Pattern Recognition Letters 70 (2016) 73–79

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Fast adaptive algorithms for optimal feature extraction from Gaussian data☆

Youness Aliyari Ghassabeh[a,b,*], Frank Rudzicz[a,b], Hamid Abrishami Moghaddam[c]

[a] Toronto Rehabilitation Institute - UHN 550 University Avenue, Toronto, Ontario, Canada
[b] Department of Computer Science, University of Toronto, Toronto, Ontario, Canada
[c] Department of Electrical Engineering, K.N.Toosi University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 18 April 2015
Available online 28 November 2015

Keywords:
Gaussian sequence
Discriminant function
Optimal feature
Accelerated adaptive algorithm

ABSTRACT

We present a new adaptive algorithm to accelerate optimal feature extraction from a sequence of multi-class Gaussian data in order to classify them based on the Bayes decision rule. The optimal Gaussian feature extraction, in the Bayes sense, involves estimation of the square root of the inverse of the covariance matrix, $\mathbf{\Sigma}^{-1/2}$. We use an appropriate cost function to find the optimal step size in each iteration, in order to accelerate the convergence rate of the previously proposed algorithm for adaptive estimation of $\mathbf{\Sigma}^{-1/2}$. The performance of the proposed accelerated algorithm is compared with other adaptive $\mathbf{\Sigma}^{-1/2}$ algorithms. The proposed algorithm is tested for Gaussian feature extraction from three classes of three-dimensional Gaussian data. Simulation results confirm the effectiveness of the proposed algorithm for adaptive optimal feature extraction from a sequence of Gaussian data.

1. Introduction

Bayes' decision rule is a simple statistical approach to the problem of pattern classification. It assumes that the decision problem is given in probabilistic terms, and all of the required probabilities are known in advance [9]. The Bayes decision rule compares the posterior probabilities and assigns the observed sample to the class with the highest one. Based on the Bayes decision rule, when the prior probabilities and the conditional distribution of data given the class label (i.e., the likelihood) are known, an explicit discriminant function can be computed for the classification task [11].

The Gaussian (or normal) distribution is probably the most widely used distribution in the real world applications. It has been used to model the frequency distribution of measurement error [12]. The errors are assumed to be independent of each other and normally distributed with mean zero and variance σ^2. Later, it was found that many real worlds observation are normally distributed, or very close to it. For example, the distribution of human heights, weights, or intelligent quotients can often be approximated by normal distributions [6]. Furthermore, when a measurement is like the sum of independent and identically distributed random variables with equal influence on the result, then the central limit theorem justifies the use of a the Gaussian distribution to model the distribution of the result [10].

The typical estimation of the discriminant functions, when the observed data are generated by different Gaussian sources, requires that all samples are available in advance. However, there are situations where the entire data set is not available and the input data are observed as a stream. In this case, it is desirable for discriminant functions to have the ability to update themselves by observing the new samples without running the algorithm on the whole data set. To address this issue, Chatterjee and Roychowdhury proposed an adaptive algorithm and associated network for optimal feature extraction from a sequence of Gaussian observations [8]. Later, Aliyari and Moghaddam presented an appropriate cost function and showed that optimizing the cost function using the gradient descent method will lead to the optimal Gaussian features [1,2]. The proposed methods in [2,8] suffers from a low convergence rate. In this paper, we present a fast version of the algorithm in [2] by finding the optimal step size in each iteration. The optimal step size is computed by taking the derivative of an appropriate cost function with respect to the step size in each iteration. In the next section, a brief review on optimal feature extraction from Gaussian data is given. The new fast adaptive algorithm is presented in Section 3. Simulation results to confirm the effectiveness

☆ This paper has been recommended for acceptance by Dr. J. Yang

* Corresponding author at:Toronto Rehabilitation Institute - UHN 550 University Avenue, Toronto, Ontario, Canada. Tel.: +1 343 333 4863, +1 416 597 3422X7879 ; fax: +1 416 597 3031.

E-mail address: aliyari@cs.toronto.edu, aliyari@mast.queensu.ca, youness.aliyari@gmail.com (Y. Aliyari Ghassabeh).

http://dx.doi.org/10.1016/j.patrec.2015.11.021
0167-8655/

of the proposed method are given in Section 4. Section 5 is devoted to the concluding remarks.

2. Optimal feature extraction from Gaussian data

Let $\boldsymbol{x} \in \mathbb{R}^D$ be a random vector which belongs to one of K different classes, $\omega_1, \ldots, \omega_K$ with a prior probability of $P(\omega_i), i = 1, \ldots, K$. According to the Bayes classification rule and using *a posteriori* probabilities, $P(\omega_i|\boldsymbol{x}), i = 1, \ldots, K$, a random vector $\boldsymbol{x}$ is assigned to class ω_i if and only if [11]

$$P(\omega_i|\boldsymbol{x}) > P(\omega_j|\boldsymbol{x}), j = 1, \ldots, K \text{and} j \neq i. \tag{1}$$

That means the above K *a posteriori* probabilities are sufficient statistics in the Bayes sense to determine the class of a random vector $\boldsymbol{x}$. Using the Bayes theorem for conditional probabilities and taking the logarithm of both sides we obtain

$$\ln P(\omega_i|\boldsymbol{x}) = \ln p(\boldsymbol{x}|\omega_i) + \ln P(\omega_i) - \ln p(\boldsymbol{x}), i = 1, \ldots, K. \tag{2}$$

Assuming that all the prior probabilities, $P(\omega_i), i = 1, \ldots, K$, are equal and since $p(\boldsymbol{x})$ is the common term in all K identities in (2), then values of $\ln p(\boldsymbol{x}|\omega_i), i = 1, \ldots, K$ can be used for classification of $\boldsymbol{x}$. Assuming that $p(\boldsymbol{x}|\omega_i)$ is a unimodal Gaussian distribution, we get

$$\ln p(\boldsymbol{x}|\omega_i) = -(\boldsymbol{x} - \boldsymbol{m}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{m}_i)/2 - \ln\left((2\pi)^{D/2} |\boldsymbol{\Sigma}_i|^{1/2}\right),$$
$$i = 1, \ldots, K, \tag{3}$$

where $\boldsymbol{m}_i$ and $\boldsymbol{\Sigma}_i$ denote the mean vector and the covariance matrix of class $\omega_i, i = 1, \ldots, K$, respectively. The sufficient information for the classification of the Gaussian data with the minimum Bayes error can be given by discriminant functions $g_i(\boldsymbol{x}), i = 1, \ldots, K$ as follows [16]

$$g_i(\boldsymbol{x}) = -\|\boldsymbol{\Sigma}_i^{-1/2}(\boldsymbol{x} - \boldsymbol{m}_i)\|^2 - \ln(|\boldsymbol{\Sigma}_i|), i = 1, \ldots, K. \tag{4}$$

In other words, the input vector $\boldsymbol{x}$ is assigned to class ω_i if and only if $g_i(\boldsymbol{x}) > g_j(\boldsymbol{x})$ for $j = 1, \ldots, K, j \neq i$. From the above discussion, it can be observed that the discriminant functions $g_i(\boldsymbol{x}), i = 1, \ldots, K$ are sufficient information for the classification of Gaussian data with the minimum Bayes error. That means the new observed vector $\boldsymbol{x}$ will be assigned to the class with the greatest $g_i(\boldsymbol{x})$. For an online application, the parameters of the Gaussian data, $\boldsymbol{\Sigma}_i$ and $\boldsymbol{m}_i, i = 1, \ldots, K$, are unknown in advance and computed during the process using incoming sequence of data samples. Therefore, adaptive estimation of $\boldsymbol{\Sigma}_i^{-1/2}$, $\boldsymbol{\Sigma}_i$, and $\boldsymbol{m}_i$ are highly necessary to compute the discriminant functions, $g_i(\boldsymbol{x})$, for online Gaussian data classification.

3. New fast adaptive algorithm for optimal feature extraction from Gaussian data

The following algorithms for adaptive estimation of the mean vector, $\boldsymbol{m}$, and the covariance matrix, $\boldsymbol{\Sigma}$ for a sequence $\{\boldsymbol{x}_k\}_{k=1,2,\ldots}$ are given

- Mean vector $\boldsymbol{m}$ [13]

$$\boldsymbol{m}_{k+1} = \boldsymbol{m}_k + \alpha_k(\boldsymbol{x}_{k+1} - \boldsymbol{m}_k), \tag{5}$$

  where $\boldsymbol{m}_{k+1}$ is the current estimate of the mean vector (after observing $k+1$ samples) and α_k is the step size that satisfies certain conditions [8]. Note that for a stationary sequence, we can simply set $\alpha_k = 1/(k+1)$, which yields to the following well-known equation [14]

$$\boldsymbol{m}_{k+1} = \frac{k}{k+1}\boldsymbol{m}_k + \frac{1}{k+1}\boldsymbol{x}_{k+1}. \tag{6}$$

- Covariance matrix $\boldsymbol{\Sigma}$ [7]

$$\boldsymbol{\Sigma}_{k+1} = \left(1 - \beta\frac{k}{k+1}\right)(\boldsymbol{x}_{k+1} - \boldsymbol{m}_{k+1})(\boldsymbol{x}_{k+1} - \boldsymbol{m}_{k+1})^T + \beta\frac{k}{k+1}\boldsymbol{\Sigma}_k, \tag{7}$$

where $\boldsymbol{\Sigma}_{k+1}$ is the current estimate of the covariance matrix (after observing $k+1$ samples), $\boldsymbol{m}_{k+1}$ is the mean estimate after $k+1$ observation (computed using (6)), and $\beta \in (0, 1]$ is called the forgetting factor [14]. If the sequence $\{\boldsymbol{x}_k\}$ is generated by a stationary process, we set $\beta = 1$ [1]. For a non-stationary process, we set $0 < \beta < 1$ to implement an effective window of size $1/(1-\beta)$ [7]. This effective window ensures that the past data samples are down-weighted with an exponentially fading window. The exact value of β depends on the specific application. In general for slow time-varying $\{\boldsymbol{x}_k\}_{k=1,2,\ldots}$, β is chosen close to one to obtain a large effective window, but for fast time-varying sequences, β is selected near zero for a small effective window [5].

Chatterjee and Roychowdhury proposed the following algorithm for adaptive estimation of $\boldsymbol{\Sigma}^{-1/2}$ [8]

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_k(\boldsymbol{I} - \boldsymbol{W}_k(\boldsymbol{x} - \boldsymbol{m}_k)(\boldsymbol{x} - \boldsymbol{m}_k)^T\boldsymbol{W}_k), \tag{8}$$

where $\boldsymbol{W}_{k+1}$ is the current estimate of $\boldsymbol{\Sigma}^{-1/2}$, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{W}_0 \in \mathbb{R}^{D\times D}$ is a symmetric, and semi-definite matrix, and η_k is the step size. Using stochastic approximation, the authors in [8] proved that, under certain conditions, the sequence $\{\boldsymbol{W}_k\}_{k=0,1,2,\ldots}$ converges to $\boldsymbol{\Sigma}^{-1/2}$ with probability one, i.e., $\lim_{k\to\infty}\boldsymbol{W}_k = \boldsymbol{\Sigma}^{-1/2}$. Later Aliyari and Moghaddam [2] introduced a cost function $J(\boldsymbol{W})$ with the global minimum at $\boldsymbol{\Sigma}^{-1/2}$ and showed that applying the gradient descent method on $J(\boldsymbol{W})$ would give the following adaptive algorithms for computing $\boldsymbol{\Sigma}^{-1/2}$

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_k(\boldsymbol{I} - \boldsymbol{W}_k^2\boldsymbol{\Sigma}), \tag{9}$$

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_k(\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{\Sigma}\boldsymbol{W}_k), \tag{10}$$

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_k(\boldsymbol{I} - \boldsymbol{\Sigma}\boldsymbol{W}_k^2), \tag{11}$$

where $\boldsymbol{W}_{k+1}$ is the $\boldsymbol{\Sigma}^{-1/2}$ estimate after $k+1$ iterations, and η_k is the step size. The proposed cost function $J(\boldsymbol{W}) : \mathcal{C} \to \mathbb{R}$ is given by [3]

$$J(\boldsymbol{W}) = \frac{1}{3}Tr\left[(\boldsymbol{W}\boldsymbol{\Sigma}^{1/2} - \boldsymbol{I})^2(\boldsymbol{W} + 2\boldsymbol{\Sigma}^{-1/2})\right], \tag{12}$$

where $\mathcal{C} \subset \mathbb{R}^{n\times n}$ is the set of all symmetric positive definite matrices $\boldsymbol{W}$ that commute with $\boldsymbol{\Sigma}^{1/2}$, i.e., $\boldsymbol{W}\boldsymbol{\Sigma}^{1/2} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{W}$, $Tr[.]$ is the matrix trace function, and $\boldsymbol{I}$ denotes the identity matrix. By definition, the cost function J($\boldsymbol{W}$) in (12) is one third of the trace of the product of a symmetric semi-positive definite matrix, $(\boldsymbol{W}\boldsymbol{\Sigma}^{1/2} - \boldsymbol{I})^2$, with a symmetric positive definite matrix, $\boldsymbol{W} + 2\boldsymbol{\Sigma}^{-1/2}$. Hence, the cost function itself is a semi-positive definite matrix [4], i.e., $J(\boldsymbol{W}) \geq 0$ for all $\boldsymbol{W} \in \mathcal{C}$. By taking the gradient of the cost function in (12) with respect to $\boldsymbol{W}$ and equating it to zero, we obtain

$$\nabla J(\boldsymbol{W}) = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W} - \boldsymbol{I} = 0. \tag{13}$$

Eq. (13) reveals that, in the domain $\mathcal{C}$, the cost function $J(\boldsymbol{W})$ has a unique stationary point that occurs at $\boldsymbol{\Sigma}^{-1/2}$. Since $J(\boldsymbol{\Sigma}^{-1/2}) = 0$, then the matrix $\boldsymbol{\Sigma}^{-1/2}$ is the unique global minimum of the cost function $J(\boldsymbol{W})$ over the convex set $\mathcal{C}$. Using the gradient descent algorithm to minimize the cost function $J(\boldsymbol{W})$ will lead to the algorithms in (9 – 11) to estimate the global minimum, $\boldsymbol{\Sigma}^{-1/2}$. Since the covariance matrix $\boldsymbol{\Sigma}$ is not known in advance, the authors in [3,4] showed that the covariance matrix can be replaced by its estimate at the kth iteration as follows

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_k(\boldsymbol{I} - \boldsymbol{W}_k^2\boldsymbol{\Sigma}_k), \tag{14}$$

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_k(\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{\Sigma}_k\boldsymbol{W}_k), \tag{15}$$

[1] Note that setting $\beta = 1$ for the stationary data will give the following well-known adaptive estimation of the covariance matrix $\boldsymbol{\Sigma}_{k+1} = \frac{1}{k+1}(\boldsymbol{x}_{k+1} - \boldsymbol{m}_{k+1})(\boldsymbol{x}_{k+1} - \boldsymbol{m}_{k+1})^T + \frac{k}{k+1}\boldsymbol{\Sigma}_k = \frac{1}{k+1}\sum_{i=1}^{k+1}(\boldsymbol{x}_i - \boldsymbol{m}_{k+1})(\boldsymbol{x}_i - \boldsymbol{m}_{k+1})^T$, where the mean value $\boldsymbol{m}_{k+1}$ is estimated using (6). For the non-stationary data, we select β between zero and one, where the proof of the convergence was given in [7].

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_k(\boldsymbol{I} - \boldsymbol{\Sigma}_k \boldsymbol{W}_k^2), \tag{16}$$

where all parameters in the above equations are as before, and $\boldsymbol{\Sigma}_k$ is the estimate of the covariance matrix after k iterations (computed using (7)).

The algorithms presented in [8] and [2] use a fixed or decreasing step size causing a low convergence rate that is not desirable. In this paper, we use the cost function in (12) to find the optimal step size in each iteration in order to accelerate the convergence rate. The optimal step size $\eta_{k,\,opt}$ at the $k+1$th iteration is found by equating the first derivative of the cost function $J(\boldsymbol{W})$ with respect to η_k to zero. By taking the derivative of (12) with respect to the step size η_k, equating to zero, and a few additional operations (see the appendix for details) we get

$$\frac{\partial J(\boldsymbol{W}_{k+1})}{\partial \eta_k} = a\eta_k^2 + b\eta_k + c = 0, \tag{17}$$

where $a = Tr(\boldsymbol{G}_k^3 \boldsymbol{\Sigma})$, $b = 2Tr(\boldsymbol{W}_k \boldsymbol{G}_k^2 \boldsymbol{\Sigma})$, $c = Tr(\boldsymbol{W}_k^2 \boldsymbol{G}_k \boldsymbol{\Sigma}) - Tr(\boldsymbol{G}_k)$, and $\boldsymbol{G}_k = \boldsymbol{I} - \boldsymbol{W}_k \boldsymbol{\Sigma} \boldsymbol{W}_k$. Eq. (17) is a quadratic equation and the roots (the optimal step sizes) are given by

$$\eta_{k,opt} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{18}$$

Since the step size η_k cannot be a negative number, only the root with the positive sign can be considered as an optimal step size and since the covariance matrix $\boldsymbol{\Sigma}$ is not available, it must be replaced by its estimate $\boldsymbol{\Sigma}_{k+1}$ in (7) (as the number of the observed samples increases we get a better estimate of the $\boldsymbol{\Sigma}$). Therefore the optimal step size in each iteration is given by

$$\eta_{k,opt} = \frac{-b_{k+1} + \sqrt{b_{k+1}^2 - 4a_{k+1}c_{k+1}}}{2a_{k+1}}, \tag{19}$$

where $a_{k+1} = Tr(\boldsymbol{G}_k^3 \boldsymbol{\Sigma}_{k+1})$, $b_k = 2Tr(\boldsymbol{W}_k \boldsymbol{G}_k^2 \boldsymbol{\Sigma}_{k+1})$, and $c_{k+1} = Tr(\boldsymbol{W}_k^2 \boldsymbol{G}_k \boldsymbol{\Sigma}_{k+1}) - Tr(\boldsymbol{G}_k)$. Therefore, the accelerated adaptive $\boldsymbol{\Sigma}^{-1/2}$ algorithm has the following equivalent forms

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_{k,opt}(\boldsymbol{I} - \boldsymbol{W}_k^2 \boldsymbol{\Sigma}_{k+1}), \tag{20}$$

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_{k,opt}(\boldsymbol{I} - \boldsymbol{W}_k \boldsymbol{\Sigma}_{k+1} \boldsymbol{W}_k), \tag{21}$$

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \eta_{k,opt}(\boldsymbol{I} - \boldsymbol{\Sigma}_{k+1} \boldsymbol{W}_k^2), \tag{22}$$

where the covariance estimate $\boldsymbol{\Sigma}_{k+1}$ is given in (7) and $\eta_{k,\,opt}$ in each iteration is computed using (19). The proposed accelerated algorithm for adaptive estimation of $\boldsymbol{\Sigma}^{-1/2}$ is summarized in Algorithm 1. Note that we use (19) to update the step size in kth iteration if $b_{k+1}^2 - 4a_{k+1}c_{k+1} \geq 0$ and $\eta_{k,\,opt} > 0$, otherwise the step size η_k remains unchanged, i.e., it is equal to η_{k-1}.

Therefore, the proposed accelerated adaptive computing of the discriminant function, $g(\boldsymbol{x})$, in (4) for a Gaussian sequence involves three steps: (*i*) estimating the mean vector $\boldsymbol{m}$ using either (5) or (6)[2], (*ii*) estimating the covariance matrix using (7), and (*iii*) estimating $\boldsymbol{\Sigma}^{-1/2}$ using (19) and (20–22). The proposed accelerated adaptive feature extraction from Gaussian observations is summarized in Algorithm 2. Note that to classify an arbitrary point $\boldsymbol{x}$, we just need to run Algorithm 2 and evaluate K discrimination functions (where K is the number of classes) at $\boldsymbol{x}$ and assign $\boldsymbol{x}$ to the class with the highest value of the discriminant function.

Algorithm 1: Accelerated $\boldsymbol{\Sigma}^{-1/2}$ algorithm.

Input : $\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \boldsymbol{x}_N$, the data sequence of length N, α_k, β, and η_0

/* The sequence members $\boldsymbol{x}_i, i = 1, \ldots, N$ are given to the $\boldsymbol{\Sigma}^{-1/2}$ algorithm one by one */

/* Note that the sequence of the step size $\{\alpha_k\}_{k=0,1,\ldots}$ is usually considered as a decreasing or fixed sequence */

Output: $\boldsymbol{\Sigma}_e^{-1/2}$, estimate of $\boldsymbol{\Sigma}^{-1/2}$ after observing N random vectors sequentially.

begin

Initialization:: $\boldsymbol{W}_0 = \boldsymbol{I}$;

Initialization:: Initialize the estimated covariance matrix $\boldsymbol{\Sigma}_0$;

Initialization:: Initialize the estimated mean vector $\boldsymbol{m}_0$;

for $i = 1$ *to* N **do**

$\boldsymbol{m}_i = \boldsymbol{m}_{i-1} + \alpha_k(\boldsymbol{x}_i - \boldsymbol{m}_{i-1})$;

/* Update the mean vector */

$$\boldsymbol{\Sigma}_i = (1 - \beta\frac{i-1}{i})(\boldsymbol{x}_i - \boldsymbol{m}_i)(\boldsymbol{x}_i - \boldsymbol{m}_i)^T + \beta\frac{i-1}{i}\boldsymbol{\Sigma}_{i-1},$$

/* Update the estimated covariance matrix using */

Compute the optimal step size using (19), if the result is a real and positive number update the step size $\eta_{i-1,opt}$ using (19), otherwise keep the step size unchanged;

$\boldsymbol{W}_i = \boldsymbol{W}_{i-1} + \eta_{i-1,opt}(\boldsymbol{I} - \boldsymbol{W}_{i-1}\boldsymbol{\Sigma}_i \boldsymbol{W}_{i-1})$;

/* Note that we can use any of equations in (20)-(22) */

end

$\boldsymbol{\Sigma}_e^{-1/2} = \boldsymbol{W}_N$

end

Algorithm 2: Computing the discriminant functions $g_i(\boldsymbol{x}), i = 1, 2, \ldots, K$ at an arbitrary point $\boldsymbol{x}$.

Input : $\boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \ldots$ the Gaussian data from ith class, $i = 1, 2, \ldots, K$

/* Here we have K classes and the input training data $\{\boldsymbol{x}_j\}_{j=1,2,\ldots}$ is observed as a sequence with unknown mean and covariance */

Output: The discriminant function $g_i(\boldsymbol{x}), i = 1, 2, \ldots, K$

begin

for $i = 1$ *to* K **do**

Initialization:: $\boldsymbol{W}_0 = \boldsymbol{I}$, $\boldsymbol{\Sigma}_0$, and $\boldsymbol{m}_0$;

/* for simplicity we assigned $\boldsymbol{I}$ to $\boldsymbol{W}_0$ */

while *There is input data from* i^{th} *class* **do**

Update the mean vector $\boldsymbol{m}_k$ using either (5) or (6);

Update the covariance matrix $\boldsymbol{\Sigma}_k$ using (7) ;

Update $\boldsymbol{W}_{k+1}$ using (19) and (20);

Find the optimal step size, η_{opt}, using (19) ;

Update $\boldsymbol{W}_{k+1}$ using (20–22);

/* $\boldsymbol{W}_{k+1}$ is an estimate of $\boldsymbol{\Sigma}^{-1/2}$ for ith class after observing $k+1$ samples */

end

The discriminant function for ith class, $g_i(\boldsymbol{x}), i = 1, 2, \ldots, K$, at an arbitrary point $\boldsymbol{x}$ is given by

$$g_i(\boldsymbol{x}) = -\|\boldsymbol{W}_{N_i}(\boldsymbol{x} - \boldsymbol{m}_{N_i})\|^2 - \ln(|\boldsymbol{\Sigma}_{N_i}|), \tag{23}$$

/* where N_i is the number of the observed samples from class i */

end

end

[2] Eq. (5)is used for anon-stationary data sequence and Eq. (6) is used for a stationary data sequence.

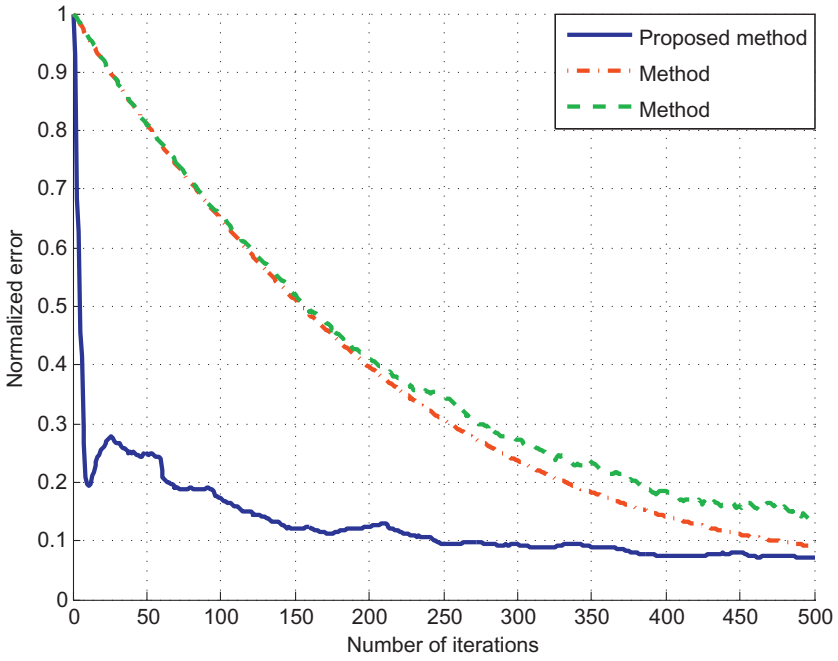

Fig. 1. Comparison between convergence rate of the proposed accelerated $\boldsymbol{\Sigma}^{-1/2}$ algorithm and algorithms in [8] (green curve) and [2] (red curve) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

Since the proposed accelerated algorithm finds the optimal step size in each iteration, it requires more computations than the algorithms in [8] and [2], which simply use a constant or decreasing step size. The computational complexity of the proposed algorithm is $\mathcal{O}(n^3)$, while the computational complexity of the aforementioned algorithms is $\mathcal{O}(n^2)$. It should be noted that although the proposed algorithm requires more computations, it converges fast and provides a more accurate estimate of $\boldsymbol{\Sigma}^{-1/2}$ in much fewer iterations comparing with the algorithms in [8] and [2].

4. Simulation results

We first show the effectiveness of the proposed algorithm for accelerated estimation of $\boldsymbol{\Sigma}^{-1/2}$, and then we use it for optimal feature extraction from multidimensional Gaussian data. For all simulations, it is assumed that the whole data set is not available in advance and the input data are fed to the proposed algorithm sequentially.

4.1. Accelerate adaptive computing of $\boldsymbol{\Sigma}^{-1/2}$

For simulation in this section, we use the first covariance matrix in [15] which is a 10×10 covariance matrix. The input sequence $\{\boldsymbol{x}_k \in \mathbb{R}^{10}\}_{k=1,2,\ldots}$ is generated from a zero mean 10-dimensional Gaussian distribution with the covariance matrix in (24).

The ten eigenvalues of this matrix in descending order are 5.90, 2.782, 1.709, 1.029, 0.394, 0.293, 0.087, 0.071, 0.061, and 0.05. We generated 1000 samples of 10-dimensional, zero-mean Gaussian data with the covariance matrix in (24). The performance of the proposed algorithm to estimate $\boldsymbol{\Sigma}^{-1/2}$ is compared with the algorithms in [8] and [2]. The proposed $\boldsymbol{\Sigma}^{-1/2}$ algorithm, the algorithms in [8], and [2] are initialized to be the identity matrix, i.e. $\boldsymbol{W}_0 = \boldsymbol{I}$. The initial step size, η_0, for all three algorithm is set to 0.01. For the algorithms in [8] and [2] we use a decreasing step size, but the proposed algorithm starts with $\eta_0 = 0.01$ and finds the optimal step size in each iteration using (19). The error at the kth iteration between the estimated and the actual $\boldsymbol{\Sigma}^{-1/2}$ matrices is computed by

$$e_k = \sqrt{\sum_{i=1}^{10}\sum_{j=1}^{10}\left(\boldsymbol{W}_k(i,j) - \boldsymbol{\Sigma}^{-1/2}(i,j)\right)^2},$$

where $\boldsymbol{W}_k(i,j)$ and $\boldsymbol{\Sigma}^{-1/2}(i,j)$ represent the ijth element of the estimated square root of the inverse covariance matrix at the kth iteration and the ijth element of the actual square root of the inverse covariance matrix, respectively. Fig. 1 compares the normalized error for estimating $\boldsymbol{\Sigma}^{1/2}$ resulting from the proposed accelerated algorithm and the algorithms in [8] and [2]. It can be observed from Fig. 1 that the proposed algorithm approaches a very low estimation

$$\begin{bmatrix}
0.091 & & & & & & & & & \\
0.038 & 0.373 & & & & & & & & \\
-0.053 & 0.018 & 1.43 & & & & & & & \\
-0.005 & -0.028 & 0.017 & 0.084 & & & & & & \\
0.010 & -0.011 & 0.055 & -0.005 & 0.071 & & & & & \\
-0.136 & -0.367 & -0.450 & 0.016 & 0.088 & 5.72 & & & & \\
0.155 & 0.154 & -0.038 & 0.042 & 0.058 & -0.544 & 2.75 & & & \\
0.030 & -0.057 & -0.298 & -0.022 & -0.069 & -0.248 & -0.343 & 1.45 & & \\
0.002 & -0.031 & -0.041 & 0.001 & -0.008 & 0.005 & -0.011 & 0.078 & 0.067 & \\
0.032 & -0.065 & -0.030 & 0.005 & 0.003 & 0.095 & -0.120 & 0.028 & 0.015 & 0.341
\end{bmatrix} \tag{24}$$

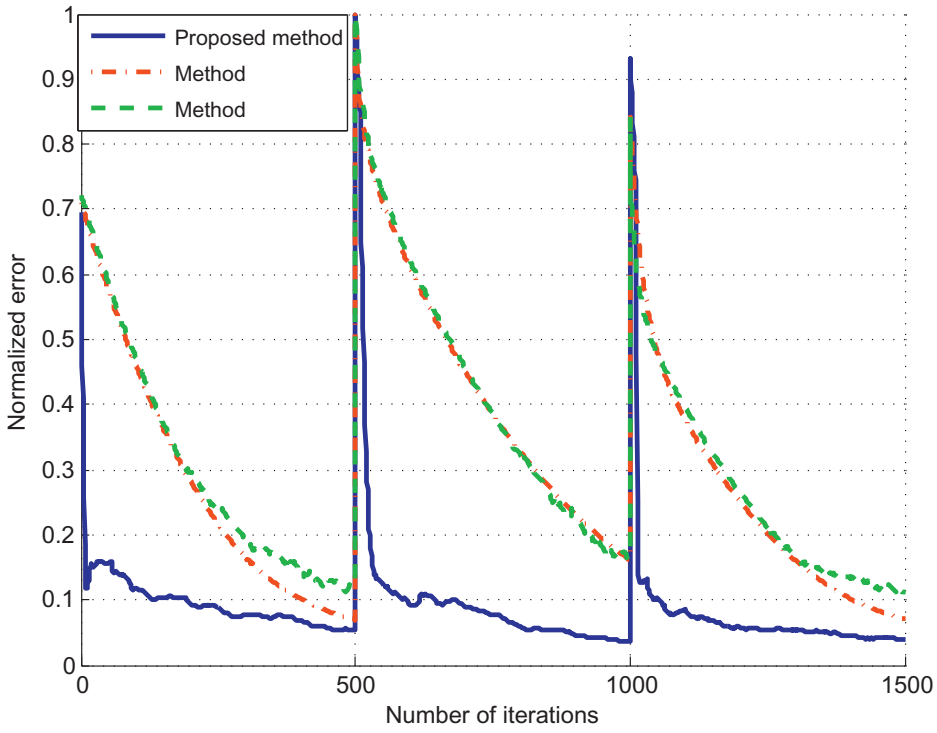

Fig. 2. Tracking ability of the proposed algorithm for non-stationary data. The proposed algorithm provides a lower normalized error in fewer iterations comparing to the algorithms in [8](green curve) and [2](red curve)(For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

error in far fewer iterations compared to the algorithms in [8] and [2]. In order to demonstrate the tracking ability of the proposed accelerated $\boldsymbol{\Sigma}^{-1/2}$ algorithm for non-stationary data, we generate 500 samples of zero-mean Gaussian data with the covariance matrix in (24). Then we drastically change the nature of the data sequence by generating another 500 zero mean 10-dimensional Gaussian data with the second covariance matrix in [15]. Finally, we alter the covariance matrix to the third covariance matrix in [15] and generate another 500 zero-mean Gaussian samples. Fig. 2 demonstrates the normalized errors for estimating $\boldsymbol{\Sigma}^{-1/2}$ resulting from the proposed accelerated algorithm and the algorithms in [8] and [2]. As it is expected when the nature of the observed data changes (e.g., after the 500th sample and the 1000th sample), there are sudden increases in the normalized estimation error. But by observing new incoming samples with the new covariance matrix, the proposed algorithm adapts itself to the new condition and gradually the estimation error decreases. Fig. 2 shows that the proposed accelerated $\boldsymbol{\Sigma}^{-1/2}$ algorithm achieves a lower estimation error under different conditions than the algorithms in [2,8].

We also repeated the above simulation ten times and recorded the number of times that (19) does not generate a positive real valued step size. The mean value and the standard deviation for the number of times that (19) does not produce a positive real valued number were 4.5, and 8.24, respectively. The observation shows that out of 1500 iterations, Eq. (19) may generate very small number of step sizes that are not positive or real valued, where in this situations the step size remains unchanged from the previous iteration.

4.2. Optimal feature extraction from three-dimensional Gaussian data

As mentioned in Section 2, the discriminant function $g_i(\boldsymbol{x})$ in (4) provides enough information (in the Bayes sense) for Gaussian data classification. In other words, for an arbitrary sample $\boldsymbol{x}$, we just need to compute K[3] discriminant functions $g_i(\boldsymbol{x})$ and assign $\boldsymbol{x}$ to the class with the highest $g_i(\boldsymbol{x})$. Since the discriminant functions for the Gaussian data are negative, for simplicity we use the absolute value of the discriminant function as the Gaussian feature $f_i(\boldsymbol{x}), i = 1, 2, 3$, i.e., $f_i(\boldsymbol{x}) = |g_i(\boldsymbol{x})|$, and assign $\boldsymbol{x}$ to the class with the smallest absolute value, i.e., $\boldsymbol{x} \in \omega_i$ if $f_i(\boldsymbol{x}) < f_j(\boldsymbol{x})$ [4] for all $i \neq j$. The input sequence is generated from three Gaussian classes, ω_1, ω_2, and ω_3, with the following parameters

$$\boldsymbol{m}_1 = \begin{bmatrix} -2 \\ 2 \\ 1 \end{bmatrix}, \boldsymbol{\Sigma}_1 = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 0 \\ 1 & 0 & 3 \end{bmatrix}, \boldsymbol{m}_2 = \begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix},$$
$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \boldsymbol{m}_3 = \begin{bmatrix} 5 \\ -5 \\ 5 \end{bmatrix}, \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 2 \\ 1 & 3 & 5 \end{bmatrix}. \quad (25)$$

The mean vectors $\boldsymbol{m}_i, i = 1, 2, 3$ and the covariance matrices $\boldsymbol{\Sigma}_i, i = 1, 2, 3$ are trained using (6) and (7) (β is set to one). The square root of the inverse of the covariance matrix $\boldsymbol{\Sigma}^{-1/2}$ is trained using the proposed algorithms in (20) and the optimal learning rate in each iteration is computed using (19). Finally, the absolute value of the discriminant function, $|g_i(\boldsymbol{x})| = \|\boldsymbol{\Sigma}_i^{-1/2}(\boldsymbol{x} - \boldsymbol{m}_i)\| + \ln(|\boldsymbol{\Sigma}_i|), i = 1, 2, 3$, is computed for each test sample and the observed sample is assigned to the class with the smallest $f_i(\boldsymbol{x}) = |g_i(\boldsymbol{x})|$. For a better visualization of distribution of the Gaussian data in the feature space, we showed the distribution in the feature spaces constructed by $f_1 - f_2$, $f_1 - f_3$, and $f_2 - f_3$ in Fig. 3. From Fig. 3, it can be observed that if an arbitrary sample $\boldsymbol{x}$ belongs to $\omega_i, i = 1, 2, 3$, then $f_i(\boldsymbol{x}) < f_j(\boldsymbol{x}), j = 1, 2, 3$ and $j \neq i$. Therefore, the estimated discriminant functions can be successfully used for the classification of Gaussian observation based on Bayes decision theory.

[3] When the number of classes is K.

[4] That is equivalent to $|g_i(\boldsymbol{x})| < |g_j(\boldsymbol{x})|$.

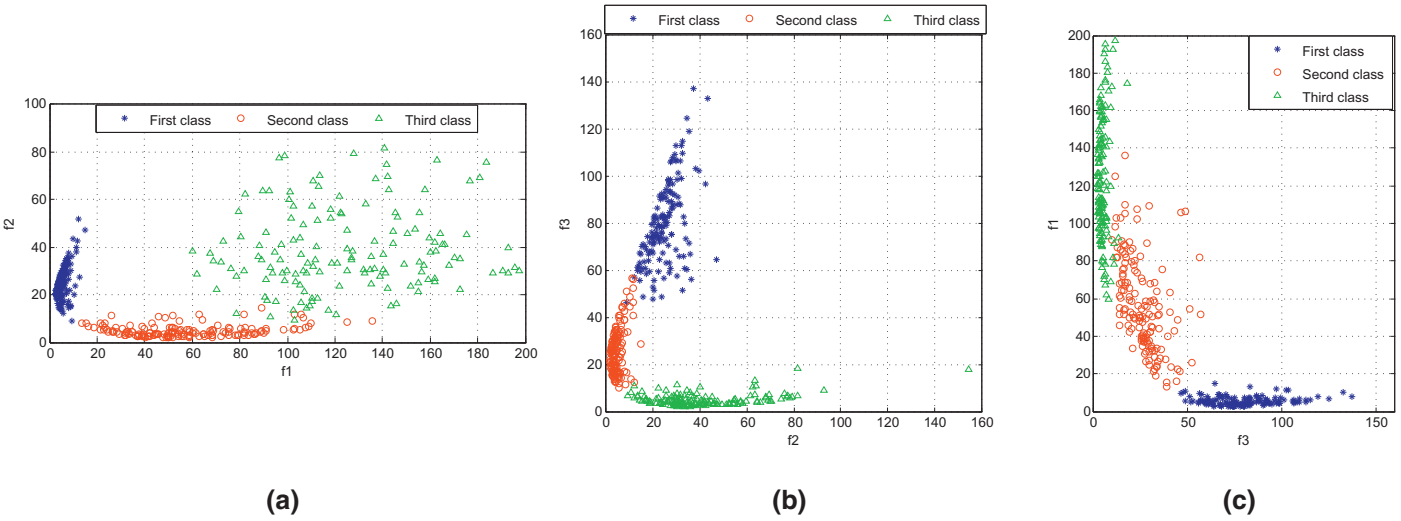

Fig. 3. Distribution of Gaussian data on the feature space: (a) The feature space constructed by f_1 and f_2, (b) The feature space constructed by f_2 and f_3, (c) The feature space constructed by f_3 and f_1.

5. Conclusion

In many real world applications such as mobile robotics or online face recognition the entire data set is not available in advance and the input data are observed as a stream. In these situations, it is desirable for a feature extraction algorithm to have the ability to update the computed features by just observing the new samples without running the algorithm on the whole data set. It is straightforward to show that the optimal features (discriminant functions) for the classification of a sequence of Gaussian data involves adaptive computation of $\mathbf{\Sigma}^{-1/2}$. The previously proposed adaptive algorithms to estimate $\mathbf{\Sigma}^{-1/2}$ use a fixed or decreasing step size that lead to a slow convergence rate. It this paper, we found the optimal step size in each iteration in order to accelerate the convergence of $\mathbf{\Sigma}^{-1/2}$ algorithm. The optimal step size in each iteration is computed by taking the derivative of an appropriate cost function with respect to the step size and equating it to zero. We presented two algorithms, the first algorithm can be used for the fast estimation of $\mathbf{\Sigma}^{-1/2}$ when observing a sequence of data. The second algorithm uses the estimated $\mathbf{\Sigma}^{-1/2}$ for the optimal feature extraction from a sequence of the Gaussian data. Through the simulations, we showed the effectiveness of the proposed accelerated technique to estimate $\mathbf{\Sigma}^{-1/2}$ for both stationary and non-stationary data. We also used the proposed technique successfully for optimal feature extraction from three classes of three-dimensional Gaussian data.

Appendix

*Taking the gradient of the cost function J(**W**) with respect to **W***

First note that if $\boldsymbol{W}$ and $\mathbf{\Sigma}^{1/2}$ commute then $\boldsymbol{W}$ and $\mathbf{\Sigma}$ also commute and we have

$$\boldsymbol{W}\mathbf{\Sigma} = \mathbf{\Sigma}\boldsymbol{W}. \tag{26}$$

By expanding the cost function $J(\boldsymbol{W})$ in (12) and using the commutative property between $\boldsymbol{W}$ and $\mathbf{\Sigma}$, the cost function $J(\boldsymbol{W})$ can be simplified as follows

$$J(\boldsymbol{W}) = \frac{1}{3}Tr(\boldsymbol{W}^3\mathbf{\Sigma}) - Tr(\boldsymbol{W}) + \frac{2}{3}Tr(\mathbf{\Sigma}^{-1/2}). \tag{27}$$

By taking the gradient of $J(\boldsymbol{W})$ in (27) with respect to $\boldsymbol{W}$, we obtain

$$\begin{aligned}\nabla J(\boldsymbol{W}) &= \frac{\mathbf{\Sigma}\boldsymbol{W}^2 + \boldsymbol{W}\mathbf{\Sigma}\boldsymbol{W} + \boldsymbol{W}^2\mathbf{\Sigma}}{3} - \boldsymbol{I} \\ &= \boldsymbol{W}\mathbf{\Sigma}\boldsymbol{W} - \boldsymbol{I},\end{aligned} \tag{28}$$

where $\boldsymbol{I}$ is the identity matrix.

*Taking the derivative of the function J(**W**) with respect to the step size*

By expanding the cost function $J(\boldsymbol{W})$ at $k+1$th iteration, we have

$$\begin{aligned}J(\boldsymbol{W}_{k+1}) &= \frac{1}{3}Tr(\boldsymbol{W}_{k+1}^3\mathbf{\Sigma}) - Tr(\boldsymbol{W}_{k+1}) + \frac{2}{3}Tr(\mathbf{\Sigma}^{-1/2}) \\ &= \frac{1}{3}Tr\Big((\boldsymbol{W}_k + \eta_k\boldsymbol{G}_k)^3\mathbf{\Sigma}\Big) - Tr(\boldsymbol{W}_k + \eta_k\boldsymbol{G}_k) \\ &\quad + \frac{2}{3}Tr(\mathbf{\Sigma}^{-1/2}),\end{aligned} \tag{29}$$

where $\boldsymbol{G}_k = \boldsymbol{I} - \boldsymbol{W}_k\mathbf{\Sigma}\boldsymbol{W}_k$.

The cost function in (29) can be further simplified to

$$\begin{aligned}J(\boldsymbol{W}_{k+1}) &= \frac{Tr(\boldsymbol{W}_k^3\mathbf{\Sigma} + 3\eta_k\boldsymbol{W}_k^2\boldsymbol{G}_k\mathbf{\Sigma} + 3\eta_k^2\boldsymbol{W}_k\boldsymbol{G}_k^2\mathbf{\Sigma} + \eta_k^3\boldsymbol{G}_k^3\mathbf{\Sigma})}{3} \\ &\quad - Tr(\boldsymbol{W}_k + \eta_k\boldsymbol{G}_k) + \frac{2}{3}Tr(\mathbf{\Sigma}^{-1/2}).\end{aligned} \tag{30}$$

By taking the derivative of (30) with respect to the step size η_k and equating it to zero, we obtain

$$\begin{aligned}\frac{\partial J(\boldsymbol{W}_{k+1})}{\partial\eta_k} &= Tr(\boldsymbol{G}_k^3\mathbf{\Sigma})\eta_k^2 + 2Tr(\boldsymbol{W}_k\boldsymbol{G}_k^2\mathbf{\Sigma})\eta_k + Tr(\boldsymbol{W}_k^2\boldsymbol{G}_k\mathbf{\Sigma}) \\ &\quad - Tr(\boldsymbol{G}_k) = a_k\eta_k^2 + b_k\eta_k + c_k = 0,\end{aligned}$$

where $a_k = Tr(\boldsymbol{G}_k^3\mathbf{\Sigma}_{k+1})$, $b_k = 2Tr(\boldsymbol{W}_k\boldsymbol{G}_k^2\mathbf{\Sigma}_{k+1})$, and $c_k = Tr(\boldsymbol{W}_k^2\boldsymbol{G}_k\mathbf{\Sigma}_{k+1}) - Tr(\boldsymbol{G}_k)$. The left side of the above equation is a quadratic function and the roots (the optimal step sizes) are given by

$$\eta_{k,opt} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{31}$$

Since the goal is to minimize the cost function (12), the optimal step size, $\eta_{k,\,opt}$, should be selected such that the second derivative of the cost function $J(\mathbf{W})$ be a positive number, i.e.,

$$\frac{\partial^2 J(\boldsymbol{W}_{k+1})}{\partial^2\eta_k} = 2a\eta_k + b \geq 0, \tag{32}$$

that means the root with the negative sign in (30) is not acceptable, and we have

$$\eta_{k,opt} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}. \tag{33}$$

References

[1] Y. Aliyari Ghassabeh, H.A. Moghaddam, New adaptive algorithms for extracting optimal features from Gaussian data, in: Proceedings of the International Conference on Computer Vision Theory and Applications, Barcelona, Spain, 2007, pp. 182–187.

[2] Y. Aliyari Ghassabeh, H.A. Moghaddam, Adaptive algorithms and networks for optimal feature extraction from Gaussian data, Patt. Recognit. Lett. 31 (1) (2010) 1331–1341.

[3] Y. Aliyari Ghassabeh, H.A. Moghaddam, Adaptive linear discriminant analysis for online feature extraction, Mach. Vis. Appl. 24 (4) (2013) 777–794.

[4] Y. Aliyari Ghassabeh, F. Rudzicz, H.A. Moghaddam, Fast incremental LDA feature extraction, Patt. Recognit. 48 (6) (2015) 1999–2012.

[5] A. Benveniste, A. Metivier, P. Priouret, Adaptive Algorithms and Stochastic Approximations, New York: Springer-Verlag, 1990.

[6] J. Brainard, D.E. Burmaster, Bivariate distributions for height and weight of men and women in the United States, Risk Anal. 12 (2) (1992) 267–275.

[7] C. Chatterjee, Z. Kang, V.P. Roychowdhury, Algorithms for accelerated convergence of adaptive PCA, IEEE Trans. Neural Netw. 11 (2) (2000) 338–355.

[8] C. Chatterjee, V.P. Roychowdhury, On self-organizing algorithms and networks for class-separability features, IEEE Trans. Neural Netw. 8 (3) (1997) 663–678.

[9] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience, 2000.

[10] H. Fischer, A History of the Central Limit Theorem, Springer-Verlag New York, 2011.

[11] K. Fukunaga, Introduction to statistical Pattern Recognition, Academic Press, 1990.

[12] G. Mimmack, D. Meyer, G. Manas, Introductory Statistics for Business: The Analysis of Business Data, Pearson Education South Africa, 2001.

[13] H.A. Moghaddam, M. Matinfar, Fast adaptive LDA using quasi-Newton algorithm, Patt. Recognit. Lett. 28 (5) (2007) 613–621.

[14] H.A. Moghaddam, M. Matinfar, S.M. Sajad Sadough, K. Amiri Zadeh, Algorithms and networks for accelerated convergence of adaptive LDA, Patt. Recognit. 34 (4) (2005) 473–483.

[15] T. Okada, S. Tomita, An optimal orthonormal system for discriminant analysis, Patt. Recognit. 18 (2) (1985) 139–144.

[16] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, 2008.