Pattern Recognition 48 (2015) 1999–2012

Contents lists available at ScienceDirect

Pattern Recognition

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

Fast incremental LDA feature extraction

Youness Aliyari Ghassabeh [a,*], 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 22 July 2014
Received in revised form
14 November 2014
Accepted 13 December 2014
Available online 24 December 2014

Keywords:
Incremental linear discriminant analysis
Accelerated algorithm
Steepest descent method
Conjugate direction method
Feature extraction

ABSTRACT

Linear discriminant analysis (LDA) is a traditional statistical technique that reduces dimensionality while preserving as much of the class discriminatory information as possible. The conventional form of the LDA assumes that all the data are available in advance and the LDA feature space is computed by finding the eigendecomposition of an appropriate matrix. However, there are situations where the data are presented in a sequence and the LDA features are required to be updated incrementally by observing the new incoming samples. Chatterjee and Roychowdhury proposed an algorithm for incrementally computing the LDA features followed by Moghaddam et al. who accelerated the convergence rate of these algorithms. The proposed algorithms by Moghaddam et al. are derived by applying the chain rule on an implicit cost function. Since the authors have not had access to the cost function they could not analyze the convergence of the proposed algorithms and the convergence of the proposed accelerated techniques were not guaranteed. In this paper, we briefly review the previously proposed algorithms, then we derive new algorithms to accelerate the convergence rate of the incremental LDA algorithm given by Chatterjee and Roychowdhury. The proposed algorithms are derived by optimizing the step size in each iteration using steepest descent and conjugate direction methods. We test the performance of the proposed algorithms for incremental LDA on synthetic and real data sets. The simulation results confirm that the proposed algorithms estimate the LDA features faster than the gradient descent based algorithm presented by Moghaddam et al., and the algorithm proposed by Chatterjee and Roychowdhury.

1. Introduction

Linear discriminant analysis (LDA) is a popular supervised technique for both dimensionality reduction and classification. The LDA has been widely used as a powerful yet simple technique for different applications in computer vision and pattern recognition community (e.g., [1–7]). The LDA technique looks for a linear transformation of the data into a lower dimensional space, for maximum discrimination between classes [8].

The typical implementation of the LDA technique 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 the LDA feature extraction to have the ability to update the computed LDA features by observing the new samples without running the algorithm on the whole data set. For example, in many real-time applications such as mobile robotics or on-line face recognition, it is important to update the extracted LDA features as soon as new observations are available. An LDA feature extraction technique that can update the LDA features by simply observing new samples is an *incremental* LDA algorithm, and this idea has been extensively studied over the last two decades.

There have been two main approaches to updating LDA features: indirect and direct. In the indirect approach, the incremental algorithms are used to update the matrices which are involved in computing the LDA features and then the LDA features are computed through solving an eigendecomposition problem. For example, Pang et al. [9] presented incremental algorithms to update the within-class and between-class scatter matrices and used them to update the LDA features. Ye et al. [10] used an incremental dimension reduction (IDR) algorithm with QR decomposition for adaptive computation of the reduced forms of within-class and between-class scatter matrices. The proposed algorithm by Uray et al. [11] involves performing PCA on a augmented matrix and then updating the LDA features. Kim et al. [12,13] used sufficient spanning approximation for updating the

* Corresponding author. Tel.: +1 343 333 4863; fax: +1 416 597 3031.
E-mail addresses: aliyari@cs.toronto.ca (Y. Aliyari Ghassabeh), frank@ai.toronto.ca (F. Rudzicz), moghaddam@eetd.kntu.ac.ir (H.A. Moghaddam).

http://dx.doi.org/10.1016/j.patcog.2014.12.012
0031-3203/

mixture scatter matrix, the between-class scatter matrix, and the projected data matrix. None of these algorithms deals with the LDA features directly, and updating the LDA features is instead done by solving an eigenvalue decomposition problem.

In contrast to the techniques above, there are incremental algorithms that update LDA features directly. Chatterjee and Roychowdhury [14] proposed an incremental self-organized LDA algorithm for updating the LDA features. The incremental LDA algorithm in [14] is composed of two parts: incremental computation of $\mathbf{Q}^{-1/2}$, where $\mathbf{Q}$ is the correlation matrix of the input data, and incremental principal component analysis (PCA). In other work, Demir and Ozmehmet [15] proposed online local learning algorithms for updating LDA features incrementally using error-correcting and the Hebbian learning rules. Both algorithms in [14,15] are highly dependent on the step size, which can be difficult to set *a priori*. Moghaddam et al. [16–18] derived new incremental algorithms to accelerate the convergence rate of the proposed algorithm in [14]. The proposed algorithms are derived based on the steepest descent, conjugate direction, Newton–Raphson, and quasi-Newton methods.

Moghaddam et al. [16–18] used an implicit cost function to find the optimal step size in order to accelerate the convergence rate. Since the authors in [16–18] have not had access to the explicit cost function, they could not guarantee the convergence of the proposed algorithms.

In this paper, we first briefly discuss the proposed algorithms in [16,17]. Then we use the steepest descent and conjugate direction methods to derive accelerated incremental algorithms for computing $\mathbf{Q}^{-1/2}$. We use the cost function in [19] to derive the accelerated $\mathbf{Q}^{-1/2}$ algorithm based on the steepest descent method. We also present a new algorithm for incremental computation of $\mathbf{Q}^{-1/2}$ using the conjugate direction method, and we introduce its accelerated version by optimizing the step size in each iteration. Finally, we combine the proposed accelerated incremental $\mathbf{Q}^{-1/2}$ algorithm with incremental PCA to derive an accelerated incremental LDA algorithm. We test the performance of the proposed algorithms using synthetic and real data sets and show that the proposed algorithms give a reliable estimate of the LDA features in fewer iterations than the algorithm in [14], and the gradient descent version in [16,17]. The incremental nature of the proposed accelerated LDA algorithms make them appropriate for fast feature extraction when the data are presented as a stream and the features can be updated as soon as each new observation is available.

The organization of the paper is as follows: in the next section, a brief review of the LDA algorithm is given. The accelerated incremental LDA feature extraction algorithm is described in Section 3. We present the accelerated $\mathbf{Q}^{-1/2}$ algorithm in Section 4. Section 5 is devoted to simulation results. Concluding remarks are given in Section 6.

2. Linear discriminant analysis

Let $\mathbf{x}_i \in \mathbb{R}^d, i=1,2,\ldots$ denote the observed data which belongs to exactly one of the available K classes, $\omega_1,\ldots,\omega_K$, and let $P(\omega_i)$, $i=1,\ldots,K$ denote the prior probability of the ith class ω_i. Let $\mathbf{m}_i, i=1,\ldots,K$ denote the mean vector for class ω_i, i.e., $\mathbf{m}_i = E(\mathbf{x}|\mathbf{x}\in\omega_i)$, and let $\mathbf{\Sigma}_i$ denote the covariance matrix of the ith class, i.e., $\mathbf{\Sigma}_i = E[(\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^t|\mathbf{x}\in\omega_i], i=1,\ldots,K$. In order to achieve the maximum class separability, in addition to dimensionality reduction, the following three matrices are defined [20]:

1. Within-class scatter matrix $\mathbf{\Sigma}_W$:

$$\mathbf{\Sigma}_W = \sum_{i=1}^{K} P(\omega_i) E[(\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^t|\mathbf{x}\in\omega_i] = \sum_{i=1}^{K} P(\omega_i)\mathbf{\Sigma}_i, \quad (1)$$

2. Between-class scatter matrix $\mathbf{\Sigma}_B$:

$$\mathbf{\Sigma}_B = \sum_{i=1}^{K} P(\omega_i)(\mathbf{m}-\mathbf{m}_i)(\mathbf{m}-\mathbf{m}_i)^t, \quad (2)$$

3. Mixture scatter matrix $\mathbf{\Sigma}_m$[1]:

$$\mathbf{\Sigma}_m = E[(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^t] = \mathbf{\Sigma}_W + \mathbf{\Sigma}_B, \quad (3)$$

where $\mathbf{m}$ is total mean vector, i.e., $\mathbf{m} = E(\mathbf{x}) = \sum_{i=1}^{K} P(\omega_i)\mathbf{m}_i$. The within-class scatter matrix, $\mathbf{\Sigma}_W$, represents the scatter of samples around their class means, the between-class scatter matrix, $\mathbf{\Sigma}_B$, represents the scatter of class means around the total mean, and the mixture scatter matrix, $\mathbf{\Sigma}_m$, is the covariance of data samples regardless of the class to which they belong. The LDA technique looks for the direction in which maximum class separability is achieved by projection of the data into those directions. That is, after projection of the data into the LDA feature space, all the samples belonging to the same class stay close together and well separated from the samples of the other classes. In order to quantify this, different measures of separation have been defined, for example [20]

$$J_1 = \mathrm{Tr}(\mathbf{\Sigma}_W^{-1}\mathbf{\Sigma}_B); \quad J_2 = \frac{\mathrm{Tr}(\mathbf{\Sigma}_B)}{\mathrm{Tr}(\mathbf{\Sigma}_W)}; \quad J_3 = \ln\|\mathbf{\Sigma}_W^{-1}\mathbf{\Sigma}_B\|; \quad J_4 = \frac{\det\mathbf{\Sigma}_B}{\det\mathbf{\Sigma}_W}. \quad (4)$$

It can be shown that the LDA transformation matrix, $\mathbf{\Phi}_{LDA,p}$, into a p-dimensional ($p<D$) space is given by p leading eigenvectors of $\mathbf{\Sigma}_W^{-1}\mathbf{\Sigma}_B$ [21]. Since $Rank(\mathbf{\Sigma}_B) \le K-1$, then the reduced dimension by the LDA technique is at most $K-1$, i.e., $p \le K-1$. The between-class scatter matrix, $\mathbf{\Sigma}_B$, is not in general a full rank matrix and using (3) it can be replaced by $\mathbf{\Sigma}_m - \mathbf{\Sigma}_W$. As a result, instead of finding leading eigenvectors of $\mathbf{\Sigma}_W^{-1}\mathbf{\Sigma}_B$, one can solve the generalized eigenvalue problem:

$$\mathbf{\Sigma}_m\mathbf{\Phi}_{LDA} = \mathbf{\Sigma}_W\mathbf{\Phi}_{LDA}\mathbf{\Lambda}, \quad (5)$$

where $\mathbf{\Lambda}$ is the diagonal eigenvalue matrix and the desired p LDA features are given by p columns of $\mathbf{\Phi}_{LDA}$ corresponding to the largest eigenvalues of $\mathbf{\Lambda}$ [20]. Further manipulation of (5) reveals that the above problem can be simplified to the following symmetric eigenvalue problem[2]:

$$\mathbf{\Sigma}_W^{-1/2}\mathbf{\Sigma}_m\mathbf{\Sigma}_W^{-1/2}\mathbf{\Psi} = \mathbf{\Psi}\mathbf{\Lambda}, \quad (6)$$

where $\mathbf{\Psi} = \mathbf{\Sigma}_W^{1/2}\mathbf{\Phi}_{LDA}$. Note that since in most of the real world applications the statistics of the observed data are not available, the above mentioned matrices can be found as [20]

$$\mathbf{\Sigma}_m = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\mathbf{m})(\mathbf{x}_i-\mathbf{m})^t; \mathbf{\Sigma}_B = \frac{1}{n}\sum_{j=1}^{K}|\omega_j|(\mathbf{m}_j-\mathbf{m})(\mathbf{m}_j-\mathbf{m})^t$$
$$\mathbf{\Sigma}_W = \frac{1}{n}\sum_{j=1}^{K}\sum_{\mathbf{x}\in\omega_j}(\mathbf{x}-\mathbf{m}_j)(\mathbf{x}-\mathbf{m}_j)^t, \quad (7)$$

where n is the total number of samples, and $|\omega_j|, j=1,\ldots,K$ denotes the size of class ω_j, i.e., $\sum_{j=1}^{K}|\omega_j| = n$.

3. Fast incremental LDA feature extraction

As mentioned, the main LDA features are the eigenvectors of $\mathbf{\Sigma}_W^{-1}\mathbf{\Sigma}$ associated with the largest eigenvalues. Let $\mathbf{x}_k \in \mathbb{R}^D$, $k=1,2,\ldots$, denote the observed vector sequence such that $\mathbf{x}_k$ belongs to exactly one of K classes $\omega_1,\ldots,\omega_K$. Define three new

[1] This is also called the covariance matrix.

[2] The within-scatter matrix, $\mathbf{\Sigma}_W$, is the sum of positive definite matrices, therefore itself is also a positive definite matrix and $\mathbf{\Sigma}_W^{-1/2}$ exists.

sequences $\{\mathbf{y}_k\}_{k=1,2,\ldots}$, $\{\mathbf{z}_k\}_{k=1,2,\ldots}$, and $\{\mathbf{u}_k\}_{k=1,2,\ldots}$ as follows:

$$\mathbf{y}_k=\mathbf{x}_k-\mathbf{m}_k^{\omega_{\mathbf{x}_k}};\quad \mathbf{z}_k=\mathbf{x}_k-\mathbf{m}_k;\quad \mathbf{u}_k=\mathbf{W}_k\mathbf{z}_k, \tag{8}$$

where $\mathbf{m}_k^{\omega_{\mathbf{x}_k}}$ denotes the sample mean of the class to which $\mathbf{x}_k$ belongs at the k-th iteration, $\mathbf{W}_k$ is estimate of the inverse of the square root of the covariance matrix (next section is devoted to incremental computation of $\mathbf{W}_k$), and $\mathbf{m}_k$ denotes the total mean estimate at the k-th iteration, i.e. $\mathbf{m}_k=\sum_{i=1}^{k}\mathbf{x}_i/k$. Each new incoming sample updates the total class mean and the class mean to which it belongs, and keeps the other class means unchanged. From Theorem 2 and Theorem 3 in [14], we have

$$\lim_{k\to\infty}E[\mathbf{z}_k\mathbf{z}_k^t]=\boldsymbol{\Sigma}_m, \tag{9}$$

$$\lim_{k\to\infty}E[\mathbf{y}_k\mathbf{y}_k^t]=\boldsymbol{\Sigma}_W, \tag{10}$$

$$\lim_{k\to\infty}E[\mathbf{u}_k\mathbf{u}_k^t]=\boldsymbol{\Sigma}_W^{-1/2}\boldsymbol{\Sigma}_m\boldsymbol{\Sigma}_W^{-1/2}. \tag{11}$$

Let $\mathbf{Q}^{-1/2}$ denote an algorithm that estimates the inverse of the square root of the covariance matrix $\boldsymbol{\Sigma}$[3] of its input data, e.g., if $\{\mathbf{x}_k\}_{k=1,2,\ldots}$, then the output is an estimate of $\boldsymbol{\Sigma}^{-1/2}$. In other words, the $\mathbf{Q}^{-1/2}$ algorithm takes $\mathbf{x}_k$'s as its input and generates a sequence $\{\mathbf{W}_k\}_{k=1,2,\ldots}$ that converges to the inverse of the square root of the covariance matrix of $\mathbf{x}_k$'s. Eq. (10) implies that if the $\mathbf{Q}^{-1/2}$ algorithm is trained using the sequence $\{\mathbf{y}_k\}_{k=1,2,\ldots}$, then the output of the $\mathbf{Q}^{-1/2}$ algorithm will converge to $\boldsymbol{\Sigma}_W^{-1/2}$, i.e., $\lim_{k\to\infty}\mathbf{W}_k=\boldsymbol{\Sigma}_W^{-1/2}$.

Chatterjee and Roychowdhury [14] showed that, in order to extract the leading LDA features incrementally, we need to compute the leading eigenvectors of the correlation matrix of the sequence $\{\mathbf{u}_k\}_{k=1,2,\ldots}$. The following formula was proposed for incrementally computing the p ($p\le n$) leading eigenvectors of the correlation matrix of a sequence $\mathbf{x}_k\in\mathbb{R}^n, k=1,2,\ldots$ [22,23]:[4]

$$\boldsymbol{\Phi}_{k+1}=\boldsymbol{\Phi}_k+\gamma_k(\mathbf{x}_k\mathbf{x}_k^t\boldsymbol{\Phi}_k-\boldsymbol{\Phi}_k UT[\boldsymbol{\Phi}_k^t\mathbf{x}_k\mathbf{x}_k^t\boldsymbol{\Phi}_k]), \tag{12}$$

where $\boldsymbol{\Phi}_k$ is a $n\times p$ matrix whose columns converge to p leading eigenvectors of the correlation matrix $\mathbf{Q}$ associated with the largest eigenvalues, γ_k is the step size, and the operator $UT[.]$ sets all the elements below the main diagonal of its entry to zero. Let $\boldsymbol{\Psi}$ and $\boldsymbol{\Lambda}_1$ denote the corresponding eigenvector and eigenvalue matrices of $\boldsymbol{\Sigma}_W^{-1/2}\boldsymbol{\Sigma}_m\boldsymbol{\Sigma}_W^{-1/2}$, i.e., $\boldsymbol{\Sigma}_W^{-1/2}\boldsymbol{\Sigma}_m\boldsymbol{\Sigma}_W^{-1/2}\boldsymbol{\Psi}=\boldsymbol{\Psi}\boldsymbol{\Lambda}_1$. Let $\boldsymbol{\Phi}_{LDA}$ and $\boldsymbol{\Lambda}_2$ denote the corresponding eigenvector and eigenvalue matrices of $\boldsymbol{\Sigma}_W^{-1}\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\Sigma}_W^{-1}\boldsymbol{\Sigma}\boldsymbol{\Phi}_{LDA}=\boldsymbol{\Phi}_{LDA}\boldsymbol{\Lambda}_2$. The incremental LDA feature extraction is done in two steps: (1) using $\mathbf{Q}^{-1/2}$ algorithm to estimate $\boldsymbol{\Sigma}_W^{-1/2}$, (2) computing the eigenvector matrix of $\boldsymbol{\Sigma}_W^{-1/2}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_W^{-1/2}$, $\boldsymbol{\Psi}$, using (12). Since $\boldsymbol{\Sigma}_W^{-1/2}\boldsymbol{\Psi}=\boldsymbol{\Phi}_{LDA}$, the product of the outputs of these two steps provides the desired LDA features, i.e., $\boldsymbol{\Phi}_{LDA}$ [14]. In the next section, we first introduce incremental algorithms for $\mathbf{Q}^{-1/2}$ and then by optimizing the learning rate, we present accelerated versions of the $\mathbf{Q}^{-1/2}$ algorithm. Note that finding the optimal learning rate for the $\mathbf{Q}^{-1/2}$ algorithm will accelerate the convergence rate of step (i), which leads a faster estimate of the desired LDA features. The proposed accelerated incremental LDA feature extraction algorithm is summarized in Algorithm 1. The structure of the proposed accelerated incremental LDA feature extraction is also given in Fig. 1.

[3] For simplicity, we use $\boldsymbol{\Sigma}$ instead of $\boldsymbol{\Sigma}_m$.

[4] There are other techniques for incremental computing of the eigenvectors of a correlation matrix, for example see [24–26].

4. A fast $\mathbf{Q}^{-1/2}$ algorithm

The authors in [14] showed that incremental LDA feature extraction involves the computation of $\mathbf{Q}^{-1/2}$, where $\mathbf{Q}$ is the symmetric positive definite correlation matrix of a uniformly bounded random vector sequence $\mathbf{x}_i\in\mathbb{R}^D, i=1,2,\ldots$. They proposed an algorithm, called the $\mathbf{Q}^{-1/2}$ algorithm, to find $\mathbf{Q}^{-1/2}$ incrementally as follows:

$$\mathbf{W}_{k+1}=\mathbf{W}_k+\eta_k(\mathbf{I}-\mathbf{W}_k\mathbf{x}_{k+1}\mathbf{x}_{k+1}^t\mathbf{W}_k^t), \tag{13}$$

where $\mathbf{W}_{k+1}$ represents the $\mathbf{Q}^{-1/2}$ estimate at the $k+1$-th iteration, $\mathbf{x}_{k+1}$ is the new incoming input vector at time $k+1$, η_k is the step size, and $\mathbf{W}_0\in\mathbb{R}^{n\times n}$ is chosen to be a symmetric positive definite matrix. Using stochastic approximation, Chatterjee and Roychowdhury [14] proved that, under certain conditions, the sequence $\{\mathbf{W}_k\}_{k=0,1,2,\ldots}$ converges to $\mathbf{Q}^{-1/2}$ with unit probability, i.e., $\lim_{k\to\infty}\mathbf{W}_k=\mathbf{Q}^{-1/2}$. The proposed incremental $\mathbf{Q}^{-1/2}$ algorithm in [14] suffers from a low convergence rate, due to using a fixed or decreasing step size. The authors in [16,17] used different techniques, including the steepest descent and conjugate direction methods, to find the optimal step size in each iteration in order to accelerate the convergence rate of the incremental $\mathbf{Q}^{-1/2}$ algorithm. They showed that $\mathbf{x}_{k+1}\mathbf{x}_{k+1}^t$ in (13) can be replaced by $\mathbf{Q}_{k+1}$, which is the correlation matrix estimate using the first $k+1$ incoming samples[5]. Therefore, the incremental $\mathbf{Q}^{-1/2}$ algorithm in (13) can be rewritten in the following form:

$$\begin{aligned}\mathbf{W}_{k+1}&=\mathbf{W}_k+\eta_k\mathbf{G}_{k+1},\\ \mathbf{G}_{k+1}&=\mathbf{I}-\mathbf{W}_k\mathbf{Q}_{k+1}\mathbf{W}_k^t.\end{aligned} \tag{14}$$

The correlation matrix estimate $\mathbf{Q}_k$ can be updated incrementally by [14]

$$\mathbf{Q}_{k+1}=\mathbf{Q}_k+\theta_k(\mathbf{x}_{k+1}\mathbf{x}_{k+1}^t-\mathbf{Q}_k), \tag{15}$$

where, for a stationary process, we have $\theta=1/(k+1)$. Note that if we use the covariance estimate, $\boldsymbol{\Sigma}_{k+1}$, instead of $\mathbf{Q}_{k+1}$ in (14), the sequence converges to $\boldsymbol{\Sigma}^{-1/2}$. The covariance estimate $\boldsymbol{\Sigma}_{k+1}$ can be updated by

$$\boldsymbol{\Sigma}_{k+1}=\boldsymbol{\Sigma}_k+\theta_k((\mathbf{x}_{k+1}-\mathbf{m}_{k+1})(\mathbf{x}_{k+1}-\mathbf{m}_{k+1})^t-\boldsymbol{\Sigma}_k), \tag{16}$$

where θ is $1/(k+1)$ for a stationary process and the mean vector $\mathbf{m}_{k+1}$ can be estimated adaptively as follows [14]:

$$\mathbf{m}_{k+1}=\mathbf{m}_k+\kappa_k(\mathbf{x}_{k+1}-\mathbf{m}_k), \tag{17}$$

where for a stationary process we have $\kappa_k=1/(k+1)$.

Replacing $\mathbf{Q}_{k+1}$ in (14) by the correlation matrix $\mathbf{Q}$ and comparing it to the general form of an adaptive algorithm [27] reveals that the updating function $\mathbf{G}(\mathbf{W})=\mathbf{I}-\mathbf{WQW}$ can be considered as the negative of the gradient of some cost function $J(\mathbf{W})$ with respect to $\mathbf{W}$, i.e.,

$$-\nabla J(\mathbf{W})=\mathbf{G}(\mathbf{W})=\mathbf{I}-\mathbf{WQW}. \tag{18}$$

The cost function $J(\mathbf{W})$ and its derivative with respect to the step size can be used to find the optimal step size $\eta_k, k=1,2,\ldots$ in each iteration. Since the cost function $J(\mathbf{W})$ was unknown, the authors in [16,17] could not compute the derivative of the cost function $J(\mathbf{W})$ with respect to η_k directly. Instead, they proposed using the chain rule in order to break the derivative into two computable parts. The proposed technique by [16,17] works fine when all the elements of $\mathbf{W}$ are independent of each other, otherwise the proposed chain rule formula may not give a right answer, as explained in Section 4.2. Furthermore, since the authors in [16,17] have not had access to the cost function they could not analyse the convergence of the proposed accelerated algorithms. In the

[5] The authors in [16,17] also introduced new formula for online estimation of the correlation matrix.

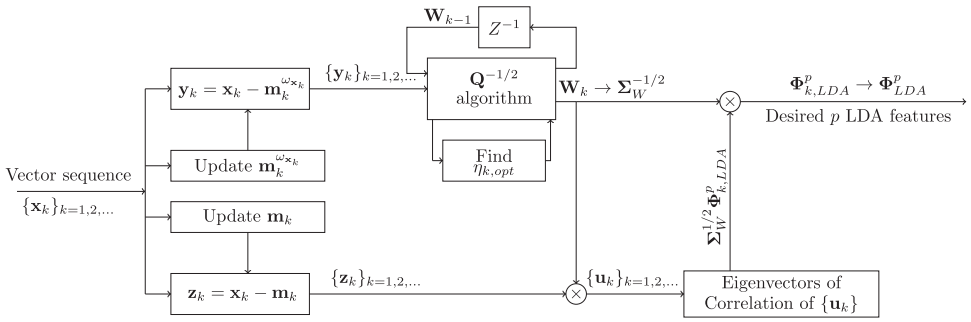

Fig. 1. Accelerated incremental LDA feature extraction. The random vector sequence $\{\mathbf{x}_k\}_{k=1,2,\ldots}$ is observed sequentially and is used to generate two new sequences $\{\mathbf{y}_k\}_{k=1,2,\ldots}$ and $\{\mathbf{z}_k\}_{k=1,2,\ldots}$. The sequence $\{\mathbf{y}_k\}_{k=1,2,\ldots}$ is used to train accelerated $\mathbf{Q}^{-1/2}$ algorithm. The new sequence $\{\mathbf{u}_k\}_{k=1,2,\ldots}$ is generated by product of the output of $\mathbf{Q}^{-1/2}$ algorithm and the sequence $\{\mathbf{z}_k\}_{k=1,2,\ldots}$. The p leading eigenvectors of the correlation matrix of the sequence $\{\mathbf{u}_k\}$ and output of $\mathbf{Q}^{-1/2}$ algorithm are used to update the LDA features.

followings, We first briefly review the $\mathbf{Q}^{-1/2}$ algorithm using the gradient descent method. Then we present the correct forms of the optimal step size computed using the steepest descent and conjugate direction methods in order to accelerate the convergence rate of the incremental $\mathbf{Q}^{-1/2}$ algorithm.

4.1. Gradient descent method

Aliyari Ghassabeh and Moghaddam [19] introduced a cost function $J(\mathbf{W})$ with the global minimum at $\mathbf{Q}^{-1/2}$ and showed that applying the gradient descent method on $J(\mathbf{W})$ would give the adaptive algorithm in (14).[6] The proposed cost function $J(\mathbf{W}): \mathcal{C} \to \mathbb{R}$ is given as [19]

$$J(\mathbf{W}) = \frac{1}{3}\mathrm{Tr}\left[(\mathbf{W}\mathbf{Q}^{1/2} - \mathbf{I})^2(\mathbf{W} + 2\mathbf{Q}^{-1/2})\right], \tag{19}$$

where $\mathcal{C} \subset \mathbb{R}^{n\times n}$ is the set of all symmetric positive definite matrices $\mathbf{W}$ that commute with $\mathbf{Q}^{1/2}$, i.e., $\mathbf{W}\mathbf{Q}^{1/2} = \mathbf{Q}^{1/2}\mathbf{W}$, Tr[.] is the matrix trace function, and $\mathbf{I}$ denotes the identity matrix. By definition, the cost function $J(\mathbf{W})$ in (19) is one third of the trace of the product of a symmetric semi-positive definite matrix, $(\mathbf{W}\mathbf{Q}^{1/2} - \mathbf{I})^2$, with a symmetric positive definite matrix, $\mathbf{W} + 2\mathbf{Q}^{-1/2}$. Hence, the cost function itself is a semi-positive definite matrix [28], i.e., $J(\mathbf{W}) \geq 0$ for all $\mathbf{W} \in \mathcal{C}$. By taking the gradient of the cost function in (19) with respect to $\mathbf{W}$ and equating it to zero, we obtain

$$\nabla J(\mathbf{W}) = \mathbf{W}\mathbf{Q}\mathbf{W} - \mathbf{I} = 0. \tag{20}$$

Eq. (20) reveals that, in the domain $\mathcal{C}$, the cost function $J(\mathbf{W})$ has a unique stationary point that occurs at $\mathbf{Q}^{-1/2}$. Since $J(\mathbf{Q}^{-1/2}) = 0$, then the matrix $\mathbf{Q}^{-1/2}$ is the unique global minimum of the cost function $J(\mathbf{W})$ over the convex set $\mathcal{C}$. Therefore, the gradient descent algorithm can be used to minimize the cost function $J(\mathbf{W})$ recursively in order to find the global minimum, $\mathbf{Q}^{-1/2}$. By applying the gradient descent method on the cost function $J(\mathbf{W})$, we obtain the following recursive definition:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k(\mathbf{I} - \mathbf{W}_k\mathbf{Q}_{k+1}\mathbf{W}_k). \tag{21}$$

Since the true value of $\mathbf{Q}$ is not known in advance, we replace it by its estimate at the $(k+1)$-th iteration.[7] It is straightforward to show that if $\mathbf{W}_0$ is a symmetric matrix that commutes with $\mathbf{Q}^{1/2}$, then the generated sequence $\{\mathbf{W}_k\}_{k=0,1,\ldots}$ will also have the same properties. The authors in [14] showed that if $\mathbf{W}_0$ is a semi-positive definite matrix, then there exists a uniform upper bound for the step size η_k such that the members of the generated sequence $\{\mathbf{W}_k\}_{k=0,1,2,\ldots}$ also remain semi-positive definite matrices (Lemma 5 in [14]). Therefore, if the initial guess $\mathbf{W}_0$ is chosen to be in $\mathcal{C}$, under certain conditions the sequence $\{\mathbf{W}_k\}_{k=0,1,\ldots}$ remains in the domain of the cost function, i.e. $\mathbf{W}_k \in \mathcal{C}, k = 1, 2, \ldots$. The cost function $J(\mathbf{W})$ along the sequence $\{\mathbf{W}_k\}_k = 0, 1, \ldots$ is a decreasing sequence and we have

$$J(\mathbf{W}_0) \geq J(\mathbf{W}_1) \geq J(\mathbf{W}_2) \geq \cdots \geq 0. \tag{22}$$

The boundedness from below and monotonically decreasing properties of the sequence $\{J(\mathbf{W}_k)\}_{k=0,1,\ldots}$ implies the convergence of $\{J(\mathbf{W}_k)\}_{k=0,1,\ldots}$ [29]. For the gradient descent algorithm the convergence occurs when the gradient of the cost function becomes zero. Since the only stationary point of the cost function $J(\mathbf{W})$ on the domain $\mathcal{C}$ happens at $\mathbf{Q}^{-1/2}$, therefore the sequence $\{\mathbf{W}_k\}$ converges to $\mathbf{Q}^{-1/2}$, i.e., $\lim_{k\to\infty}\mathbf{W}_k = \mathbf{Q}^{-1/2}$ and $\lim_{k\to\infty}J(\mathbf{W}_k) = 0$.

4.2. Steepest descent method

In steepest descent, the optimal step size $\eta_{k,opt}$ at the $k+1$-th iteration is found by equating the first derivative of the cost function $J(\mathbf{W})$ with respect to η_{k+1} to zero [30]. The authors in [16,17] claimed that the first derivative can be written as product of two parts using the chain rule as follows (Eq. (12) in [16] and Eq. (15) in [17]):

$$\frac{\partial J(\mathbf{W}_{k+1})}{\partial \eta_k} = \frac{J(\mathbf{W}_{k+1})}{\partial \underline{\mathbf{W}}_{k+1}} \cdot \frac{\partial \mathbf{W}_{k+1}}{\partial \eta_k}, \tag{23}$$

where $\underline{\mathbf{W}}$ represent the vector form of matrix $\mathbf{W}$ and '$\cdot$' is the inner product between two vectors. The above equality is correct when all the elements of matrix $\mathbf{W}_{k+1}$ are independent of each other. Otherwise, Eq. (23) may not be correct in general. For example consider the following situation where diagonal elements of matrix $\mathbf{W}$ are dependent and the cost function is defined as the trace of its matrix input

$$\mathbf{W} = \begin{pmatrix} \delta & w_{1,2} \\ w_{2,1} & 2\delta \end{pmatrix} \quad \text{and} \quad J(\mathbf{W}) = \mathrm{Tr}(\mathbf{W}). \tag{24}$$

By taking the direct derivative of the cost function with respect to δ, we get $\partial J(\mathbf{W})/\partial\delta = \partial 3\delta/\partial\delta = 3$. Using the chain rule in (23), we obtain

$$\frac{J(\mathbf{W})}{\partial \mathbf{W}} = \begin{pmatrix} 3 & 0 \\ 0 & 1.5 \end{pmatrix} \quad \text{and} \quad \frac{\mathbf{W}}{\partial\delta} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \tag{25}$$

where their inner product, 6, is not equal to $\partial J(\mathbf{W})/\partial\delta = 3$.

Furthermore, since the authors in [16,17] have not had access to the explicit form of the cost function, they could not show the convergence of the proposed algorithms. Hence, although the resulting optimal step size seem working fine in simulations, lack of an explicit cost function makes them less appealing. In the following by exploiting the explicit cost function we derive new accelerated algorithms for

[6] Specifically, applying the gradient descent method on the introduced cost function in [19] gives $\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k(\mathbf{I} - \mathbf{W}_k\mathbf{Q}\mathbf{W}_k)$. Since $\mathbf{Q}$ is not available in advance, it will be replaced by its estimate $\mathbf{Q}_{k+1}$, which asymptotically converges to $\mathbf{Q}$.

[7] Note that replacing $\mathbf{Q}_{k+1}$ in (21) by $\mathbf{x}_{k+1}\mathbf{x}_{k+1}^t$ (as an estimate of the correlation matrix) gives (13). Eq. (21) was proposed in [16,17] as a smooth version of (13).

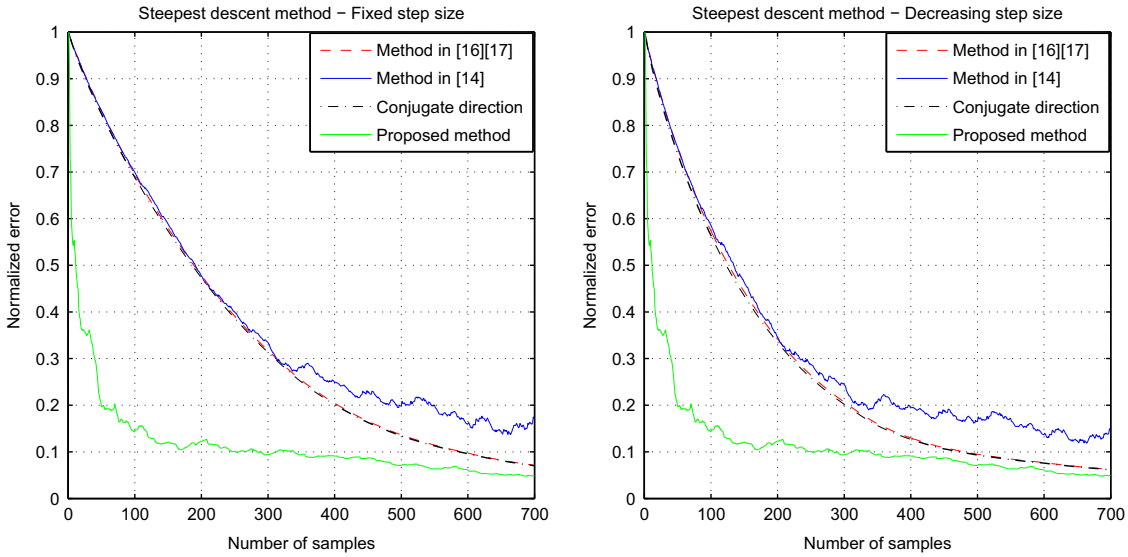

Fig. 2. Estimating $\mathbf{Q}^{-1/2}$ using the steepest descent method. The proposed algorithm in [14], the gradient descent based algorithm in [16,17], and the proposed algorithm in (33) use a fixed step size $\eta_k = 0.01$ (α_k for algorithm in (33)) in the left subfigure and use a decreasing step size $\eta_k = 1/(50+0.1\times k)$ (α_k for algorithm in (33)) in the right subfigure. The proposed $\mathbf{Q}^{-1/2}$ algorithm based on the steepest descent method finds the optimal step size in each iteration.

incremental LDA using the steepest descent and conjugate direction (next subsection) methods.

By taking the derivative of (19) with respect to the step size η_k, equating to zero, and a few additional operations (for details see the appendix) we get

$$\frac{\partial J(\mathbf{W}_{k+1})}{\partial \eta_k} = a\eta_k^2 + b\eta_k + c = 0, \tag{26}$$

where $a = \mathrm{Tr}(\mathbf{G}_k^3\mathbf{Q})$, $b = 2\,\mathrm{Tr}(\mathbf{W}_k\mathbf{G}_k^2\mathbf{Q})$, and $c = \mathrm{Tr}(\mathbf{W}_k^2\mathbf{G}_k\mathbf{Q}) - \mathrm{Tr}(\mathbf{G}_k)$. Eq. (26) is a quadratic equation and the roots, the optimal step sizes, are given by

$$\eta_{k,opt} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{27}$$

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 correlation matrix $\mathbf{Q}$ is not available, it must be replaced by its estimate $\mathbf{Q}_{k+1}$ (as the number of the observed samples increases we get a better estimate of the $\mathbf{Q}$). The optimal step size using the steepest descent method is given by

$$\eta_{k,opt} = \frac{-b_{k+1} + \sqrt{b_{k+1}^2 - 4a_{k+1}c_{k+1}}}{2a_{k+1}}, \tag{28}$$

where $a_{k+1} = \mathrm{Tr}(\mathbf{G}_k^3\mathbf{Q}_{k+1})$, $b_{k+1} = 2\,\mathrm{Tr}(\mathbf{W}_k\mathbf{G}_k^2\mathbf{Q}_{k+1})$, and $c_{k+1} = \mathrm{Tr}(\mathbf{W}_k^2\mathbf{G}_k\mathbf{Q}_{k+1}) - \mathrm{Tr}(\mathbf{G}_k)$. Therefore, the accelerated incremental $\mathbf{Q}^{-1/2}$ algorithm using the steepest descent method has the following form:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_{k,opt}(\mathbf{I} - \mathbf{W}_k\mathbf{Q}_{k+1}\mathbf{W}_k^t), \tag{29}$$

where the correlation estimate $\mathbf{Q}_{k+1}$ is given in (15) and $\eta_{k,opt}$ in each iteration is computed using (28). The accelerated $\mathbf{Q}^{-1/2}$ algorithm (based on the steepest descent method) is summarized in Algorithm 2.

4.3. Conjugate direction method

The adaptive conjugate direction algorithm for minimizing a cost function $J(\mathbf{W})$ can be written as [31]

$$\begin{aligned}&\mathbf{W}_{k+1} = \mathbf{W}_k + \alpha_k\mathbf{D}_k,\\ &\mathbf{D}_{k+1} = -\nabla_{\mathbf{W}}J(\mathbf{W}_{k+1}) + \beta_k\mathbf{D}_k,\end{aligned} \tag{30}$$

where the scalar β_k can be chosen by several different methods [32]. For simulations in this paper, we computed β based on the Polak–Reeves (PR) method as [32]

$$\beta_k = \frac{\|\nabla J(\mathbf{W}_{k+1})^T(\nabla J(\mathbf{W}_{k+1}) - \nabla J(\mathbf{W}_{k+1})\|}{\|\nabla J(\mathbf{W}_k)\|^2} \tag{31}$$

where $\|.\|$ denotes the matrix norm. It is common to initialize $\mathbf{D}_0$ to be the gradient of the cost function at $\mathbf{W}_0$ with negative sign, i.e., $\mathbf{D}_0 = -\nabla J(\mathbf{W}_0)$. Using (18) and (30) the adaptive conjugate direction algorithm for computing $\mathbf{Q}^{-1/2}$ is

$$\begin{aligned}&\mathbf{D}_0 = \mathbf{I} - \mathbf{W}_0\mathbf{Q}\mathbf{W}_0,\\ &\mathbf{W}_{k+1} = \mathbf{W}_k + \alpha_k\mathbf{D}_k,\\ &\mathbf{D}_{k+1} = (\mathbf{I} - \mathbf{W}_{k+1}\mathbf{Q}\mathbf{W}_{k+1}) + \beta_k\mathbf{D}_k,\end{aligned} \tag{32}$$

where β_k is computed using (31). Since the data are presented as a stream, the correlation matrix $\mathbf{Q}$ is not known in advance. We need to replace $\mathbf{Q}$ in (32) by its estimate at $k+1$-th iteration which gives a new algorithm for incremental computation of $\mathbf{Q}^{-1/2}$ based on conjugate direction method as follows:

$$\begin{aligned}&\mathbf{D}_0 = \mathbf{I} - \mathbf{W}_0\mathbf{Q}_0\mathbf{W}_0,\\ &\mathbf{W}_{k+1} = \mathbf{W}_k + \alpha_k\mathbf{D}_k,\\ &\mathbf{D}_{k+1} = (\mathbf{I} - \mathbf{W}_{k+1}\mathbf{Q}_{k+1}\mathbf{W}_{k+1}) + \beta_k\mathbf{D}_k,\end{aligned} \tag{33}$$

where $\mathbf{Q}_0$ is the initial estimate of the correlation matrix. The algorithm in (33) is a new algorithm for incremental computing of $\mathbf{Q}^{-1/2}$ based on conjugate direction method.

To find the optimal value of the step size in order to accelerate the convergence rate of the proposed $\mathbf{Q}^{-1/2}$ algorithm in (33), we need to find a step size α to minimize $f(\alpha) = J(\mathbf{W}_k + \alpha\mathbf{D}_k)$, i.e., $\alpha_{k,opt} = \arg\min_{\alpha\in\mathbb{R}} J(\mathbf{W}_k + \alpha\mathbf{D}_k)$. This goal can be achieved by simply taking the first derivative of the cost function J with respect to α_k and equate it to zero. Expanding (19) and using (32), the cost function $J(\mathbf{W}_{k+1})$ can be written as follows:

$$J(\mathbf{W}_{k+1}) = \frac{1}{3}\mathrm{Tr}\left((\mathbf{W}_k + \alpha_k\mathbf{D}_k)^3\mathbf{Q}\right) - \mathrm{Tr}(\mathbf{W}_k + \alpha_k\mathbf{D}_k) + \frac{2}{3}\mathrm{Tr}(\mathbf{Q}^{-1/2}), \tag{34}$$

where $\mathbf{D}_k$ is given in (32). By taking the first derivative of the cost function $J(\mathbf{W}_{k+1})$ with respect to α_k and equating to zero, we obtain (see details in Appendix)

$$\frac{\partial J(\mathbf{W}_{k+1})}{\partial \alpha_k} = a_k\alpha_k^2 + b_k\alpha_k + c_k = 0, \tag{35}$$

where

$$a_k = \mathrm{Tr}(\mathbf{D}_k^3\mathbf{Q}_{k+1}), b_k = \frac{2}{3}\ \mathrm{Tr}\Big((\mathbf{W}_k\mathbf{D}_k^2+\mathbf{D}_k^2\mathbf{W}_k+\mathbf{D}_k\mathbf{W}_k\mathbf{D}_k)\mathbf{Q}_{k+1}\Big),$$

$$c_k = \frac{1}{3}\ \mathrm{Tr}\Big((\mathbf{W}_k^2\mathbf{D}_k+\mathbf{W}_k\mathbf{D}_k\mathbf{W}_k+\mathbf{D}_k\mathbf{W}_k^2)\mathbf{Q}_{k+1}\Big) - \mathrm{Tr}(\mathbf{D}_k). \quad (36)$$

Note that in the aforementioned formulas the correlation matrix $\mathbf{Q}$ is replaced by its estimate $\mathbf{Q}_{k+1}$. The only acceptable solution of the quadratic equation in (23) is given by

$$\alpha_{k,opt} = \frac{-b_k+\sqrt{b_k^2-4a_kc_k}}{2a_k}, \quad (37)$$

where $\alpha_{k,opt}$ is the optimal step size in order to accelerate the convergence rate of the incremental $\mathbf{Q}^{-1/2}$ algorithm in (32).

Algorithm 1. Accelerated incremental LDA feature extraction.

Input : $\mathbf{x}_1, \ldots, \mathbf{x}_N$ the data sequence of length N, p the desired number of LDA features

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

Output: p significant LDA features

begin

Initialization:: $\mathbf{W}_{old} \leftarrow \mathbf{I}$; ;

/* where $\mathbf{I}$ is the identity matrix. */

Initialize the estimated correlation matrix $\mathbf{Q}_e$ using (15);

Initialize the estimated total mean $\mathbf{m}_e$, and estimated class means $\mathbf{m}^{\omega_1}, \ldots, \mathbf{m}^{\omega_k}$ using (17);

/* The above two steps are done using a set of training data */

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

Update the estimated correlation matrix $\mathbf{Q}_e$ using $\mathbf{x}_i$ and (15) ;

Update the estimated total mean $\mathbf{m}_e$ and the estimated class mean that $\mathbf{x}_i$ belongs to it. ;

$\mathbf{y}_i \leftarrow \mathbf{x}_i - \mathbf{m}_i^{\omega_{\mathbf{x}_i}}$; $\mathbf{z}_i \leftarrow \mathbf{x}_i - \mathbf{m}_i$;

Feed $\mathbf{y}_i$ into $\mathbf{Q}^{-1/2}$ algorithm (Algorithm 2) and get $\mathbf{W}_i$;

$\mathbf{u}_i \leftarrow \mathbf{W}_i\mathbf{z}_i$;

Estimate p leading eigenvectors of $\sum_{i=1}^{k} \mathbf{u}_i\mathbf{u}_i^T$ (p leading eigenvectors of the correlation matrix of $\{\mathbf{u}_k\}_{k=1,2,\ldots}$) ;

end

$\mathbf{\Phi}_{LDA}^{p} \leftarrow$ Multiply p leading eigenvectors of correlation matrix of $\{\mathbf{u}_k\}$ and the output of $\mathbf{Q}^{-1/2}$ algorithm ;

/* Columns of $\mathbf{\Phi}_{LDA}^{p}$ are the desired p LDA features. */

end

Algorithm 2. Accelerated $\mathbf{Q}^{-1/2}$ algorithm.

Input : $\mathbf{x}_1, \ldots, \mathbf{x}_N$, the data sequence of length N

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

Output: $\mathbf{W}_N$, estimate of $\mathbf{Q}^{-1/2}$ after observing N random vectors sequentially.

begin

Initialization:: $\mathbf{W}_{old} \leftarrow \mathbf{I}$; ;

/* where $\mathbf{I}$ is the identity matrix. */

Initialize the estimated correlation matrix $\mathbf{Q}_e$;

/* This step can be done using a set of training data using (15) */

/* $\mathbf{Q}_e$ represents the estimated correlation estimate. */

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

$\mathbf{Q}_e \leftarrow \mathbf{Q}_e + (\mathbf{x}_i \mathbf{x}_i^t - \mathbf{Q}_e)/(i+1)$;

/* This step update the estimated correlation matrix, $\mathbf{Q}_e$ */

$\mathbf{G} \leftarrow \mathbf{I} - \mathbf{W}_{old}\mathbf{Q}_e\mathbf{W}_{old}$;

$a \leftarrow Tr(\mathbf{G}^3\mathbf{Q}_e)$; $b \leftarrow 2Tr(\mathbf{W}_{old}\mathbf{G}^2\mathbf{Q}_e)$;

$c \leftarrow Tr(\mathbf{W}_{old}^2\mathbf{G}\mathbf{Q}_e) - Tr(\mathbf{G})$;

$\eta = \frac{-b+\sqrt{b^2-4ac}}{2a}$;

$\mathbf{W}_{new} \leftarrow \mathbf{W}_{old} + \eta(\mathbf{I} - \mathbf{W}_{old}\mathbf{Q}_e\mathbf{W}_{old})$;

$\mathbf{W}_{old} \leftarrow \mathbf{W}_{new}$;

end

$\mathbf{W}_N \leftarrow \mathbf{W}_{new}$

end

5. Simulation results

In this section we test the performance of the proposed learning algorithms for incremental LDA feature extraction. To this end, we first compare the performance of the proposed accelerated incremental $\mathbf{Q}^{-1/2}$ algorithm with the algorithm proposed in [14], and the gradient descent based algorithm in [17].[8] Then we apply the proposed accelerated incremental LDA technique for feature extraction from both synthetic and real data sets. For all simulations in this paper it is assumed that no prior knowledge about the nature or statistics of the input data is available. The random input vectors are observed one-by-one sequentially and used to train the proposed systems.

5.1. Accelerated incremental $\mathbf{Q}^{-1/2}$ algorithm

In this simulation, we use the second 10×10 matrix given in [33] and multiply it by 20.[9] The input sequence $\{\mathbf{x}_k \in \mathbb{R}^{10}\}_{k=1,2,\ldots}$ is generated from a zero mean 10-dimensional Gaussian distribution

[8] The gradient descent based algorithm in [17], can be considered as the smooth version of the algorithm in [14].

[9] It is the matrix that has been used in [16,17].

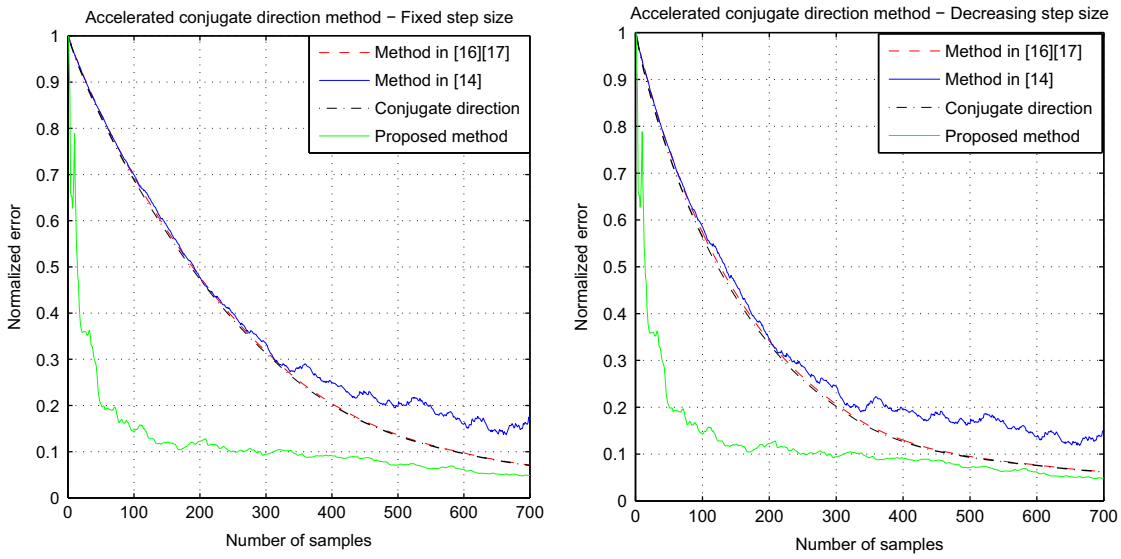

Fig. 3. Estimated $\mathbf{Q}^{-1/2}$ using the accelerated conjugate direction method. The proposed algorithm in [14], the gradient descent based algorithm in [16,17], and the proposed algorithm in (33) use a fixed step size $\eta_k = 0.01$ (α_k for algorithm in (33)) in the first figure and use a decreasing step size $\eta_k = 1/(50+0.1\times k)$ (α_k for algorithm in (33)) in the second figure. The proposed accelerated $\mathbf{Q}^{-1/2}$ algorithm based on the conjugate direction method finds the optimal step size in each iteration.

Table 1
The scaled normalized error of estimating $\mathbf{Q}^{-1/2}$ as a function of number of iterations for different algorithms. The fixed step size is $\eta = 0.01$ and the decreasing step size is $\eta_k = 1/(50+0.1\times k)$.

Method	Number of iterations							
	2	10	50	80	150	250	500	700
Method in [14], fixed η	0.994	0.961	0.832	0.748	0.589	0.402	0.199	0.161
Method in [14], decreasing η_k	0.992	0.941	0.749	0.637	0.461	0.281	0.166	0.147
Gradient descent based method in [16,17], fixed η	0.995	0.961	0.832	0.747	0.580	0.388	0.135	0.072
Gradient descent based method in [16,17], decreasing η_k	0.992	0.939	0.746	0.629	0.441	0.260	0.089	0.062
Steepest descent based method	0.915	0.417	0.384	0.286	0.164	0.135	0.074	0.059
Conjugate direction, fixed η	0.995	0.958	0.825	0.739	0.571	0.383	0.131	0.072
Conjugate direction, decreasing η_k	0.991	0.934	0.739	0.627	0.438	0.259	0.094	0.064
Accelerated conjugate direction	0.936	0.791	0.205	0.171	0.124	0.107	0.074	0.057

with the covariance matrix given by

$$\begin{bmatrix} 8.54 & & & & & & & & & \\ 0.22 & 113.80 & & & & & & & & \\ -0.10 & -1.38 & 0.60 & & & & & & & \\ -0.50 & -5.64 & 1.96 & 56.00 & & & & & & \\ 1.60 & -14.62 & 0.90 & -2.14 & 68.80 & & & & & \\ -1.58 & 1.80 & -0.82 & 3.0 & 5.06 & 45.40 & & & & \\ -0.38 & -2.48 & 0.46 & -3.86 & 5.02 & -3.60 & 6.54 & & & \\ 1.48 & 0.20 & 0.44 & 1.90 & 6.32 & 5.90 & 0.54 & 14.54 & & \\ 1.78 & 8.64 & -0.70 & -4.52 & 0.78 & -0.78 & 0.52 & -1.92 & 14.30 & \\ 0.10 & -2.06 & 0.24 & 0.92 & -0.20 & -2.26 & -0.32 & -0.34 & -0.18 & 1.30 \end{bmatrix} \tag{38}$$

We generate 700 samples in order to train the proposed $\mathbf{Q}^{-1/2}$ algorithms. The members of the input sequence are given one-by-one to each algorithm and the relative error in each iteration is recorded. The relative error is given by

$$e_k = \frac{\|\mathbf{W}_k - \mathbf{Q}^{-1/2}\|_F}{\|\mathbf{Q}^{-1/2}\|_F}, \quad k = 1, 2, \ldots, 700, \tag{39}$$

where $\|.\|_F$ denote the Frobenius-norm, $\mathbf{W}_k$ represents the estimate at the k-th iteration, and $\mathbf{Q}$ is the actual covariance[10] matrix in (38).

The initial value of the correlation matrix $\mathbf{Q}_0$ is estimated using the first 10 samples using (15) and the initial step size for all algorithms is empirically chosen to be 0.01. We compare the performance of the proposed algorithm to estimate $\mathbf{Q}^{-1/2}$ with the algorithm given in [14], and the gradient descent based algorithm in [16,17] in two scenarios. We first use a fixed step size, $\eta_k = 0.01$, for the algorithms in [14,16,17] then we repeat the simulations with a decreasing step size given by $\eta_k = 1/(50+k\times 0.1)$. For a better illustration, we scale all relative errors to the interval [0, 1]. Fig. 2 compares the relative errors resulting from estimation of the correlation matrix given in (38) for different algorithms as functions of the number of iterations. All the algorithms in Fig. 2 start with the step size $\eta = 0.01$. Although the initial condition for the proposed algorithm based on the steepest descent method is similar to the other algorithms, it is clear from Fig. 2 that the proposed algorithm outperforms all other algorithms in both cases (fixed and decreasing step size) and reaches a low error in far fewer iterations. We repeat a similar experiment and compare the performance of the previously given algorithms with the proposed accelerated $\mathbf{Q}^{-1/2}$ algorithm based on the conjugate direction method in Fig. 3. The algorithm based on the conjugate direction method in (33) uses either a fixed or decreasing step size, α_k, but the accelerated version finds the optimal step size using (37) in each iteration. The proposed accelerated algorithm based on the conjugate direction method provides a small relative error in fewer iterations compared to existing algorithms. Table 1 compares the amount of error in estimating the correlation matrix in (38) for different numbers of iterations. It is clear from Table 1 that the proposed accelerated algorithms based on the steepest descent and conjugate direction methods give an accurate estimate of $\mathbf{Q}^{-1/2}$ in fewer number of

[10] Note that in this simulation, the correlation and covariance matrices are equal.

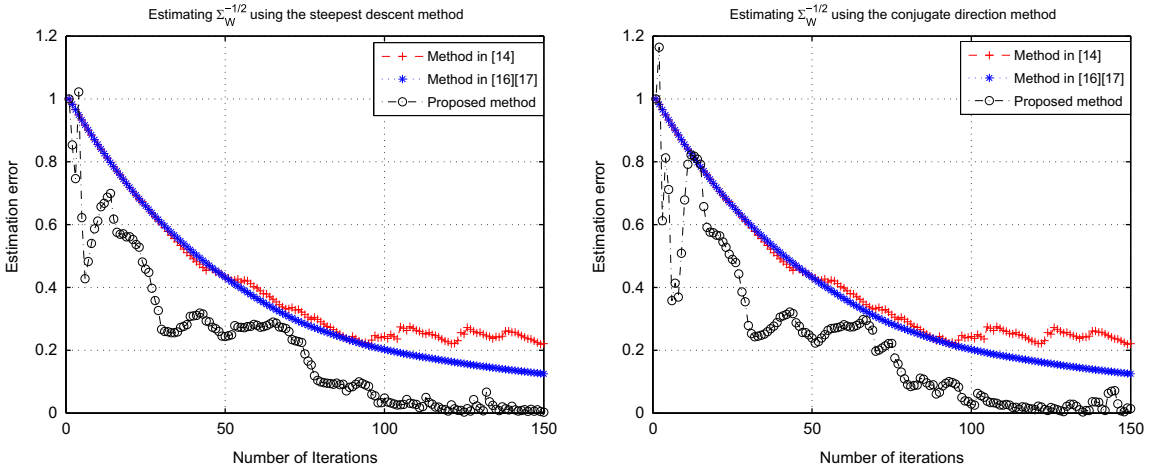

Fig. 4. The left side compares the performance of the proposed algorithm based on the steepest descent method with the algorithm in [14] and the gradient descent based algorithm in [17] for estimating $\mathbf{\Sigma}_W^{-1/2}$. The right side compares the performance of the proposed algorithm based on the conjugate direction method with the algorithms in [14,17].

Table 2

The scaled relative error of estimating $\mathbf{\Sigma}_W^{-1/2}$ for the Iris data set as a function of the number of iterations for different algorithms. The algorithms in [14,16,17] use a decreasing step size given by $1/(10+i\times 0.15)$. The initial step size for the proposed accelerated algorithms based on the steepest descent and conjugate direction is set to $1/10$.

Method	Number of iterations							
	2	5	20	40	75	100	130	150
Method in [14]	0.983	0.933	0.717	0.491	0.313	0.244	0.257	0.221
Gradient descent based method in [16,17]	0.983	0.933	0.719	0.512	0.284	0.203	0.150	0.125
Steepest descent based method	0.854	0.622	0.561	0.309	0.186	0.052	0.015	0.005
Accelerated conjugate direction method	1.165	0.712	0.566	0.306	0.220	0.032	0.025	0.011

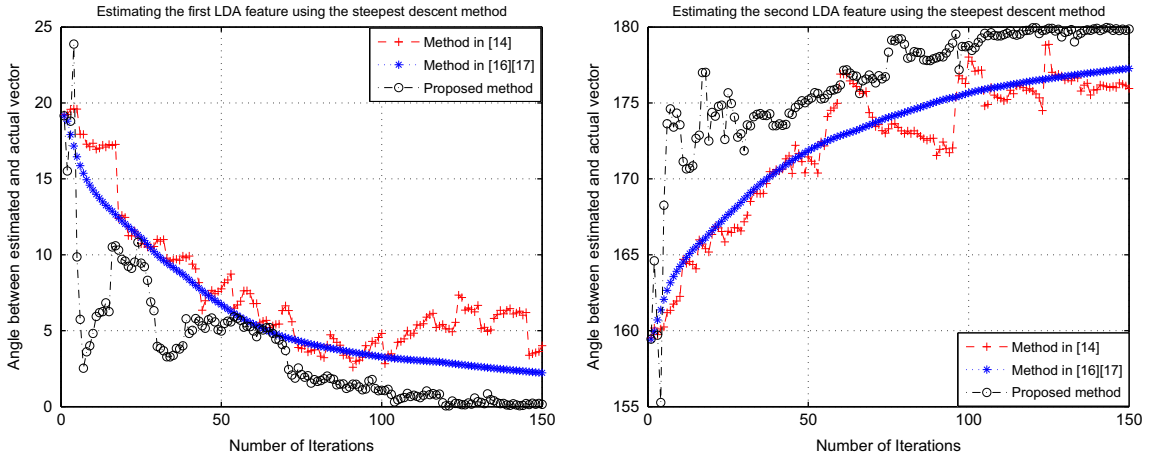

Fig. 5. This figure compares the performance of the steepest descent method with the algorithm in [14] and the gradient descent based algorithm in [16,17]. The left figure shows the angle between the estimated first LDA feature and the actual first LDA features as a function of number of iterations for different algorithms. The right figure shows the angle between the estimated second LDA feature and the actual second LDA features as a function of number of iterations for different algorithms

iterations comparing with the given algorithm in [14], and the gradient descent based algorithm in [16,17].

5.2. Iris data set

The Iris data set is a very popular data sets in the pattern recognition community.[11] The data set contains 50 samples from each of three species of iris, namely setosa, versicolor, and virginica. Four features from each sample were measured: the length and width of the sepals and the petals, in centimetres. Therefore, the input sequence consists of 150 observations of four-dimensional vectors. The sequence of four-dimensional vectors are used to train the proposed accelerated incremental $\mathbf{Q}^{-1/2}$ algorithm and the output of $\mathbf{Q}^{-1/2}$ algorithm (i.e., $\mathbf{\Sigma}_W^{-1/2}$) is used to extract two leading LDA features. For the $\mathbf{Q}^{-1/2}$ algorithm we set the initial step sizes for all algorithms to be 0.1. The algorithm in [14], and the gradient descent based algorithm in [16,17] use a decreasing step size given by $1/(10+i\times 0.15)$ and the proposed algorithms based on the steepest descent method and accelerated conjugate direction method find the optimal step size in each iteration. Fig. 4 compares the relative errors resulting from each algorithm on estimating $\mathbf{\Sigma}_W^{-1/2}$ as a function of number of iteration. Clearly, the proposed algorithm gives a good estimate of $\mathbf{\Sigma}_W^{-1/2}$ in fewer iterations compared with existing algorithms. The normalized errors of

[11] According to UC Irvine machine learning repository, the data set is the most popular set with 569,993 hits since 2007. These data can be downloaded at http://archive.ics.uci.edu/ml/datasets/Iris.

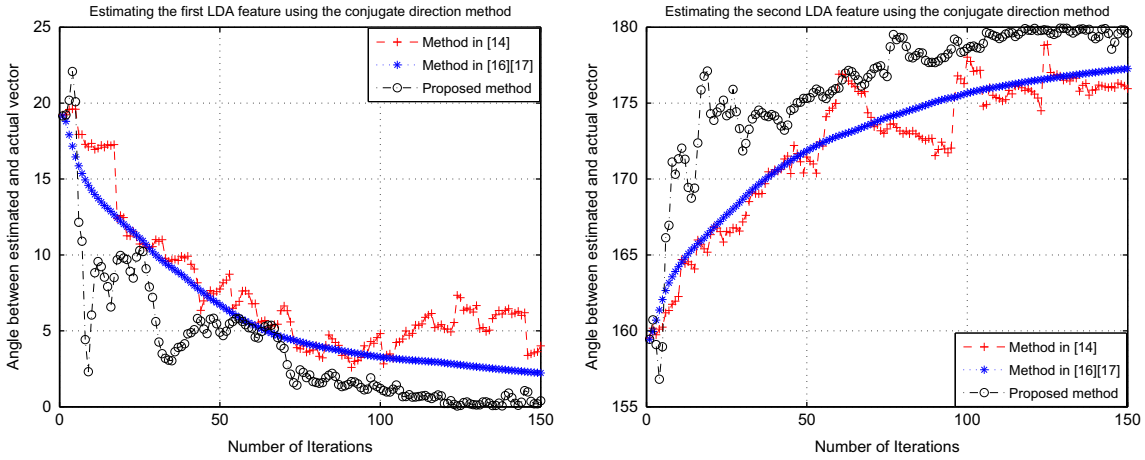

Fig. 6. This figure compares the performance of the conjugate direction method with the algorithm in [14] and the gradient descent based algorithm in [16,17]. The left figure shows the angle between the estimated first LDA feature and the actual first LDA features as a function of number of iterations for different algorithms. The right figure shows the angle between the estimated second LDA feature and the actual second LDA features as a function of number of iterations for different algorithms

Table 3
The angle between the first estimated LDA feature and the actual LDA feature as a function of the number of iterations for different algorithms.

Method	Number of iterations							
	2	5	20	40	75	100	130	150
Method in [14]	19.24	19.60	12.47	9.91	3.82	4.82	6.66	4.01
Gradient descent based method in [16,17]	18.80	16.45	12.03	8.36	4.26	3.28	2.63	2.22
Steepest descent based method	15.52	9.86	9.59	4.82	2.09	1.11	0.32	0.18
Accelerated conjugate direction method	19.23	20.09	9.80	4.80	2.48	1.29	0.21	0.35

Table 4
The angle between the second estimated LDA feature and the actual LDA feature as a function of the number of iterations for different algorithms.

Method	Number of iterations							
	2	5	20	40	75	100	130	150
Method in [14]	160.17	160.24	166.33	170.62	173.29	178.02	176.50	175.98
Gradient descent based method in [16,17]	159.96	162.06	166.60	170.46	174.05	175.64	176.75	177.27
Steepest descent based method	164.61	168.26	174.37	173.97	178.40	178.74	179.57	179.81
Accelerated conjugate direction method	160.72	158.95	174.30	174.16	176.89	178.55	179.86	179.63

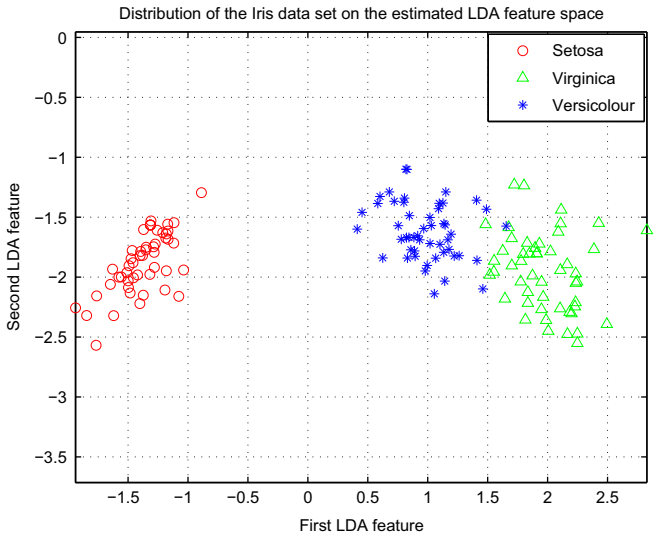

Fig. 7. Projection of four-dimensional samples of the Iris data set into estimated two-dimensional LDA-feature space using the proposed steepest descent method.

estimating $\mathbf{\Sigma}_W^{-1/2}$ for different numbers of iterations are shown in Table 2. Fig. 5 compares the performance of the proposed algorithms based on the steepest descent and conjugate direction methods to estimate the first LDA feature with algorithms given in [14,16,17]. Here, as the number of iterations increases, the first LDA feature estimated by the proposed technique moves towards the actual first LDA feature faster than existing techniques. Similar graphs are shown in Fig. 6 for estimating the second LDA feature. The angles between the estimated LDA features and actual LDA features resulting from the proposed algorithms and the algorithms in [14,16,17] are given in Tables 3 and 4. In Tables 3 and 4 the angles either converges to zero or to 180°. When the angle converges to zero, the algorithm gives us LDA direction $\mathbf{a}$, and when the angle converges to 180, the algorithm gives us the LDA direction $-\mathbf{a}$. Note that, in LDA both $\mathbf{a}$ and $-\mathbf{a}$ can be considered as the same LDA features. In other words, projection of an arbitrary vector $\mathbf{x}$ in direction of both $\mathbf{a}$ and $-\mathbf{a}$ gives the same vector.[12] Fig. 7 depicts the projection of the Iris data set into the estimated LDA

[12] The projection of $\mathbf{x}$ in the direction of $\mathbf{a}$ is $(\mathbf{x}\cdot\mathbf{a})\mathbf{a} = \|\mathbf{x}\|\|\mathbf{a}\|\cos(\theta)\mathbf{a}$ and in the direction of $-\mathbf{a}$ is similarly $(\mathbf{x}\cdot(-\mathbf{a}))(-\mathbf{a}) = \|\mathbf{x}\|\|\mathbf{a}\|\cos(180-\theta)(-\mathbf{a}) = \|\mathbf{x}\|\|\mathbf{a}\|\cos(\theta)\mathbf{a}$.

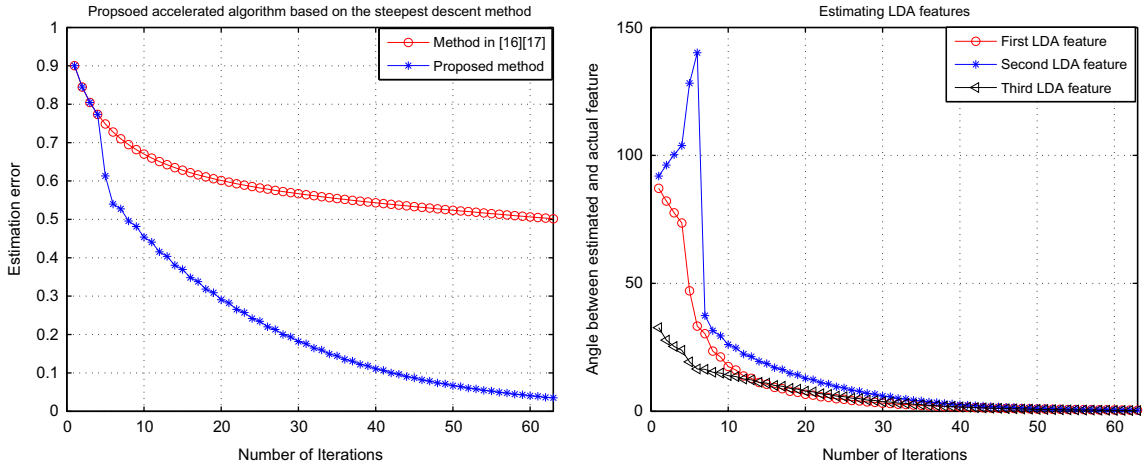

Fig. 8. The left part compares the performance of the proposed algorithm based on the steepest descent method to estimate $\Sigma_W^{-1/2}$ with the gradient descent algorithm given in [17]. The right side shows the angle between the estimated leading LDA features and actual leading LDA features as a function of the number of iterations.

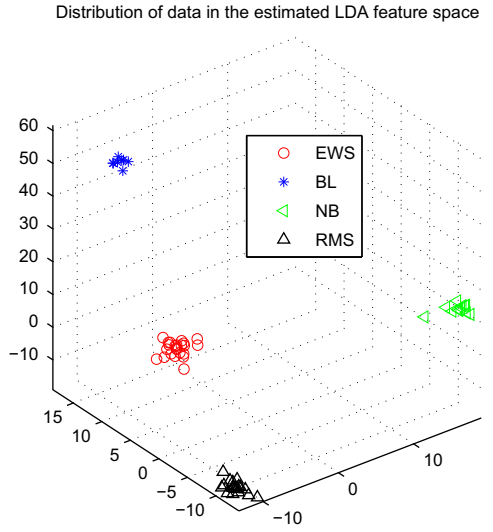

Fig. 9. Distribution of SRBCT data set in the estimate three dimensional LDA feature space. Although the dimensionality is reduced from 50 to 3, the four classes are linearly separable in the estimated feature space.

feature space using the proposed steepest descent method. It can be observed from Fig. 7 that although the dimensionality of the samples reduced from four to two, three classes are linearly separable.

5.3. SRBCT data set

The small round blue cell tumors (SRBCTs) data set [34] contains information of 63 samples and 2308 genes. The samples are distributed in four classes: 23 Ewing's sarcoma (EWS), 8 Burkitt's lymphoma (BL), 12 neuroblastoma (NB), and 20 rhabdomyosarcoma (RMS). Each class has widely differing prognoses and treatment options, making it extremely important that doctors are able to classify the tumor category quickly and accurately. Since the dimensionality of the input data is much bigger than the number of samples, the within-class scatter matrix (Σ_W) will be singular and the LDA features can therefore not be computed. To solve this problem, we first reduce the dimensionality of the data set to 50 by applying the PCA and projecting the data into leading 50 principal components (since the first 50 eigenvalues of the covariance matrix dominate the rest).[13] Then we use the sequence of 50-dimensional data to train the proposed algorithms. Fig. 8 compares the performance of the proposed algorithm based on the steepest descent method to estimate $\Sigma_W^{-1/2}$ with the algorithm given in [17]. The initial step size for both algorithms is empirically set to $\eta_0 = 0.008$. Fig. 8 shows that the proposed algorithm provides a low estimation error in fewer iterations by optimizing the learning rate in each iteration. Fig. 8 also shows the angle between the estimated three leading LDA features and the actual LDA features. It is clear from the right side of Fig. 8 that the angle between all three estimates and the actual LDA features becomes negligible after about 40 iterations. The projection of 50-dimensional samples into a 3-dimensional estimated LDA feature space is shown in Fig. 9. It can be observed from Fig. 9 that although the dimensionality is reduced from 50 into 3, the four classes are linearly separable.

5.4. Extended Yale Face Database B

To show the effectiveness of the proposed structure for incremental LDA feature extraction, we implement it on the extended Yale face database B[14] [35]. The extended Yale face database B contains face images of 28 individuals and around 64 near-frontal images under different illuminations per individual [36]. We selected 5 individuals with 64 images per individual (a total of 320 face images), cropped every face image to remove the background, and resized them to 32×32 pixels [37]. Therefore, each face image is represented by a 1024-dimensional (32×32) vector. The histogram for all face images is equalized in order to spreads out the intensity in an image and makes the resultant image as flat as possible [38]. Fig. 10 shows the cropped, histogram-equalized face images of five subjects under different poses and illumination conditions. Before applying the proposed algorithm, we reduce the dimensionality of the face images using the PCA algorithm. Computing the eigenvalues of the covariance matrix of the face images[15] reveals that the first three largest eigenvalues are 33.436, 8.965, and 6.737, but the fortieth eigenvalue drops to 0.082. Therefore, we only choose 40 significant eigenvectors corresponding to the largest eigenvalues and reduce the dimensionality of the face images to 40 by projecting them into the feature space spanned by the significant eigenvectors of the covariance matrix. Fig. 11 shows eigenfaces [39] corresponding to the 40 significant eigenvectors of the covariance matrix. The 40-dimensional vectors are fed into the proposed incremental LDA feature extraction algorithm sequentially.

[13] For this part, we assumed that the whole data set is available in advance, but after dimensionality reduction we trained the proposed algorithms using the sequential data.

[14] The Yale Face Database B is available online at http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html.

[15] The covariance matrix is computed using the vectorized representation of face images.

Fig. 10. Face images of 5 individuals that have been used in our simulations.

Fig. 11. Eigenfaces corresponding to the 40 significant eigenvectors of the covariance matrix of the vectorized face images.

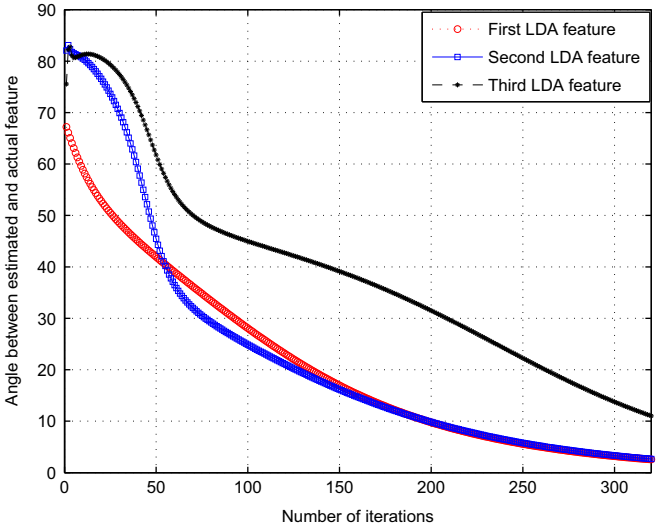

Fig. 12. The angle between the estimated LDA features and actual LDA features as a function of number of iterations.

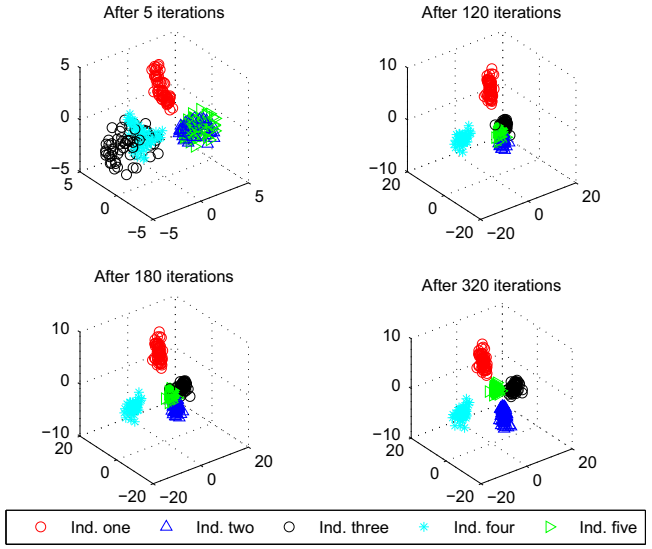

Fig. 13. Distribution of face images in the estimated LDA feature space after different number of iterations.

For the $\mathbf{Q}^{-1/2}$ algorithm, we start with the identity matrix and the estimate improves by observing new samples. Fig. 12 shows the angle between the estimated significant LDA features and actual ones as a function of number of iterations. It can be observed from Fig. 12 that as the number of iterations increases (i.e., the proposed algorithm observes more samples) the angle reduces and the proposed algorithm provides a better estimate of significant LDA features. For a better understanding of the performance of the proposed structure, the distribution of the face images in the estimated three-dimensional LDA feature space for different number of iterations is shown in Fig. 13. The top left side of Fig. 13 shows the projection of face images into estimated LDA feature space after just 5 iterations. It can be observed that five subjects are mixed and are not linearly separable, due to an

inaccurate estimate of LDA features. As the number of the iterations increases (i.e., the proposed structure observes more face images), the classes gradually start to separate from each other and the overlapping between them decreases. Finally, after 320 iterations, the proposed structure gives a reliable estimate of the significant LDA features and the five subjects are almost linearly separable as it is shown in the bottom right of Fig. 13.

6. Conclusion

Chatterjee and Roychowdhury showed that finding the LDA features incrementally involves computing $\mathbf{\Sigma}_W^{-1/2}$ using a fixed or decreasing step size [14]. The proposed technique in [14] suffered from low convergence rate. In this paper, we apply the steepest descent and conjugate direction methods on an explicit cost function to find the optimal step size in each iterations in order to accelerate the convergence rate of $\mathbf{\Sigma}_W^{-1/2}$ algorithm. Similar to [14], we combine the proposed accelerated $\mathbf{\Sigma}_W^{-1/2}$ algorithm with an adaptive PCA algorithm to derive the LDA features. We compare the performance of the proposed structure for incremental LDA feature extraction with the algorithm in [14], and the gradient descent based algorithm in [16,17]. The simulations results showed that the proposed algorithm provide a good estimate of the LDA features in fewer iterations compared to the other methods.

The proposed algorithms can be used for on-line applications where the whole data set is not available and is instead presented as a stream. As soon as a new observation is available, the proposed structure can update LDA features by simply using the old features and the new sample without having to run the algorithm on the entire data set.

Conflict of interest

None declared.

Appendix A

Let $\mathbf{A}\in\mathcal{C}$, then all leading principal minors of $\mathbf{A}$ are positive, and are continuous function of matrix entries [40]. Therefore, there exists $\epsilon>0$ such that by perturbing each entries of $\mathbf{A}$ by at most ϵ will still leave all the leading principal minors positive. Then $\mathbf{A}+\epsilon\mathbf{B}\in\mathcal{C}$ for every $\mathbf{B}\in\mathcal{C}$ such that $\|\mathbf{B}\|_2<1$ [40]. Therefore, $\mathcal{C}$ is an open convex set.

By expanding the cost function at $k+1$-th iteration and using (18), we have

$$\begin{aligned} J(\mathbf{W}_{k+1}) &= \tfrac{1}{3}\,\mathrm{Tr}(\mathbf{W}_{k+1}^3\mathbf{Q}) - \mathrm{Tr}(\mathbf{W}_{k+1}) + \tfrac{2}{3}\,\mathrm{Tr}(\mathbf{Q}^{-1/2}) \\ &= \tfrac{1}{3}\,\mathrm{Tr}\Big((\mathbf{W}_k+\eta_k\mathbf{G}_k)^3\mathbf{Q}\Big) - \mathrm{Tr}(\mathbf{W}_k+\eta_k\mathbf{G}_k) + \tfrac{2}{3}\,\mathrm{Tr}(\mathbf{Q}^{-1/2}), \end{aligned} \tag{A.1}$$

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

The cost function in (A.1) can be further simplified to

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

By taking the derivative of (A.2) with respect to the step size η_k and equating it to zero, we obtain

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

where $a_k=\mathrm{Tr}(\mathbf{G}_k^3\mathbf{Q}_{k+1})$, $b_k=2\,\mathrm{Tr}(\mathbf{W}_k\mathbf{G}_k^2\mathbf{Q}_{k+1})$, and $c_k=\mathrm{Tr}(\mathbf{W}_k^2\mathbf{G}_k\mathbf{Q}_{k+1})-\mathrm{Tr}(\mathbf{G}_k)$.

Appendix B

If we expand the matrix products, we get

$$\begin{aligned} J(\mathbf{W}_k) &= \tfrac{1}{3}\,\mathrm{Tr}\Big((\mathbf{W}_k+\alpha_k\mathbf{D}_k)^3\mathbf{Q}\Big) - \mathrm{Tr}(\mathbf{W}_k+\alpha_k\mathbf{D}_k) + \tfrac{2}{3}\,\mathrm{Tr}(\mathbf{Q}^{-1/2}) \\ &= \tfrac{1}{3}\,\mathrm{Tr}\Big((\mathbf{W}_k^3+3\alpha_k\mathbf{W}_k^2\mathbf{D}_k+3\alpha_k^2\mathbf{W}_k\mathbf{D}_k^2+\alpha_k^3\mathbf{D}_k^3)\mathbf{Q}\Big) \\ &\quad - \mathrm{Tr}(\mathbf{W}_k+\alpha_k\mathbf{D}_k) + \tfrac{2}{3}\,\mathrm{Tr}(\mathbf{Q}^{-1/2}). \end{aligned} \tag{B.1}$$

By taking the first derivative of (B.1) with respect to α_k and equating it to zero, we obtain

$$\frac{\partial J(\mathbf{W}_{k+1})}{\partial\alpha_k} = a_k\alpha_k^2 + b_k\alpha_k + c_k = 0, \tag{B.2}$$

where $a_k=\mathrm{Tr}(\mathbf{D}_k^3\mathbf{Q}_{k+1})$, $b_k=\tfrac{2}{3}\,\mathrm{Tr}\Big((\mathbf{W}_k\mathbf{D}_k^2+\mathbf{D}_k\mathbf{W}_k\mathbf{D}_k+\mathbf{D}_k^2\mathbf{W}_k)\mathbf{Q}_{k+1}\Big)$, and $c_k=\tfrac{1}{3}\,\mathrm{Tr}\Big((\mathbf{W}_k^2\mathbf{D}_k+\mathbf{W}_k\mathbf{D}_k\mathbf{W}_k+\mathbf{D}\mathbf{W}_k^2)\mathbf{Q}_{k+1}\Big)-\mathrm{Tr}(\mathbf{D}_k)$.

References

[1] A. Sharmaa, K.K. Paliwala, G.C. Onwubolu, Class-dependent PCA, MDC and LDA: a combined classifier for pattern classification, Pattern Recognit. 39 (7) (2006) 1215–1229.

[2] M. Kana, S. Shana, Y. Suc, D. Xub, X. Chen, Adaptive discriminant learning for face recognition, Pattern Recognit. 46 (9) (2013) 2497–2509.

[3] K. Dasa, Z. Nenadic, Approximate information discriminant analysis: a computationally simple heteroscedastic feature extraction technique, Pattern Recognit. 41 (5) (2008) 1548–1557.

[4] L. Jin, K. Ding, Z. Huang, Incremental learning of LDA model for chinese writer adaptation, Neurocomputing 73 (12) (2010) 1614–1623.

[5] E.K. Tanga, P.N. Suganthana, X. Yaob, A.K. Qina, Linear dimensionality reduction using relevance weighted LDA, Pattern Recognit. 38 (4) (2005) 485–493.

[6] Y.A. Ghassabeh, H.A. Moghaddam, A new incremental face recognition system, in: 4th IEEE workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 07), Dortmund, Germany, 2007, pp. 335–340.

[7] L.-F. Chen, H.-Y. Liao, M.-T. Ko, J.-C. Lin, G.-J. Yu, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognit. 33 (10) (2000) 713–1726.

[8] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2009.

[9] S. Pang, S. Ozawa, N. Kasabov, Incremental linear discriminant analysis for classification of data stream, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 35 (5) (2005) 905–914.

[10] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, V. Kumar, IDR/QR: an incremental dimension reduction algorithm via QR decomposition, IEEE Trans. Knowl. Data Eng. 17 (9) (2011) 1208–1222.

[11] M. Uray, D. Skocaj, P.M. Roth, A.L.H. Bischof, Incremental LDA learning by combining reconstructive and discriminative approaches, in: Proceedings of British Machine Vision Conference BMVC, University of Warwick, UK, 2007, pp. 272–281.

[12] T. Kim, S. Wong, B. Stenger, J. Kittler, R. Cipolla, Incremental linear discriminant analysis using sufficient spanning set approximation, in: Proceedings IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), University of Cambridge, Cambridge, UK, 2007, pp. 1–8.

[13] T. Kim, S. Wong, B. Stenger, J. Kittler, R. Cipolla, Incremental linear discriminant analysis using sufficient spanning sets and its applications, Int. J. Comput. Vis. 9 (2) (2011) 216–232.

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

[15] G.K. Demir, K. Ozmehmet, Online local learning algorithms for linear discriminant analysis, Pattern Recognit. Lett. 26 (4) (2005) 421–431.

[16] H.A. Moghaddam, K.A. Zadeh, Fast adaptive algorithms and network for class-separability features, Pattern Recognit. 36 (8) (2003) 1695–1702.

[17] H.A. Moghaddam, M. Matinfar, S.M.S. Sadough, K.A. Zadeh, Algorithms and networks for accelerated convergence of adaptive LDA, Pattern Recognit. 38 (4) (2005) 473–483.

[18] H.A. Moghaddam, M. Matinfar, Fast adaptive LDA using quasi-newton algorithm, Pattern Recognit. Lett. 28 (4) (2007) 613–621.

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

[20] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd edition, Academic Press, New York, 1990.

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

[22] T.D. Sanger, Optimal unsupervised learning in a single-layer linear FRRD forward neural network, Neural Netw. 2 (6) (1989) 459–473.
[23] E. Oja, Principal components, minor components, and linear neural networks, Neural Netw. 5 (6) (1992) 927–935.
[24] S. Ozawa, S. Pang, N. Kasabov, Incremental learning of feature space and classifier for on-line pattern recognition, Int. J. Knowl.-Based Intell. Eng. Syst. 10 (1) (2006) 57–65.
[25] J. Weng, Y. Zhang, W.-S. Hwang, Candid covariance-free incremental principal component analysis, IEEE Trans. Pattern Anal. Mach. Intell. 25 (8) (2003) 1034–1040.
[26] Y. Li, On incremental and robust subspace learning, Pattern Recognit. 37 (7) (2004) 1509–1518.
[27] A. Benveniste, M. Métivier, P. Priouret, Adaptive Algorithms and Stochastic Approximations, Springer, Berlin, 1990.
[28] G.H. Golub, C.F.V. Loan, Matrix Computations, Johns Hopkins University Press, 1996.
[29] A.N. Kolmogorov, S.V. Fomin, R.A. Silverman, Introductory Real Analysis, Dover Publications, 1975.
[30] M.T. Hagan, H.B. Demuth, M.H. Beale, Neural Network Design, Martin Hagan, 2002.
[31] J. Arora, Introduction to Optimum Design, 3rd edition, Academic Press, 2011.
[32] D.G. Luenberger, Linear and Nonlinear Programming, Springer, 2003.
[33] T. Okada, S. Tomita, An optimal orthonormal system for discriminant analysis, Pattern Recognit. 18 (2) (1984) 139–144.
[34] J. Khan, J.S. Wei, M. Ringnér, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, A.S.M.C. Peterson, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nat. Med. 7 (6) (2001) 673–679.
[35] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.
[36] D. Cai, X. He, Y. Hu, J. Han, T. Huang, Learning a spatially smooth subspace for face recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Machine Learning (CVPR'07), 2007.
[37] D. Cai, X. He, J. Han, H.J. Zhang, Orthogonal Laplacian faces for face recognition, IEEE Trans. Image Process. 15 (11) (2006) 3608–3614.
[38] N. Rajkumar, S. Vijayakumar, C. Murukesh, Intellectually combined face recognition using curvelet based principle component analysis for feature extraction and bayesian classifier, in: IEEE International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), Chennai, India, 2011.
[39] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3 (1) (1991) 71–86.
[40] A. Berman, N. Shaked-Monderer, Completely Positive Matrices, World Scientific, 2003.

Youness Aliyari Ghassabeh received the B.S. degree in Electrical Engineering from University of Tehran in 2004 and the M.S. degree from K.N. Toosi University of Technology in 2006. He received the Ph.D. degree in Mathematics and Engineering from the Department of Mathematics and Statistics, Queens University, Kingston, Canada in 2013. He is a postdoctoral fellow at Toronto Rehabilitation Institute, Toronto, Canada. His research interests include machine learning, statistical pattern recognition, image processing, source coding, and information theory.

Frank Rudzicz is a scientist at the Toronto Rehabilitation Institute with an assistant professor status in the department of Computer Science at the University of Toronto. He is the author of about 40 papers generally about natural language processing but focusing mostly on atypical speech and language as observed in individuals with physical disorders (e.g., cerebral palsy and Parkinson's disease) and in individuals with cognitive disorders (e.g. dementia and Alzheimer's disease). He is the founder and CEO of a company, Thotra Inc., that transforms speech signals to be more intelligible, a co-author on a best student paper award at Inter speech 2013, an Ontario Brain Institute entrepreneur, and winner of the Alzheimer's Society Young Investigator award. Dr. Rudzicz is secretary-treasurer of the joint ACL-ISCA special interest group on speech and language processing for assistive technologies and co-organizer of a number of its workshops, and he is an associate editor for special issues of the ACM Transactions on Accessible Computing and Computer Speech and Language.

Hamid Abrishami Moghaddam was born in Iran in 1964. He received the B.S. degree in electrical engineering from Amirkabir University of Technology, Tehran, Iran, in 1988, the M.S. degree in biomedical engineering from Sharif University of Technology, Tehran, in 1991 and the Ph.D. degree in biomedical engineering from Université de Technologie de Compiégne, Compiégne, France, in 1998. His research interests include pattern recognition, image processing, and machine vision, of which he has published more than 150 articles in scientific journals and conferences. Since 2004, he has been collaborating with Biophysics Laboratory at Université de Picardie Jules Verne in Amiens, France as an invited researcher on Medical Image Processing. He is a Professor of Biomedical Engineering at K.N. Toosi University of Technology, Tehran. Prof. Abrishami Moghaddam chaired the Machine Vision and Image Processing (MVIP2003) conference in Iran in 2003. He is the vice president of Iranian Society of Machine Vision and Image Processing (ISMVIP) and a member of editorial board of Iranian Journal of Biomedical Engineering (IJBME). He has also served as reviewer for several international journals in his fields of expertise.