples scatter matrices. The graph in the left side of Fig. 3 shows the convergence of the first algorithm. As mentioned before through this algorithm, \mathbf{W}_k converges to the square root of the inverse within-class scatter matrix. The graph in the right side of Fig. 3 illustrates the convergence of the first and second feature vectors of $\Sigma_w^{-1}\Sigma$ corresponding to the largest eigenvalues.

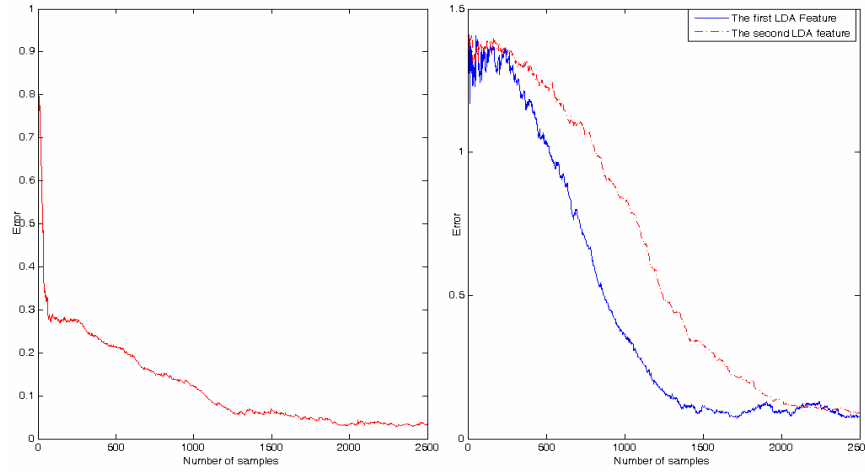


Fig. 3 Left: Convergence of the first algorithm toward the square root of the inverse within-class scatter matrix, Right: Convergence of the estimated first and second LDA features toward their final values.

Normalized error E_φ is defined as $E_\varphi = \|\varphi_i - \hat{\varphi}_i\| / \|\varphi_i\|, i=1,2$, where φ is computed from the sample scatter matrices and $\hat{\varphi}$ is estimated using the proposed ALDA. It can be observed that the feature vectors computed by the new adaptive LDA algorithm converge to their actual values through the training process. The normalized errors at the end of 2500 samples are $E_{\varphi_1} = 0.0724$, $E_{\varphi_2} = 0.0891$. Figure 4 illustrates the distribution of samples during the training process. The graph in the top left side of Fig. 4 illustrates the distribution of training data on estimated LDA feature space after 500 iterations, the top right graph in fig.4 demonstrate the distribution of samples on estimated LDA feature sub-space after 1000 iterations. The left below and right below graphs in fig. 4 shows the distribution of samples on estimated LDA feature space after 1500 and 2500 iterations, respectively. It is obvious that the distribution of data is not clearly separable at first iterations; however by training the algorithm, they separated into five clusters (although overlapping) with only two significant feature vectors. Fig. 4 verifies ability of proposed algorithm for adaptive dimension reduction while preserving separability.

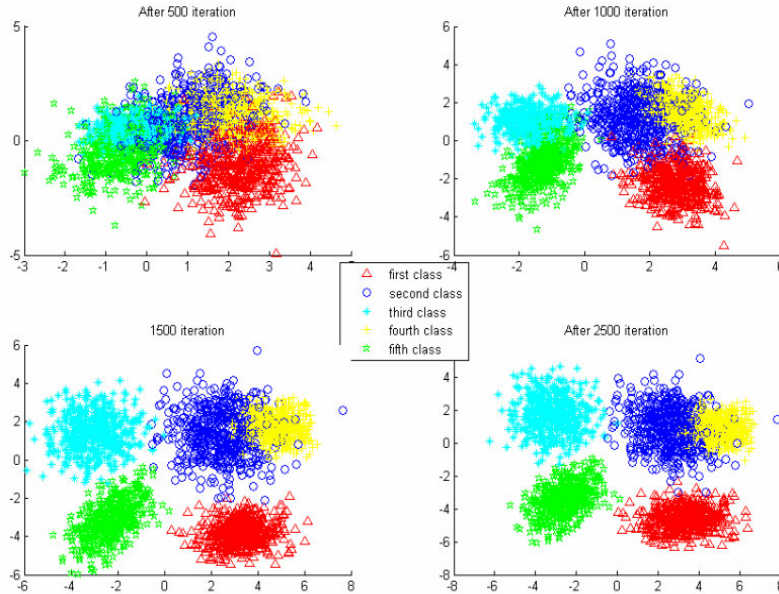


Fig. 4 top-Left: Distribution of data on the estimated LDA sub-space after 500 iterations. Top-right after 1000 iterations. Down left: after 1500 iteration. Down right: after 2500 iteration.

4.2.2 Experiment on PIE data base

This database contains images of 68 people under different poses and illuminations with 4 different expressions. In this experiment, we chose 3 random subjects and for each subject 150 images are considered. We manually cropped all images to size of 40×40 in order to omit the background. Figure 5 shows some of selected subjects in different position and illumination. We vectorized these images (every image produce a 1600×1 vector) and considered them as a sequence of data. Prior to our algorithm, we applied PCA algorithm on the training images and considering the 60 important eigen-faces, we reduced the vector sizes to 60. We trained the proposed algorithm with this sequence of images and reduced the dimensionality of the feature space into three. Figure 6 shows estimated fisher faces [15] at the end of process. Hence there are 3 subjects, the adaptive algorithm will estimate the two fisher faces. Figure 6 shows distribution of images related to each subject in the three dimensional feature space. The top left diagram shows distribution of subject images in feature space after 200, 300 and 450 iteration, respectively. it is clear from figure 6 that images at first iterations are not clearly separable but gradually by training of the algorithm, each subjects separate from others and at the end of process (after 450 iteration) all of the subjects are linearly separable (although overlapping) in three dimensional estimated feature space

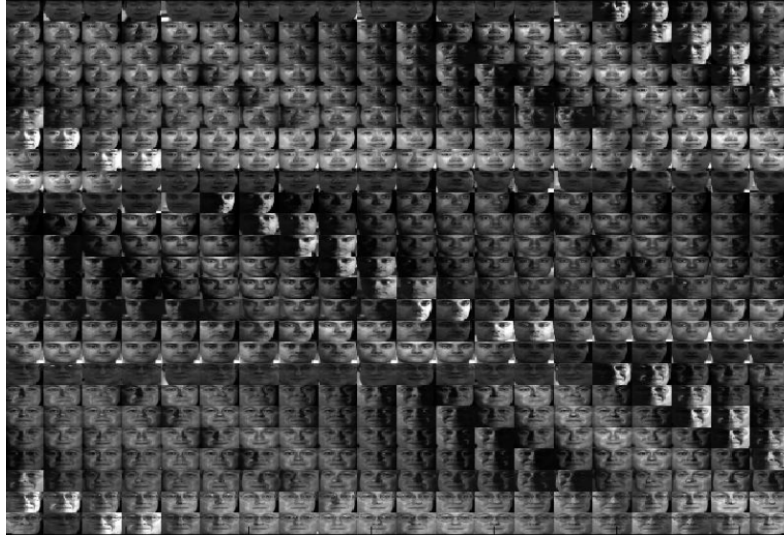


Fig. 5 Sample images from five subjects in different illumination and poses.

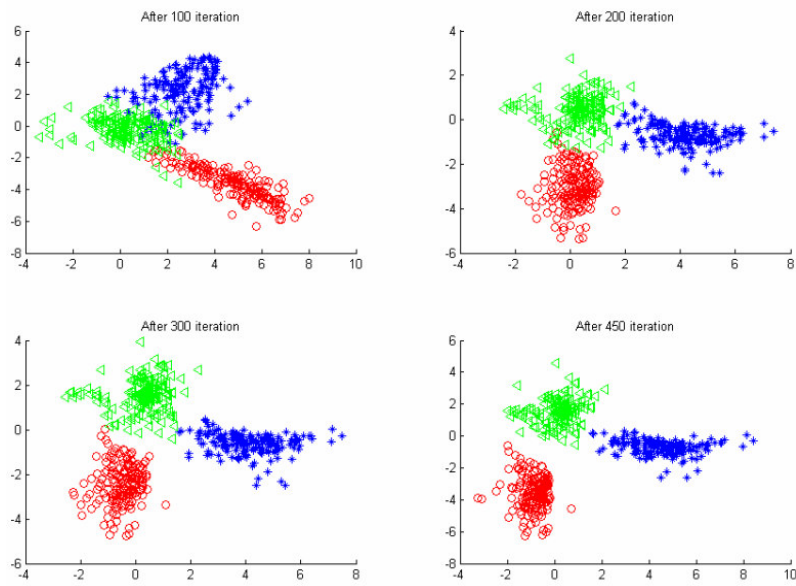


Fig. 6 Distribution of subject images in the estimated three dimensional feature space, after 100, 200, 300 and 450 iteration.

5. Conclusion Remarks

In this paper, a new ALDA feature extraction algorithm was presented. The new algorithm was considered as a combination of a new adaptive $\Sigma^{-1/2}$ algorithm in cascade with APCA. Convergence of the new adaptive algorithms was proved. Simulation results for LDA feature extraction using synthetic and real multidimensional data demonstrated the ability of the proposed algorithm for adaptive optimal feature extraction. The new adaptive algorithm can be used in many fields of on-line pattern recognition applications such as face and gesture recognition.

Acknowledgment. This project was partially supported by Iranian telecommunication research center (ITRC).

References

1. K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, New York, 1990.
2. L. Chen, H.M. Liao, M. Ko, J. Lin, G. Yu, A new LDA based face recognition system which can solve the small sample size problem, Pattern Recognition., vol. 33, no. 10, pp. 1713-1726, 2000.
3. Chellappa, R., Wilson, C., Sirohey, Human and machine recognition of faces, Proc. IEEE Vol. 83 no.5, pp 705-740, 1995.
4. H.Yu and J.Yang, A direct LDA algorithm for high-dimensional data with application to face recognition, Pattern Recog. , vol. 34, no. 10, pp. 2067-2070, 2001.
5. J. Mao and A.K. Jain, Discriminant analysis neural networks, In IEEE Int. Conf. on Neural Networks, CA, pp. 300-305, 1993.
6. C. Chatterjee, V. P. Roychowdhury, On self-organizing algorithm and networks for class separability features, IEEE Trans. Neural Network, Vol. 8, No. 3, pp. 663-678, 1997.
7. H. Abrishami Moghaddam, M. Matinfar, S.M. Sajad Sadough, Kh. Amiri Zadeh, Algorithms and networks for accelerated convergence of adaptive LDA, Pattern Recognition. , Vol. 38, No. 4, pp. 473-483, 2005.
8. H. Hongo, N. Yasumoto, Y. Niva, K. Yamamoto, Hierarchical face recognition using an adaptive discriminant space Proc. IEEE Int. Conf. Computers Communications, Control and Power Engineering (TENCON'02), Vol. 1, 523-528, 2002.
9. Y. Rao, .N. Principe, J.C. Wong, Fast RLS like algorithm for generalized eigen decomposition and its applications, Journal of VLSI Signal processing systems, vol. 37, no. 3, pp. 333-344, 2004.
10. T.D. Sanger, optimal unsupervised learning in a single-layer linear feed forward neural network, Neural Networks, Vol. 2, pp. 459-473, 1989.
11. J.R. Magnus, H. Neudecker, Matrix Differential Calculus, John Wiley, 1999.
12. B.Widrow, S. Stearns, Adaptive Signal Processing, Prentice-Hall, 1985.
13. L. Ljung, Analysis of recursive stochastic algorithms, IEEE Trans. Automat Control, Vol. 22, pp. 551-575, 1977.
14. T. Okada, S.Tomita, An Optimal orthonormal system for discriminant analysis, Pattern Recognition, Vol. 18, No.2, pp. 139-144, 1985.
15. P.N. Belhumeur, J.P. Hespanha, and D. J. Kriegman, Eigenfaces vs. Fisher faces: Recognition using class specific linear projection, IEEE Trans. Pattern Anal. Machine Intel. vol. 19, pp. 711-720, may 1997.