Calcolo (2020) 57:18
https://doi.org/10.1007/s10092-020-00366-6

On general matrix exponential discriminant analysis methods for high dimensionality reduction

Wenya Shi[1] · **Youwei Luo**[1] · **Gang Wu**[1]

Received: 20 July 2019 / Revised: 10 May 2020 / Accepted: 12 May 2020 / Published online: 20 May 2020

Abstract
Recently, some matrix exponential-based discriminant analysis methods were proposed for high dimensionality reduction. It has been shown that they often have more discriminant power than the corresponding discriminant analysis methods. However, one has to solve some large-scale matrix exponential eigenvalue problems which constitutes the bottleneck in this type of methods. The main contribution of this paper is twofold: First, we propose a framework of fast implementation on general matrix exponential-based discriminant analysis methods. The key is to equivalently transform large-scale matrix computation problems into much smaller ones. On the other hand, it was mentioned in Wang et al. (IEEE Trans Image Process 23:920–930, 2014) that the exponential model is more reliable than the original one and suppresses the sensitivity to pertubations. However, the interpretation is only heuristic, and to the best of our knowledge, there is no theoretical justification for reliability and stability of the matrix discriminant analysis methods. To fill in this gap, the second contribution of our work is to provide stability analysis for the fast exponential discriminant analysis method from a theoretical point of view. Numerical experiments illustrate the numerical behavior of the proposed algorithm, and demonstrate that our algorithm is more stable than many state-of-the-art algorithms for high dimensionality reduction.

Keywords Dimensionality reduction · Small-sample-size problem · Matrix exponential · Large-scale eigenvalue problem · Stability analysis

Mathematics Subject Classification 65F15 · 65F10

This work is supported by the Natural Science Foundation of Jiangsu Province under grant BK20171185, and the Fundamental Research Funds for the Central Universities of China under grant 2019XKQYMS89.

✉ Gang Wu
gangwu@cumt.edu.cn; gangwu76@126.com

Extended author information available on the last page of the article

1 Introduction

Many objects in the real world are stored or represented as high dimensional data, and a direct processing of the data with regular methods is unfeasible and impractical [12, 35]. Dimensionality reduction is a technique to represent high dimensional data by their low-dimensional embedding, so that the low-dimensional data can be used effectively [10, 12, 20, 31, 35]. Nowadays, dimensionality reduction plays a crucial role in many application areas such as face recognition, machine learning, data mining, image reconstruction, information retrieval, and so on [12, 20, 24, 35, 38–40].

Principal Component Analysis (PCA) [19, 33] and Linear Discriminant Analysis (LDA) [2] are two extensively utilized approaches for dimensionality reduction. PCA is a unsupervised linear subspace transformation method that maximizes the variance of the transformed features in the projected subspace. LDA is a supervised linear subspace transformation method that encodes discriminant information by maximizing the between-class covariance, while minimizing the within-class covariance in the projected subspace. However, both PCA and LDA may fail to discover the underlying manifold structure [37], where the high dimensional image information lies in. In order to uncover the essential manifold structure of the data, some manifold learning methods have been proposed, to name a few, Laplacianfaces [17], Locality Preserving Projections (LPP) [16], Unsupervised Discriminant Projections (UDP) [43], Marginal Fisher Analysis (MFA) [42], Local Discriminant Embedding (LDE) [4], etc.

In Yan et al. [42], brought most of dimensionality reduction methods into a general framework called *graph embedding*. By embedding the high dimensional data into an optimal lower-dimensional space, the discriminant power obtained from graph embedding methods is usually stronger than classical classification methods. In the graph embedding framework, the neighbor relationship is measured by the artificially constructed adjacent graph. One concise criterion for feature extraction can be obtained from maximizing the ratio of the nonlocal scatter to the local scatter. Generally speaking, direct graph embedding and its extensions, such as linearization, kernelization and tensorization, all belong to this framework [42].

Most of the above approaches involve matrix inversion operation which may lead to matrix singularity problem in numerical computation, especially when the number of samples is (much) smaller than the dimension of samples. This is called the *small sample size problem* (SSS) or the *undersampled problem* [25]. In order to deal with the SSS problem, many techniques were proposed in decades. For example, the regularized method [11], PCA+LDA [2, 23, 33], the null-space method [5], LDA/QR [44], and so on.

Recently, some matrix exponential-based discriminant analysis methods were proposed to cure the drawback of SSS problem. For instance, Exponential Discriminant Analysis method (EDA) [46], Exponential Marginal Fisher Analysis (EMFA) [15, 37], Exponential Locality Preserving Projections (ELPP) [36], Exponential Local Discriminant Embedding (ELDE) [8], Exponential Elastic

Preserving Projections (EEPP) [45], and so on. In [37], a general exponential framework for dimensionality reduction was proposed. As exponential of any square matrix is nonsingular [14], exponential transformation cures the drawback of the SSS problem naturally. Furthermore, the matrix exponential can be considered as the cumulative sum of the similarity/transition matrices after the random walk over the feature similarity matrix, and the behavior of the decay property of matrix exponential is more significant in emphasizing small distance pairs [37]. Consequently, the matrix exponential discriminant analysis methods often have more discriminant power than their original counterparts, and they are competitive candidates for dimensionality reduction [1, 8, 9, 15, 36, 37, 39, 41, 45, 46].

In all the matrix exponential discriminant analysis methods, however, one has to compute two large-scale matrix exponentials and to solve a large-scale eigenproblem. The computational cost is prohibitively large for data with high dimension. The first contribution of this paper is to propose a fast implementation framework for all the above mentioned matrix exponential-based discriminant analysis methods. The key is to equivalently transform the original large matrix computation problems of size d into smaller ones of size n, where n is the number of training samples being much smaller than the data dimension d. So the proposed algorithms will be much cheaper than their original counterparts. Moreover, we show that the transformation is mathematically equivalent, so there will be no recognition rate lost in the accelerated algorithms.

As pointed out in [37, 46], the exponential model can be roughly interpreted as a random walk over the feature similarity matrix, which makes the feature similarity matrix more reliable and suppresses the sensitivity to the size of neighbors. The interpretation is just heuristic, however, to the best of our knowledge, there is no theoretical justification for reliability and stability of the matrix discriminant analysis methods. To fill in this gap, the second contribution of our work is to provide stability analysis for the fast exponential discriminant analysis methods from a theoretical point of view, and show why it is stable to the perturbation of the data matrix. This is also the main contribution of our work.

The remainder of this work is organized as follows. In Sect. 2, we give some preliminaries for this work and present a general framework for the matrix exponential discriminant analysis methods. In Sect. 3, we focus on accelerating the matrix exponential discriminant algorithms and propose a fast implementation framework. In Sect. 4, we provide stability analysis on the proposed algorithm. In Sect. 5, we perform some numerical experiments on some benchmark face databases to show the numerical behavior of our new algorithm. Section 6 gives some concluding remarks.

Let us introduce some notations. Given a matrix A, we denote by $\mathrm{tr}(A)$ the trace of A, and by $\exp(A)$ the matrix exponential of A. Let $\sigma_{\max}(A), \sigma_{\min}(A)$ be the maximal and the minimal (nonzero) singular values of A, and let $\mathrm{span}\{A\}$ be the space spanned by the columns of A. Denote by A^T, A^H the transpose and conjugate transpose of A, respectively, and by $A^\dagger$ the Moore-Penrose inverse of A. In this paper, $\mathbf{1}_i$ stands for the vector of all ones with dimension i; and r, L for the reducing dimension and for the number of training samples in each class, respectively. Let $\|A\|_F$ be the Frobenius norm of A, and let $\|A\|_2$ be the 2-norm of A, i.e., the largest singular value of A. Let A, B be two square matrices of the same size, then (A, B) represents

a matrix pair. Suppose that the two matrices X, Y have the same rows, then $[X,\ Y]$ denotes the matrix composed of the columns of X and Y, and $\sin\angle(X, Y)$ denotes the distance between the two subspaces $\text{span}\{X\}$ and $\text{span}\{Y\}$. Let I be the identity matrix and $\mathbf{0}$ be a zero matrix or vector, whose order are clear from the context.

2 Preliminaries

In this paper, we are interested in the following large generalized eigenvalue problem

$$XFX^T\mathbf{v} = \lambda XHX^T\mathbf{v}, \tag{2.1}$$

where $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ with each column $\mathbf{x}_i$ $(1 \le i \le n)$ being a sample, and the data dimension d is much larger than the number of samples, i.e., $d \gg n$. The matrices $F, H \in \mathbb{R}^{n\times n}$ are symmetric positive semidefinite matrices of order n. The large-scale eigenproblem (2.1) arises in many applications in machine learning and dimensionality reduction. Indeed, all the eigenproblems appeared in graph embedding methods, such as LDA [25], LDE [4], LPP [16], MFA [42], Fast and Orthogonal Locality Preserving Projections (FOLPP) [27], and Sparsity Preserving Projections (SPP) [26], can be reformulated as the form of (2.1) [20, 37, 42]. Moreover, in many popular dimension reduction methods such as PCA [19, 33], Multi-Dimensional Scaling (MDS) [32], ISOMAP [31], Neighborhood Preserving Projections (NPP) [21], Orthogonal Neighborhood Preserving Projection (ONPP) [22], and Orthogonal Locality Preserving Projections (OLPP) [21], all the eigenproblems involved are in the form of (2.1) [20]; see Table 1 in [42].

2.1 Linear discriminant analysis

LDA is one of notable subspace transformation methods for dimensionality reduction [25]. In this subsection, we illustrate how the generalized eigenvalues problem involved in LDA can be rewritten as the form of (2.1).

Suppose that the original data $X = [C_1, C_2, \ldots, C_K]$, where $C_i \in \mathbb{R}^{d\times n_i}$ is the data corresponding to the i-th class. We derive the mean vector of the data matrix as $\mathbf{m} = \sum_{i=1}^{n} \mathbf{x}_i/n = X\mathbf{1}_n/n$, and the mean vector of the i-th class as $\mathbf{m}_i = \sum_{\mathbf{x}_j\in C_i} \mathbf{x}_j/n_i$, $i = 1, 2, \ldots, K$. Then we define the within-class scatter matrix as $S_W = H_W H_W^T$ and the between-class scatter matrix as $S_B = H_B H_B^T$ [25], with

$$H_W = [C_1 - \mathbf{m}_1\mathbf{1}_{n_1}^T, C_2 - \mathbf{m}_2\mathbf{1}_{n_2}^T, \ldots, C_K - \mathbf{m}_K\mathbf{1}_{n_K}^T],$$

and

$$H_B = [\sqrt{n_1}(\mathbf{m}_1 - \mathbf{m}), \sqrt{n_2}(\mathbf{m}_2 - \mathbf{m}), \ldots, \sqrt{n_K}(\mathbf{m}_K - \mathbf{m})].$$

Moreover, the total scatter matrix $S_T = S_B + S_W$. LDA encodes discriminant information by maximizing the between-class scatter, and meanwhile minimizing the within-class scatter in the projected subspace. This resorts to solving the following optimization problem [2]

$$V^* = \arg \max_{\substack{V \in \mathbb{R}^{d\times r} \\ V^T V = I}} \operatorname{tr}\big((V^T S_W V)^{-1}(V^T S_B V)\big), \tag{2.2}$$

whose solution is obtained from solving the generalized eigenvalue problem as follows [2]

$$H_B H_B^T \mathbf{v} = \lambda H_W H_W^T \mathbf{v}. \tag{2.3}$$

We now show that (2.3) can be reformulated as a form of (2.1). Consider the diagonal matrix $D_K = diag(n_1, n_2, \ldots, n_K)$, then under the above notations, we have

$$\begin{aligned} H_B &= \big([\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K] - [\mathbf{m}, \mathbf{m}, \ldots, \mathbf{m}]\big) D_K^{\frac{1}{2}} \\ &= \big([\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K] - \mathbf{m}\mathbf{1}_K^T\big) D_K^{\frac{1}{2}}. \end{aligned} \tag{2.4}$$

Recall that $\mathbf{m}_i$ is the mean vector of i-th class, so it is a linear combination of data matrix X. Define

$$P = \begin{bmatrix} \mathbf{1}_{n_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{n_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{n_K} \end{bmatrix} \in \mathbb{R}^{n\times K}. \tag{2.5}$$

whose (i, j)-th entry is

$$P_{ij} = \begin{cases} 1, & \mathbf{x}_i \in C(j), \\ 0, & else. \end{cases} \tag{2.6}$$

Decompose the matrix column-by-column as $P = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_K]$, where $\mathbf{p}_i$ denotes the i-th column of P. Then we have $\mathbf{m}_i = \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j / n_i = X\mathbf{p}_i / n_i,\ i = 1, 2, \ldots, K$, and

$$[\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K] = XPD_K^{-1}. \tag{2.7}$$

From (2.4)–(2.7), we obtain

$$\begin{aligned} H_B &= (XPD_K^{-1} - X\mathbf{1}_n\mathbf{1}_K^T/n) D_K^{\frac{1}{2}} \\ &= X(PD_K^{-\frac{1}{2}} - \mathbf{1}_n\mathbf{1}_K^T D_K^{\frac{1}{2}}/n) \\ &= XL_B, \end{aligned}$$

where $L_B = PD_K^{-\frac{1}{2}} - \mathbf{1}_n\mathbf{1}_K^T D_K^{\frac{1}{2}}/n \in \mathbb{R}^{n\times K}$. As a result, the between-class matrix can be expressed as

$$S_B = H_B H_B^T = XL_B L_B^T X^T.$$

Next, we focus on the within-scatter matrix $S_W = H_W H_W^T$. Notice that

$$\begin{aligned} H_W &= [C_1 - \mathbf{m}_1\mathbf{1}_{n_1}^T, C_2 - \mathbf{m}_2\mathbf{1}_{n_2}^T, \ldots, C_K - \mathbf{m}_K\mathbf{1}_{n_K}^T] \\ &= [C_1, C_2, \ldots, C_K] - [\mathbf{m}_1\mathbf{1}_{n_1}^T, \mathbf{m}_2\mathbf{1}_{n_2}^T, \ldots, \mathbf{m}_K\mathbf{1}_{n_K}^T] \\ &= X - [\mathbf{m}_1\mathbf{1}_{n_1}^T, \mathbf{m}_2\mathbf{1}_{n_2}^T, \ldots, \mathbf{m}_K\mathbf{1}_{n_K}^T]. \end{aligned} \tag{2.8}$$

By (2.7) and (2.8),

$$\begin{aligned} H_W &= X - [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K]P^T \\ &= X - XPD_K^{-1}P^T \\ &= XL_W, \end{aligned}$$

where $L_W = I_n - PD_K^{-1}P^T \in \mathbb{R}^{n\times n}$. Thus, the within-class matrix S_W can be reformulated as

$$S_W = XL_WL_W^TX^T.$$

In conclusion, we have the following theorem.

Theorem 2.1 *Under the above notations, we have*

$$S_B = XL_BL_B^TX^T, \quad S_W = XL_WL_W^TX^T, \tag{2.9}$$

and (2.3) *can be rewritten as*

$$XL_BL_B^TX^T\mathbf{v} = \lambda XL_WL_W^TX^T\mathbf{v}, \tag{2.10}$$

where

$$L_B = PD_K^{-\frac{1}{2}} - \mathbf{1}_n\mathbf{1}_K^TD_K^{\frac{1}{2}}/n, \text{ and } L_W = I_n - PD_K^{-1}P^T. \tag{2.11}$$

Remark 2.1 Denote by $\widehat{T} = \mathbf{1}_n\mathbf{1}_n^T/n \in \mathbb{R}^{n\times n}$, $T_j = \mathbf{1}_{n_j}\mathbf{1}_{n_j}^T/n_j \in \mathbb{R}^{n_j\times n_j}$, and by the block diagonal matrix $T = diag(T_1, T_2, \ldots, T_K)$, it was pointed out that [6]

$$S_B = X(T - \widehat{T})X^T, \quad S_T = X(I - \widehat{T})X^T. \tag{2.12}$$

In addition, denote by $\bar{X} = X - \mathbf{m}\mathbf{1}_n^T$, we have that [20]

$$S_B = \bar{X}T\bar{X}^T, \quad S_W = \bar{X}(I - T)\bar{X}^T, \tag{2.13}$$

see also [42, Eqn(12)]. Note that (2.9) is different from (2.13), and we express H_B and H_W as a linear combination of the columns of X. Recall that $\bar{X}$ is a column rank-deficient matrix even when X is of full column rank.

2.2 A general exponential framework for dimensionality reduction

The main difference between the matrix exponential-based discriminant analysis methods and the classical discriminant analysis methods is that the former applies matrix exponential transformation on scatter matrices. More precisely, if we denote by

$$S_H = XHX^T, \quad S_F = XFX^T,$$

then in the matrix exponential-based discriminant analysis methods, such as EDA [46], EMFA [15, 37], ELPP [15], ELDE [8], and Exponential Unsupervised Discriminant Projections (EUDP) [37], all of them resort to solving the objective function as follows [1, 8, 9, 15, 36, 37, 39, 41, 46]

$$V^* = \arg \max_{\substack{V \in \mathbb{R}^{d\times r} \\ V^TV = I}} \operatorname{tr}\big((V^T\exp(S_H)V)^{-1}(V^T\exp(S_F)V)\big).$$

As S_F and S_H are symmetric, both $\exp(S_F)$ and $\exp(S_H)$ are positive definite. The matrix exponential-based discriminant analysis methods seek the projection matrix V^* via solving the following symmetrical generalized eigenvalue problem [1, 8, 9, 15, 36, 37, 39, 41, 45, 46]

$$\exp(S_F)\mathbf{v} = \lambda\exp(S_H)\mathbf{v}. \tag{2.14}$$

Algorithm 1 A framework for the exponential-based discriminant analysis algorithms

Require: the data matrix $X \in \mathbb{R}^{d\times n}$ $(d \gg n)$, and the reduced dimension r;
Ensure: projection matrix V;
1. Computing the scatter matrices S_H and S_F, and normalizing them with their Frobenius norm: $S_H = S_H/\|S_H\|_F$, $S_F = S_F/\|S_F\|_F$;
2. Computing $\exp(S_H)$ and $\exp(S_F)$;
3. Solving the generalized eigenproblem of the matrix pair $\big(\exp(S_F), \exp(S_H)\big)$: Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ be the eigenvectors corresponding to the r largest eigenvalues, and set $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r]$;
4. Orthogonalize the projection matrix: $V = orth(V)$.

A framework for the exponential-based discriminant analysis algorithms is presented in Algorithm 1. At the first glance, the computational complexities of the matrix exponential-based discriminant algorithms are prohibitively large and are much higher than those of the classical discriminant analysis algorithms. It will require $\mathcal{O}(d^3)$ flops to form the two large matrix exponentials $\exp(S_H)$ and $\exp(S_F)$ explicitly, and another $\mathcal{O}(d^3)$ flops to solve (2.14). Furthermore, we have to store the two large matrix exponentials which are d-by-d (possibly) dense matrices. Therefore, it is unfavorable to compute and store these large matrix exponentials explicitly, and to solve the large eigenproblem (2.14) directly. Therefore, it is necessary to seek new strategies to speedup the matrix exponential-based discriminant analysis algorithms in practical calculations.

3 A Fast Implementation on Matrix Exponential-based Discriminant Analysis Methods

In this section, we consider how to accelerate the matrix exponential-based discriminant analysis algorithms. The key is to equivalently transform the large matrix computation problems of size d into smaller ones of size n. Consequently, there is no need to form and store the two large matrix exponentials explicitly. To this aim, we first rewrite the matrices $\exp(S_H)$ and $\exp(S_F)$ as low-rank updates of the identity matrix, and then reduce the large eigenproblem (2.14) into smaller one.

3.1 Solving the large eigenvalue problem efficiently

The QR decomposition is a powerful tool for small-sample-size problems [25, 44]. Consider the "economical" QR decomposition [13] of the data matrix $X \in \mathbb{R}^{d\times n}$: $X = QR$, where $Q \in \mathbb{R}^{d\times n}$ is orthogonal and $R \in \mathbb{R}^{n\times n}$ is upper triangular. As $d \gg n$, the QR decomposition refers to the economized QR decomposition throughout this paper. We have

$$S_F = XFX^T = Q(RFR^T)Q^T = Q\widetilde{S}_F Q^T, \tag{3.1}$$

$$S_H = XHX^T = Q(RHR^T)Q^T = Q\widetilde{S}_H Q^T, \tag{3.2}$$

where

$$\widetilde{S}_F = RFR^T \quad \text{and} \quad \widetilde{S}_H = RHR^T \tag{3.3}$$

are two n-by-n matrices. The following theorem indicates that there is a close relationship between the eigenpairs of the matrix pencil $\big(\exp(S_F), \exp(S_H)\big)$ and those of $\big(\exp(\widetilde{S}_F), \exp(\widetilde{S}_H)\big)$.

Theorem 3.1 *Let $X = QR$ be the economized QR decomposition of the data matrix X, where $Q \in \mathbb{R}^{d\times n}$ is orthonormal and $R \in \mathbb{R}^{n\times n}$ is upper triangular. Denote by $\widetilde{S}_H = RHR^T \in \mathbb{R}^{n\times n}$ and by $\widetilde{S}_F = RFR^T \in \mathbb{R}^{n\times n}$. If $(\lambda, \mathbf{y})$ is an eigenpair of $\big(\exp(\widetilde{S}_F), \exp(\widetilde{S}_H)\big)$, then $(\lambda, Q\mathbf{y})$ is an eigenpair of $\big(\exp(S_F), \exp(S_H)\big)$, and vice versa.*

Proof It is easy to verify that

$$\exp(S_F) = I_d + Q\big(\exp(\widetilde{S}_F) - I_n\big)Q^T, \tag{3.4}$$

$$\exp(S_H) = I_d + Q\big(\exp(\widetilde{S}_H) - I_n\big)Q^T. \tag{3.5}$$

On one hand, if $\exp(\widetilde{S}_F)\mathbf{y} = \lambda\exp(\widetilde{S}_H)\mathbf{y}$, by (3.4)–(3.5), we have that

$$\begin{aligned}\exp(S_F)(Q\mathbf{y}) &= [I_d + Q\big(\exp(\widetilde{S}_F) - I_n\big)Q^T](Q\mathbf{y})\\ &= Q\exp(\widetilde{S}_F)\mathbf{y} = \lambda Q\exp(\widetilde{S}_H)\mathbf{y}\\ &= \lambda[I_d + Q\big(\exp(\widetilde{S}_H) - I_n\big)Q^T](Q\mathbf{y})\\ &= \lambda\exp(S_H)(Q\mathbf{y}).\end{aligned}$$

On the other hand, if $\exp(S_F)(Q\mathbf{y}) = \lambda\exp(S_H)(Q\mathbf{y})$, then it follows from (3.4)–(3.5) that

$$Q\exp(\widetilde{S}_F)\mathbf{y} = \lambda Q\exp(\widetilde{S}_H)\mathbf{y}.$$

Applying Q^T on both sides gives $\exp(\widetilde{S}_F)\mathbf{y} = \lambda\exp(\widetilde{S}_H)\mathbf{y}$, which completes the proof. □

Remark 3.1 By (3.1), one can reduce the d-by-d large-scale symmetric positive definite generalized eigenproblem of (S_H, S_F) into an n-by-n symmetric positive definite generalized eigenproblem of $(\widetilde{S}_H, \widetilde{S}_F)$.

3.2 A fast matrix exponential discriminant analysis algorithm

In practice, it is required to compute r ($r \leq n$) eigenpairs of (2.14), where r is the reducing dimension. By Theorem 3.1, we only need to solve the eigenproblems of the $n \times n$ symmetrical generalized eigenproblem with respect to the n-by-n matrix pencil $\big(\exp(\widetilde{S}_F), \exp(\widetilde{S}_H)\big)$, whose cost is in $\mathcal{O}(n^3)$ flops [13]. The main algorithm of this paper is described as follows.

Algorithm 2 A fast exponential discriminant analysis algorithm (Alg.2)

Require: the data matrix $X \in \mathbb{R}^{d\times n}$ ($d \gg n$), and the reducing dimension r;
Ensure: the projection matrix V;
1. Compute the economized QR decomposition of the data matrix: $X = QR$, where $Q \in \mathbb{R}^{d\times n}$ and $R \in \mathbb{R}^{n\times n}$;
2. Form the n-by-n matrices F and H, as well as $\widetilde{S}_F = RFR^T$, $\widetilde{S}_H = RHR^T$;
3. Normalize $\widetilde{S}_F$ and $\widetilde{S}_H$ with their Frobenius norm;
4. Solve the eigenproblem with respect to the matrix pencil $\big(\exp(\widetilde{S}_F), \exp(\widetilde{S}_H)\big)$;
5. Let $[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r]$ be the r dominant eigenvectors of above eigenproblem, then form the projection matrix $V = Q[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r]$, and orthogonalize its columns.

As was suggested in [15, 37, 46], to prevent overflow, it is necessary to normalize the scatter matrices S_F and S_H (say, by using their Frobenius norm) in Step 1 of Algorithm 1. However, it is unfavorable to form and store these two matrices as they are very large and dense. The following theorem indicates that it only needs to normalize $\widetilde{S}_F$ and $\widetilde{S}_H$ of size n in practice; see Step 3 of Algorithm 2.

Theorem 3.2 *Under the above notations, we have*

$$\exp\Big(\frac{S_F}{\| S_F \|_F}\Big) = I_d + Q\Big(\exp\Big(\frac{\widetilde{S}_F}{\| \widetilde{S}_F \|_F}\Big) - I_n\Big)Q^T, \tag{3.6}$$

$$\exp\Big(\frac{-S_H}{\| S_H \|_F}\Big) = I_d + Q\Big(\exp\Big(\frac{-\widetilde{S}_H}{\| \widetilde{S}_H \|_F}\Big) - I_n\Big)Q^T. \tag{3.7}$$

Proof We only prove (3.6), and the proof of (3.7) is similar. From (3.1) and the fact that $\|S_F\|_F = \|Q\widetilde{S}_F Q^T\|_F = \|\widetilde{S}_F\|_F$, we have

$$\begin{aligned}
\exp\Big(\frac{S_F}{\| S_F \|_F}\Big) &= I_d + \frac{Q\widetilde{S}_F Q^T}{\| \widetilde{S}_F \|_F} + \frac{Q\widetilde{S}_F^2 Q^T}{2!\,\| \widetilde{S}_F \|_F^2} + \cdots \\
&= I_d + Q\Big(\frac{\widetilde{S}_F}{\| \widetilde{S}_F \|_F} + \frac{\widetilde{S}_F^2}{2!\,\| \widetilde{S}_F \|_F^2} + \cdots\Big)Q^T \\
&= I_d + Q\Big(\exp\Big(\frac{\widetilde{S}_F}{\| \widetilde{S}_F \|_F}\Big) - I_n\Big)Q^T.
\end{aligned}$$

□

So far we refrain from forming and storing the large matrices S_F, S_H and $\exp(S_F)$, $\exp(S_H)$ in Algorithm 2. On one hand, we only need to store a $d \times n$ matrix Q and some $n \times n$ matrices in the new algorithm, and the main storage requirement of the algorithm is $\mathcal{O}(dn)$ as $d \gg n$, rather than $\mathcal{O}(d^2)$. On the other hand, it is only required to compute the economized QR decomposition of X once for all, in $\mathcal{O}(dn^2)$ flops, moreover, solving the small generalized eigenproblem $\exp(\widetilde{S}_F)\mathbf{y} = \lambda\exp(\widetilde{S}_H)\mathbf{y}$ needs $\mathcal{O}(n^3)$ flops, rather than $\mathcal{O}(d^3)$ flops. As a result, there is no need to form and store the two large matrix exponentials explicitly. Moreover, the transformations are mathematically equivalent, so the recognition rate as well as the standard derivation obtained from Algorithm 2 would be the same to those from Algorithm 1; see the numerical experiments performed in Sect. 5.

Remark 3.2 Recently, two inexact Krylov subspace algorithms, Arnoldi-EDA and Lanczos-EDA were proposed for solving the large matrix exponential eigenproblem arising in the Exponential Discriminant Analysis (EDA) method [39]. These two algorithms are based on Krylov subspace projection techniques and inexact solvers, in which the matrix exponentials need not to form or store explicitly, and the eigenpairs are solved only approximately.

However, in each step of the Arnoldi or Lanczos process, one has to perform a matrix exponential-vector multiplication of size d, in addition to Gram-Schmidt orthogonalizations. Thus, the proposed algorithm is cheaper than the two inexact Krylov subspace algorithms advocated in [39]; one refers to Sect. 5 for a comprehensive comparison of Algorithm 2 with these two inexact Krylov subspace algorithms.

4 Stability analysis on the proposed algorithm

In practical applications, however, the data is often perturbed or contaminated, and a natural question is [28]: whether the classification performance of the discriminant methods will be affected by the perturbations seriously or not? In this section, we focus on this problem and show the stability of the exponential discriminant methods. More precisely, we consider how the recognition rate will be affected by the perturbation on the original data.

Without loss of generality, we suppose that X is of full column rank and $\|X\|_F = 1$. Let $E \in \mathbb{R}^{d\times n}$ be a perturbation matrix with $\|E\|_F = \varepsilon \ll 1$, and denote by $\underline{X} = X + E$ the perturbed matrix of X, such that both the rank and the classification of the data points after perturbation are unchanged. Indeed, as the elements in the matrices F and H are determined by the classification of the data points, we make the assumption that both F and H are unchanged under the perturbation of E. Otherwise, the modifications in H and F will be in the order of $\mathcal{O}(1)$, which is not a perturbation problem any more; see, for example, (2.6).

Denote $\underline{S}_F = \underline{X}F\underline{X}^T$ and $\underline{S}_H = \underline{X}H\underline{X}^T$, then the symmetric generalized exponential eigenvalue problem (2.14) turns out to be

$$\exp(\underline{S}_F)\underline{\boldsymbol{v}} = \underline{\lambda}\exp(\underline{S}_H)\underline{\boldsymbol{v}}. \tag{4.1}$$

To analyze the stability of the fast matrix exponential-based algorithms, we want to establish the relationship between the eigenspace of (2.14) and that of (4.1). Let V be the (orthonormalized) solution matrix obtained from X by using Algorithm 2, and let $\underline{V}$ be that from $\underline{X}$, then the distance (in terms of Frobinus norm) between the subspaces span$\{V\}$ and span$\{\underline{V}\}$ is defined as [29]

$$\sin_F \angle(V, \underline{V}) = \frac{1}{\sqrt{2}}\|(I - VV^T)\underline{V}\|_F. \tag{4.2}$$

Let $\widehat{\mathbf{x}}_i$ be a sample from the training set, and $\widehat{\mathbf{y}}_j$ be a sample from the testing set. Then the nearest neighbour classifier gives class membership via investigating the Euclidean distance as follows [7]

$$d_{ij} = \|VV^T(\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}_j)\|_2 = \|V^T(\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}_j)\|_2, \tag{4.3}$$

where $\|V^T(\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}_j)\|_2$ is the 2-norm or the Euclidean norm of the vector $V^T(\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}_j)$. The following theorem indicates that one can use sine of the angle between the two subspaces span$\{V\}$ and span$\{\underline{V}\}$ as a criterion for stability of classification.

Theorem 4.1 [39] *Let $V, \underline{V} \in \mathbb{R}^{d\times r}$ be orthonormal matrices whose columns are the "exact" and "computed" solutions of (2.14) and (4.1), respectively. Denote by $d_{ij} = \|V^T(\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}_j)\|_2$ and $\underline{d}_{ij} = \|\underline{V}^T(\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}_j)\|_2$ the "exact" and the "computed" Euclidean distances, respectively. If $\|\widehat{\mathbf{x}}_j\|_2, \|\widehat{\mathbf{y}}_j\|_2 = 1$ and $\cos\angle(V, \underline{V}) \neq 0$, then*

$$\frac{\underline{d}_{ij} - 2\sin\angle(V,\underline{V})}{\cos\angle(V,\underline{V})} \le d_{ij} \le \underline{d}_{ij}\cos\angle(V,\underline{V}) + 2\sin\angle(V,\underline{V}). \tag{4.4}$$

Indeed, the above theorem shows that if $\sin\angle(V,\underline{V})$ is sufficiently small, then the $\{d_{ij}\}'s$ and the $\{\underline{d}_{ij}\}'s$ will be close to each other, where $\underline{d}_{ij} = \|\underline{V}^T(\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}_j)\|_2$ is an approximation to d_{ij}. Consequently, the recognition results of the "exact solution" and those of the "perturbed solution" are about the same. For more details, refer to [39].

Under the framework of Algorithm 2, we will divide the stability analysis into four steps.

(1) Firstly, according to Steps 1–2 of Algorithm 2, we consider the perturbation of E on $\widetilde{S}_H$ and $\widetilde{S}_F$.

If we denote by

$$\varepsilon_1 = \varepsilon/\|X\|_2, \quad G = E/\varepsilon_1,$$

and by $X(t) = X + tG$, then $X + tG$ has a unique QR decomposition $X(t) = Q(t)R(t)$ for all $|t| \le \varepsilon_1$ [3, Corollary 2.2]. Consider the (economized) QR decomposition of $\underline{X}$:

$$\underline{X} = \underline{Q}\underline{R} = (Q + \Delta Q)(R + \Delta R),$$

where $\Delta R = \varepsilon_1\dot{R}(0) + \mathcal{O}(\varepsilon_1^2)$, $\Delta Q = \varepsilon_1\dot{Q}(0) + \mathcal{O}(\varepsilon_1^2)$, and $\dot{R}(0)$, $\dot{Q}(0)$ are the first order derivatives of $R(t)$ and $Q(t)$ at $t = 0$, respectively. It was shown that [3]

$$\|\dot{R}(0)\|_F \le \sqrt{2}\kappa_2(R)\|Q^TG\|_F \quad \text{and} \quad \|\Delta Q\|_F \le \sqrt{2}\kappa_2(X)\varepsilon_1 + \mathcal{O}(\varepsilon_1^2),$$

where $\kappa_2(R)$ and $\kappa_2(X)$ are the 2-norm condition number of R and X, respectively. Note that Q is orthonormal, so $\kappa_2(R) = \kappa_2(X)$, moreover, as $\|X\|_F = 1$, we have

$$\kappa_2(X) = \|X\|_2\|X^\dagger\|_2 = \frac{\sigma_{\max}(X)}{\sigma_{\min}(X)} \le \frac{\|X\|_F}{\sigma_{\min}(X)} = \frac{1}{\sigma_{\min}(X)},$$

where $\sigma_{\max}(X)$ and $\sigma_{\min}(X)$ are the largest and the smallest (nonzero) singular values of X, respectively. Thus,

$$\|\dot{R}(0)\|_F \le \sqrt{2}\kappa_2(R)\|Q^TG\|_F \le \sqrt{2}\kappa_2(X)\|G\|_F \le \frac{\sqrt{2}\varepsilon}{\sigma_{\min}(X)\varepsilon_1}, \tag{4.5}$$

$$\|\Delta Q\|_F \le \sqrt{2}\kappa_2(X)\varepsilon_1 + \mathcal{O}(\varepsilon_1^2) \le \frac{\sqrt{2}\varepsilon}{\sigma_{\min}(X)\|X\|_2} + \mathcal{O}(\varepsilon^2). \tag{4.6}$$

Let $\widetilde{\underline{S}}_H = \underline{R}H\underline{R}^T$, then we have

$$\begin{aligned}\widetilde{\underline{S}}_H &= (R + \Delta R)H(R + \Delta R)^T \\ &= RHR^T + \varepsilon_1(RH\dot{R}^T(0) + \dot{R}(0)HR^T) + \varepsilon_1^2\dot{R}(0)H\dot{R}^T(0) + (HR^T + RH)\mathcal{O}(\varepsilon_1^2).\end{aligned} \tag{4.7}$$

If we denote by

$$\Delta_1 = \varepsilon_1(RH\dot{R}^T(0) + \dot{R}(0)HR^T) + \varepsilon_1^2\dot{R}(0)H\dot{R}^T(0) + (HR^T + RH)\mathcal{O}(\varepsilon_1^2),$$

then $\widetilde{\underline{S}}_H = \widetilde{S}_H + \Delta_1$. Combining the above relation with (4.5) and (4.7), we arrive at

$$\begin{aligned}
\|\Delta_1\|_F &= \|\widetilde{\underline{S}}_H - \widetilde{S}_H\|_F \\
&\le \varepsilon_1\|RH\dot{R}^T(0) + \dot{R}(0)HR^T\|_F + \|\varepsilon_1^2\dot{R}(0)H\dot{R}^T(0)\|_F + \|HR^T + RH\|_F\mathcal{O}(\varepsilon_1^2) \\
&\le 2\varepsilon_1\|R\|_2\|H\|_2\|\dot{R}(0)\|_F + \varepsilon_1^2\|\dot{R}(0)\|_F^2\|H\|_2 + 2\|H\|_2\|R\|_F\mathcal{O}(\varepsilon_1^2) \\
&\le 2\varepsilon_1\|X\|_2\|H\|_2\frac{\sqrt{2}\varepsilon}{\sigma_{\min}(X)\varepsilon_1} + \varepsilon_1^2\|H\|_2\frac{2\varepsilon^2}{\sigma_{\min}^2(X)\varepsilon_1^2} + \mathcal{O}(\varepsilon_1^2) \\
&\le 2\sqrt{2}\|H\|_2\frac{\varepsilon}{\sigma_{\min}(X)} + 2\|H\|_2\frac{\varepsilon^2}{\sigma_{\min}^2(X)} + \mathcal{O}(\varepsilon_1^2) \\
&= 2\sqrt{2}\|H\|_2\frac{\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma_{\min}^2(X)}\right),
\end{aligned} \tag{4.8}$$

where we use the fact that $\|R\|_2 = \|X\|_2 \le \|X\|_F = 1$.

(2) Secondly, in Step 3 of Algorithm 2, we need to normalize $\widetilde{\underline{S}}_H$. Let

$$\widehat{\underline{S}}_H = \widetilde{\underline{S}}_H/\|\widetilde{\underline{S}}_H\|_F, \quad \widehat{S}_H = \widetilde{S}_H/\|\widetilde{S}_H\|_F, \tag{4.9}$$

we investigate the perturbation of E on $\|\widehat{\underline{S}}_H - \widehat{S}_H\|_F$ and $\|\exp(\widehat{\underline{S}}_H) - \exp(\widehat{S}_H)\|_F$.

Suppose that $\varepsilon/\sigma_{\min}(X) \ll 1$, then $\| \triangle_1 \|_F = \mathcal{O}\big(\varepsilon/\sigma_{\min}(X)\big) \ll 1$, and there exists a constant $0 \ll \eta \le 1$ satisfying $\|\widetilde{\underline{S}}_H\|_F = \eta\|\widetilde{S}_H\|_F + \|\Delta_1\|_F$. Let $\Delta_2 = \widehat{\underline{S}}_H - \widehat{S}_H$, we have from (4.8) that

$$\begin{aligned}
\|\Delta_2\|_F &= \|\widehat{\underline{S}}_H - \widehat{S}_H\|_F \\
&= \frac{\left\|\|\widetilde{S}_H\|_F(\widetilde{S}_H + \Delta_1) - \|\widetilde{S}_H + \Delta_1\|_F\widetilde{S}_H\right\|_F}{\|\widetilde{\underline{S}}_H\|_F\|\widetilde{S}_H\|_F} \\
&= \frac{\left\|\widetilde{S}_H(\|\widetilde{S}_H\|_F - \|\widetilde{S}_H + \Delta_1\|_F) + \|\widetilde{S}_H\|_F\Delta_1\right\|_F}{(\eta\|\widetilde{S}_H\|_F + \|\Delta_1\|_F)\|\widetilde{S}_H\|_F} \\
&\le \frac{\left|\|\widetilde{S}_H\|_F - \|\widetilde{S}_H + \Delta_1\|_F\right|\|\widetilde{S}_H\|_F + \|\widetilde{S}_H\|_F\|\Delta_1\|_F}{\eta\|\widetilde{S}_H\|_F^2} \\
&\le \frac{\|\Delta_1\|_F\|\widetilde{S}_H\|_F + \|\widetilde{S}_H\|_F\|\Delta_1\|_F}{\eta\|\widetilde{S}_H\|_F^2} \le \frac{2\|\Delta_1\|_F}{\eta\|\widetilde{S}_H\|_F} \\
&\le \frac{4\sqrt{2}\|H\|_2}{\eta\|RHR^T\|_F}\cdot\frac{\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma_{\min}^2(X)}\right).
\end{aligned}$$

Let

$$\eta_1 = \frac{4\sqrt{2}\|H\|_2}{\eta\|RHR^T\|_F}, \tag{4.10}$$

then

$$\|\Delta_2\|_F \le \eta_1 \frac{\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right). \tag{4.11}$$

Next, we investigate the relationship between $\exp(\underline{\widehat{S}}_H)$ and $\exp(\widehat{S}_H)$, under the condition that $\varepsilon/\sigma_{\min}(X) \ll 1$. For any matrices $A, \Delta \in \mathbb{R}^{n\times n}$ and $t > 0$, it follows that [34]

$$\exp\big((A+\Delta)t\big) - \exp(At) = \int_0^t \exp\big((A(t-s)\big) \cdot \Delta \cdot \exp\big((A+\Delta)s\big) ds.$$

Note that $\|\exp(A)\| \le e^{\|A\|}$ for any subordinate matrix norm [14, pp.237], we have

$$\begin{aligned}
\|\exp(A+\Delta) - \exp(A)\|_F &= \left\| \int_0^1 \exp\big((A(1-s)\big) \cdot \Delta \cdot \exp\big((A+\Delta)s\big) ds \right\|_F \\
&\le \|\Delta\|_F \int_0^1 \|\exp(A(1-s))\|_2 \cdot \|\exp\big((A+\Delta)s\big)\|_2 ds \\
&\le \|\Delta\|_F \int_0^1 e^{\|A\|_2(1-s)} \cdot e^{\|(A+\Delta)\|_2 s} ds \\
&\le \|\Delta\|_F \int_0^1 e^{\|A\|_2 + \|\Delta\|_2 s} ds \\
&\le \|\Delta\|_F \cdot e^{\|A\|_2+\|\Delta\|_2} \le \|\Delta\|_F \cdot e^{\|A\|_2+\|\Delta\|_F}.
\end{aligned}$$

Specifically, if A is a symmetric positive definite (SPD) matrix, we conclude that

$$\|\exp(A+\Delta) - \exp(A)\|_F \le \|\Delta\|_F \cdot e^{\lambda_{\max}(A)+\|\Delta\|_F}. \tag{4.12}$$

Let $\Delta_3 = \exp(\underline{\widehat{S}}_H) - \exp(\widehat{S}_H)$, it follows from (4.11) and $\varepsilon/\sigma_{\min}(X) \ll 1$ that

$$\begin{aligned}
\|\Delta_3\|_F =& \|\exp(\underline{\widehat{S}}_H) - \exp(\widehat{S}_H)\|_F \\
\le& \|\Delta_2\|_F \cdot e^{\|\Delta_2\|_F + \lambda_{\max}(\widehat{S}_H)} \\
\le& e\eta_1 \frac{\varepsilon}{\sigma_{\min}(X)} e^{\eta_1\varepsilon/\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right),
\end{aligned}$$

where we use $\Delta_2 = \underline{\widehat{S}}_H - \widehat{S}_H$ and $\lambda_{\max}(\widehat{S}_H) \le \|\widehat{S}_H\|_F = 1$. As $\varepsilon/\sigma_{\min}(X) \ll 1$, we can apply the Taylor expansion of $e^{\eta_1\varepsilon/\sigma_{\min}(X)}$ at 0, which gives

$$e^{\eta_1\varepsilon/\sigma_{\min}(X)} = 1 + \frac{\eta_1\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right).$$

Thus,

$$\begin{aligned}\|\Delta_3\|_F &\le \frac{e\eta_1\varepsilon}{\sigma_{\min}(X)}\left(1+\frac{\eta_1\varepsilon}{\sigma_{\min}(X)}\right)+\mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right)\\ &\le e\eta_1\frac{\varepsilon}{\sigma_{\min}(X)}+e\eta_1^2\frac{\varepsilon^2}{\sigma^2_{\min}(X)}+e\eta_1\mathcal{O}\left(\frac{\varepsilon^3}{\sigma^3_{\min}(X)}\right)\\ &= e\eta_1\frac{\varepsilon}{\sigma_{\min}(X)}+\mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right).\end{aligned} \tag{4.13}$$

Analogously, let $\widetilde{\underline{S}}_F \equiv \underline{R}F\underline{R}^T$, $\widehat{\underline{S}}_F = \widetilde{\underline{S}}_F/\|\widetilde{\underline{S}}_F\|_F$, $\widehat{S}_F = \widetilde{S}_F/\|\widetilde{S}_F\|_F$, $\Delta_4 \equiv \exp(\widehat{\underline{S}}_F) - \exp(\widehat{S}_F)$, and

$$\eta_2 = \frac{4\sqrt{2}\|F\|_2}{\widehat{\eta}\|RFR^T\|_F}, \tag{4.14}$$

where $0 \ll \widehat{\eta} \le 1$ satisfying $\|\widetilde{\underline{S}}_F\|_F = \widehat{\eta}\|\widetilde{S}_F\|_F + \|\widetilde{\underline{S}}_F - \widetilde{S}_F\|_F$. We can prove that

$$\|\Delta_4\|_F = \|\exp(\widehat{\underline{S}}_F) - \exp(\widehat{S}_F)\|_F \le e\eta_2\frac{\varepsilon}{\sigma_{\min}(X)}+\mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right). \tag{4.15}$$

(3) Thirdly, we focus on the distance between the eigenspace of $\left(\exp(\widehat{S}_F), \exp(\widehat{S}_H)\right)$ and that of $\left(\exp(\widehat{\underline{S}}_F), \exp(\widehat{\underline{S}}_H)\right)$.

We first need the following theorem:

Theorem 4.2 [30] *Let the definite pair (J, S) be decomposed as following*

$$\begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix} J[Y_1, Y_2] = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix},$$

and

$$\begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix} S[Y_1, Y_2] = \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix},$$

where Y_1 and Y_2 have orthonormal columns and the column of Y_1 span an eigenspace of (J, S). Let the analogous decomposition be given for the pair $(\widetilde{J}, \widetilde{S}) = (J + \Delta J)(S + \Delta S)$. Set

$$\delta \equiv \min_{i,j}\{\rho((\alpha_i, \beta_i), (\widetilde{\alpha}_j, \widetilde{\beta}_j)) : (\alpha_i, \beta_i) \in \lambda(J_1, S_1), (\widetilde{\alpha}_j, \widetilde{\beta}_j) \in (\widetilde{J}_2, \widetilde{S}_2)\},$$

if $\delta > 0$, then

$$\sin_F\angle(Y_1, \widetilde{Y}_1) \le \frac{\sqrt{\|J^2 + S^2\|_2}}{c(J, S)\cdot c(\widetilde{S}, \widetilde{J})}\cdot\frac{\sqrt{\|\Delta J\cdot Y_1\|_F^2 + \|\Delta S\cdot Y_1\|_F^2}}{\delta}, \tag{4.16}$$

where

$$\rho\big((\alpha,\beta),(\widetilde{\alpha},\widetilde{\beta})\big)=\frac{|\alpha\widetilde{\beta}-\beta\widetilde{\alpha}|}{\sqrt{(|\alpha|^2+|\beta|^2)(|\widetilde{\alpha}|^2+|\widetilde{\beta}|^2)}},$$

and

$$c(J,S)=\min_{\|\mathbf{z}\|_2=1}\left|\mathbf{z}^H(J+iS)\mathbf{z}\right|>0.$$

is a Crawford number [30].

Recall that both $\big(\exp(\widehat{S}_F),\exp(\widehat{S}_H)\big)$ and $\big(\exp(\underline{\widehat{S}}_F),\exp(\underline{\widehat{S}}_H)\big)$ are definite matrix pencils. According to [29, pp.79–80], let the columns of the orthonormal matrices Z_1 and $\underline{Z}_1$ span an eigenspace of $\big(\exp(\widehat{S}_F),\exp(\widehat{S}_H)\big)$ and $\big(\exp(\underline{\widehat{S}}_F),\exp(\underline{\widehat{S}}_H)\big)$, respectively. Then there exist $Z_2,\underline{Z}_2$ such that $[Z_1,Z_2]$, $[\underline{Z}_1,\underline{Z}_2]$ are nonsingular and $\underline{Z}_1,\underline{Z}_2$ are orthonormal, such that

$$\begin{bmatrix}Z_1^H\\Z_2^H\end{bmatrix}\exp(\widehat{S}_F)[Z_1,Z_2]=\begin{bmatrix}F_1&0\\0&F_2\end{bmatrix},\ \begin{bmatrix}Z_1^H\\Z_2^H\end{bmatrix}\exp(\widehat{S}_H)[Z_1,Z_2]=\begin{bmatrix}H_1&0\\0&H_2\end{bmatrix},$$

and

$$\begin{bmatrix}\underline{Z}_1^H\\\underline{Z}_2^H\end{bmatrix}\exp(\underline{\widehat{S}}_F)[\underline{Z}_1,\underline{Z}_2]=\begin{bmatrix}\underline{F}_1&0\\0&\underline{F}_2\end{bmatrix},\ \begin{bmatrix}\underline{Z}_1^H\\\underline{Z}_2^H\end{bmatrix}\exp(\underline{\widehat{S}}_H)[\underline{Z}_1,\underline{Z}_2]=\begin{bmatrix}\underline{H}_1&0\\0&\underline{H}_2\end{bmatrix}.$$

In the terminology of Theorem 4.2, we denote by

$$\xi_1=\frac{\sqrt{\|\exp^2(\widehat{S}_H)+\exp^2(\widehat{S}_F)\|_2}}{c\big(\exp(\underline{\widehat{S}}_F),\exp(\underline{\widehat{S}}_H)\big)\cdot c\big(\exp(\widehat{S}_F),\exp(\widehat{S}_H)\big)},$$

and note that

$$\begin{aligned}\sqrt{\|\exp^2(\widehat{S}_H)+\exp^2(\widehat{S}_F)\|_2}&\le\sqrt{\|\exp(\widehat{S}_H)\|_2^2+\|\exp(\widehat{S}_F)\|_2^2}\\&\le\sqrt{e^{2\lambda_{\max}(\widehat{S}_H)}+e^{2\lambda_{\max}(\widehat{S}_F)}}\\&\le\sqrt{2}e,\end{aligned}$$

where $e=\exp(1)$, moreover,

$$\begin{aligned}c^2\big(\exp(\widehat{S}_F),\exp(\widehat{S}_H)\big)&=\min_{\|\mathbf{z}\|_2=1}\left(\left[\mathbf{z}^H\exp(\widehat{S}_F)\mathbf{z}\right]^2+\left[\mathbf{z}^H\exp(\widehat{S}_H)\mathbf{z}\right]\right)^2\\&\ge\lambda_{\min}^2\Big(\exp(\widehat{S}_F)\Big)+\lambda_{\min}^2\Big(\exp(\widehat{S}_H)\Big)\\&\ge 2,\end{aligned}\tag{4.17}$$

where we use the fact that both $\widehat{S}_H$ and $\widehat{S}_F$ are semi-positive definite. Similarly, we have that $c^2\big(\exp(\underline{\widehat{S}}_F),\exp(\underline{\widehat{S}}_H)\big)\ge 2$. As a result,

$$\xi_1 \leq \frac{e}{\sqrt{2}}. \tag{4.18}$$

On the other hand, (4.13) and (4.15) yield

$$\begin{aligned}\sqrt{\|\Delta_3 \cdot Z_1\|_F^2 + \|\Delta_4 \cdot Z_1\|_F^2} &\leq \sqrt{\|\Delta_3\|_F^2 + \|\Delta_4\|_F^2} \\ &\leq \sqrt{2}\max\{\|\Delta_3\|_F, \|\Delta_4\|_F\} \\ &\leq \sqrt{2}e\max\{\eta_1, \eta_2\}\frac{\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma_{\min}^2(X)}\right).\end{aligned} \tag{4.19}$$

Let

$$\delta_1 = \min_{i,j}\left\{\rho((\alpha_i, \beta_i), (\widetilde{\alpha}_j, \widetilde{\beta}_j)) : (\alpha_i, \beta_i) \in \lambda(F_1, H_1), (\widetilde{\alpha}_j, \widetilde{\beta}_j) \in \lambda(\underline{F}_2, \underline{H}_2)\right\}, \tag{4.20}$$

if $\delta_1 > 0$, we obtain from Theorem 4.2, (4.18) and (4.19) that

$$\begin{aligned}\sin_F\angle(Z_1, \underline{Z}_1) &\leq \delta_1^{-1}\xi_1\sqrt{\|\Delta_3 \cdot Z_1\|_F^2 + \|\Delta_4 \cdot Z_1\|_F^2} \\ &\leq \delta_1^{-1}e^2\max\{\eta_1, \eta_2\}\frac{\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma_{\min}^2(X)}\right).\end{aligned} \tag{4.21}$$

(iv) Finally, we determine the distance between the eigenspace of the matrix pair $\big(\exp(S_F), \exp(S_H)\big)$ and that of $\big(\exp(\underline{S}_F), \exp(\underline{S}_H)\big)$.

Indeed,

$$\begin{aligned}\sin_F\angle(QZ_1, \underline{QZ}_1) &= \frac{1}{\sqrt{2}}\|(I_d - QZ_1Z_1^TQ^T)\underline{QZ}_1\|_F \\ &= \frac{1}{\sqrt{2}}\|(I_d - QZ_1Z_1^TQ^T)(Q + \Delta Q)\underline{Z}_1\|_F \\ &\leq \frac{1}{\sqrt{2}}\|(I_d - QZ_1Z_1^TQ^T)Q\underline{Z}_1\|_F + \frac{1}{\sqrt{2}}\|(I_d - QZ_1Z_1^TQ^T)\Delta Q\underline{Z}_1\|_F \\ &\leq \sin_F\angle(QZ_1, Q\underline{Z}_1) + \frac{1}{\sqrt{2}}\|\Delta Q\|_F \\ &= \sin_F\angle(Z_1, \underline{Z}_1) + \frac{1}{\sqrt{2}}\|\Delta Q\|_F.\end{aligned} \tag{4.22}$$

By (4.6) and (4.21),

$$\sin_F\angle(QZ_1, \underline{Q}\underline{Z}_1) \le \delta_1^{-1}e^2\max\{\eta_1,\eta_2\}\frac{\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right) + \frac{\varepsilon}{\sigma_{\min}(X)\|X\|_2} + \mathcal{O}(\varepsilon^2)$$
$$= \left(\delta_1^{-1}e^2\max\{\eta_1,\eta_2\} + \sigma^{-1}_{\max}(X)\right)\frac{\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right).$$

In summary, we have the main theorem in this paper:

Theorem 4.3 *Under the above notations and assumptions, if* $\delta_1 > 0$, *then*

$$\sin_F\angle(V, \underline{V}) \le \left(\delta_1^{-1}e^2\max\{\eta_1,\eta_2\} + \sigma^{-1}_{\max}(X)\right)\frac{\varepsilon}{\sigma_{\min}(X)} + \mathcal{O}\left(\frac{\varepsilon^2}{\sigma^2_{\min}(X)}\right) \tag{4.23}$$

where $V = QZ_1$, $\underline{V} = \underline{Q}\underline{Z}_1$ *are orthonormal bases for the eigenspaces of the definite pairs* $\left(\exp(\underline{S}_F), \exp(\underline{S}_H)\right)$ *and* $\left(\exp(S_F), \exp(S_H)\right)$, *respectively*.

Remark 4.1 Given a perturbation matrix E to the data matrix X, whose norm is in the order of ε, Theorem 4.3 indicates that the perturbation to the "exact" solution V will be in the order of $\varepsilon/\sigma_{\min}(X)$. Thus, if $\varepsilon \ll 1$ and $\sigma_{\min}(X) \gg 0$, we have $\sin_F\angle(V, \underline{V}) \ll 1$, and the approximation obtained from Algorithm 2 will be insensitive to the perturbation.

On the other hand, we note that the upper bound provided in Theorem 4.3 may be not sharp and even pessimistic in practice. However, it reveals that the stability of matrix exponential discriminant analysis methods is closely related to the value of $\varepsilon/\sigma_{min}(X)$, and thus $\kappa(X) = \sigma_{\max}(X)/\sigma_{\min}(X)$ could be used as a condition number to the distance between the two subspaces span$\{V\}$ and span$\{\underline{V}\}$.

5 Numerical experiments

In this section, we carry out some numerical experiments to illustrate the numerical behavior and demonstrate the efficiency of Algorithm 2. One refers to [1, 8, 9, 15, 36, 37, 39, 41, 46] for superiority of the exponential-based methods over many state-of-the-art methods for recognition. The aim of this section is two-fold. First, we show that Algorithm 2 runs much faster than its original counterpart Algorithm 1, while the classification accuracy and the standard deviation of the former is comparable to that of the latter. Second, we show the stability of Algorithm 2 over many popular algorithms for dimensionality reduction and face recognition.

Four benchmark face databases, Yale[1], CMU PIE[2], AR[3] and Feret[4] are used. All the experiments are run on a Hp workstation with 16 cores double Intel(R)Xeon(R) E5-2640 v3 processors, and with CPU 2.60 GHz and RAM 128 GB. The operation

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.
[2] http://www.ri.cmu.edu/projects/project_418.html.
[3] http://rvl1.ecn.purdue.edu/aleix/aleix_face_DB.html.
[4] http://www.nist.gov/itl/iad/ig/colorferet.cfm.

system is 64-bit Windows 10. All the numerical results are obtained from using MATLAB R2018b. In all the algorithms, we use the nearest neighbor classifier (NN) [7] for classification, in which the distance is chosen as the Euclidean distance. In all the matrix exponential related algorithms, we use the MATLAB built-in functions `expm.m` to compute the matrix exponential, and make use of `eig.m` to solve the eigenvalue problem. Moreover, we exploit the MATLAB built-in functions `qr.m` and `orth.m` for the (economized) QR decomposition and for orthonormalizing the projection matrix V, respectively.

In each experiment, we randomly select L images from each class as the training set, and the rest of the images are used as the testing set. That is, there are $n = KL$ images in the training set, where K is the number of classes. Each experiment will be repeated for 10 times, and the numerical results, i.e., the CPU time in seconds, the recognition rate and the standard deviation (SD), are the mean values from the 10 runs.

5.1 Efficiency of the proposed algorithm

In this subsection, we aim to show the superiority of Algorithm 2 over Algorithm 1. We apply our fast implementation on the following matrix exponential-based discriminant analysis methods proposed recently:

- EDA: the Matrix Exponential Discriminant Analysis method proposed in [46]. The application of Algorithm 2 on EDA is denoted by FEDA (Alg. 2). As a comparison, we also run the two inexact Krylov-type algorithms, i.e., Arnoldi-EDA and Lanczos-EDA proposed in [39]. The stopping criterion for the two algorithms is chosen as $\epsilon = 10^{-4}$.
- EMFA: the Exponential Marginal Fisher Analysis method proposed in [15, 37]. The application of Algorithm 2 on EMFA is denoted by FEMFA (Alg. 2).
- ELPP: the Exponential Locality Preserving Projection method proposed in [36]. The application of Algorithm 2 on ELPP is denoted by FELPP (Alg. 2).
- ELDE: the Exponential Local Discriminant Embedding method proposed in [8]. The application of Algorithm 2 on ELPP is denoted by FELDE(Alg.2).

In this experiment, we choose $K_1 = L - 1$ and $K_2 = 2K_1$ in MFA, EMFA and FEMFA (Alg. 2), and use $K_1 = L - 1$ and $K_2 = 10$ in LDE, ELDE and FELDE(Alg.2), where K_1 and K_2 are the number of nearest neighbors in the same class and different classes, respectively.

(1) *First, we illustrate the superiority of Algorithm 2 over its original counterpart as well as the two inexact Krylov-type algorithms.*

There are total 165 images for 15 persons in the Yale facial database, with 11 images for each individual. The images demonstrate variation with the following expressions or configurations: (1) lighting: center light, left light, and right light; (2) with/without glasses; (3) facial expressions: normal, happy, sad, sleepy, surprised, and winkling. The original size of images is 320×243 pixels, and we crop the images to $d = 64 \times 64$ and 100×100 pixels in this example. We randomly pick

$L = 2,\ 3,\ 4,\ 5$ images from each class as the training set, and the remaining images are used as the testing set. The data dimension d is chosen as 4096 and 10000, respectively, the number of classes $K = 15$, and we choose the reducing dimension $r = K - 1 = 14$.

There are eighteen algorithms on this problem altogether: EDA and its fast implementation FEDA as well as its two inexact Krylov-type algorithms Arnoldi-EDA and Lanczos-EDA; EMFA, ELDE, ELPP and their fast implementations. In order to measure the times and accuracy when PCA is used as a preprocessing step on the exponential methods, the PCA plus exponential graph embedding methods are also run on this problem [28], i.e, PCA + EDA, PCA + LDA EMFA, PCA + LDA ELDE, PCA + LDA ELPP, in which we fix the dimensionality reduction of PCA to $n - 1$. Notice that Algorithm 2 and PCA plus exponential graph embedding methods have comparable amount of computational cost. We also list the numerical results of the original PCA plus graph embedding methods algorithms (PCA + LDA LDA, PCA + LDA MFA, PCA + LDA LDE, PCA + LDA LPP) in our experiment, where we preserve 99% energy in the PCA stage. Table 1 lists the numerical results.

It is obvious to see from Table 1 that our proposed algorithms are very effective. For example, as $d = 10{,}000$, the four matrix exponential-based algorithms are very slow, which used about 166 s. As a comparison, the four fast algorithms run much faster than their original counterparts, especially when the dimension d is large. Indeed, all the fast algorithms require less than 0.1 s, implying that our new algorithms are over 1660 times faster than the original algorithms. This is because our fast algorithms only need to compute matrix exponentials and solve eigenvalue problems of $n \times n$ matrices, instead of $d \times d$ matrices that are required in their original counterparts. Specifically, the proposed FEDA (Alg.2) is about three times faster than the two Krylov-type algorithms Arnoldi-EDA and Lanczos-EDA. On the other hand, we observe that the recognition rates and the standard deviations of the fast algorithms and their original counterparts are the same. This is because the fast algorithms and their original algorithms are mathematically equivalent.

Furthermore, it is seen that cropping the original images may lose some useful information and thus may result in a low recognition accuracy. For instance, the recognition rate of the exponential methods is about 86% when $d = 10000, L = 2$, while it decreases to about 75% as $d = 4096, L = 2$. Therefore, it is interesting to consider high dimensionality reduction problems for the (uncropped) data. In this case, our fast implementation is preferable as d is large.

For this Yale data base, we see that the CPU timings of PCA plus exponential graph embedding algorithms and PCA plus graph embedding algorithms are comparable, which is a little faster than Algorithm 2. This is because the PCA-preprocessing algorithms construct the adjacency matrices by using the "reduced" data matrices instead of the "original" data matrices; see also Table 2. On the other hand, the PCA plus exponential algorithms and Algorithm 2 share about the same recognition rates which are a little lower than those from the PCA plus exponential graph embedding algorithms in some cases. However, as will be shown latter, the exponential-based algorithms are more stable to perturbations.

(2) *Next, we show the feasibility of our new algorithms on data with very high dimensionality.*

Table 1 Example Sect. 5.1 (1): experiments on Yale database,$K = 15,\ n = KL,\ r = 14$

Algorithms	CPU time/s (Recognition rate ± SD%)			
(Methods)	*L=2*	*L=3*	*L=4*	*L=5*
Dimension $d = 64 \times 64$				
EDA	51.014 (76.00 ± 3.41%)	51.074 (81.33 ± 3.36%)	50.931 (85.24 ± 3.09%)	51.141 (85.33 ± 4.44%)
FEDA (Alg. 2)	0.0068 (76.00 ± 3.41%)	0.0078 (81.33 ± 3.36%)	0.0150 (85.24 ± 3.09%)	0.0172 (85.33 ± 4.44%)
Arnoldi-EDA	0.0085 (75.93 ± 3.46%)	0.0339 (81.83 ± 3.11%)	0.0370 (85.90 ± 3.80%)	0.0458 (87.00 ± 4.66%)
Lanczos-EDA	0.0055 (76.44 ± 3.18%)	0.0250 (81.67 ± 3.26%)	0.0320 (84.57 ± 3.86%)	0.0388 (85.78 ± 4.71%)
PCA + LDA LDA(99%)	0.0068 (76.44 ± 3.63%)	0.0128 (82.17 ± 3.07%)	0.0147 (87.14 ± 3.92%)	0.0192 (88.00 ± 4.28%)
PCA + EDA($n-1$)	0.0050 (76.00 ± 3.41%)	0.0128 (81.33 ± 3.36%)	0.0150 (85.24 ± 3.09%)	0.0191 (85.33 ± 4.44%)
EMFA	50.636 (75.93 ± 3.51%)	51.1083 (78.83 ± 2.49%)	50.748 (83.33 ± 3.43%)	50.768 (84.56 ± 5.30%)
FEMFA (Alg. 2)	0.0123 (75.93 ± 3.51%)	0.0238 (78.83 ± 2.49%)	0.0400 (83.33 ± 3.43%)	0.0553 (84.56 ± 5.30%)
PCA + LDA MFA (99%)	0.0090 (76.96 ± 3.34%)	0.0132 (82.25 ± 2.81%)	0.0114 (86.48 ± 3.91%)	0.0194 (88.11 ± 4.77%)
PCA + LDA EMFA ($n-1$)	0.0062 (75.93 ± 3.51%)	0.0102 (78.83 ± 2.49%)	0.0164 (83.24 ± 3.54%)	0.0182 (84.44 ± 5.24%)
ELDE	51.203 (74.96 ± 3.45%)	51.313 (80.50 ± 3.38%)	51.283 (83.43 ± 3.92%)	51.065 (85.22 ± 4.65%)
FELDE (Alg.2)	0.0079 (74.96 ± 3.45%)	0.0125 (80.50 ± 3.38%)	0.0190 (83.43 ± 3.92%)	0.0245 (85.22 ± 4.65%)
PCA + LDA LDE(99%)	0.0065 (77.03 ± 3.29%)	0.0185 (82.42 ± 2.90%)	0.0157 (86.86 ± 4.25%)	0.0185 (88.11 ± 4.77%)
PCA + LDA ELDE($n-1$)	0.0076 (74.96 ± 3.45%)	0.0104 (80.50 ± 3.38%)	0.0120 (83.14 ± 3.68%)	0.0176 (84.44 ± 4.54%)
ELPP	51.226 (74.89 ± 3.78%)	51.287 (80.83 ± 2.36%)	51.175 (85.90 ± 4.21%)	51.152 (87.33 ± 5.55%)
FELPP (Alg. 2)	0.0085 (74.89 ± 3.78%)	0.0106 (80.83 ± 2.36%)	0.0161 (85.90 ± 4.21%)	0.0268 (87.33 ± 5.55%)
PCA + LDA LPP(99%)	0.0075 (75.70 ± 3.66%)	0.0099 (81.67 ± 3.12%)	0.0120 (86.48 ± 3.99%)	0.0157 (88.11 ± 4.69%)
PCA + LDA ELPP($n-1$)	0.0035 (75.26 ± 4.01%)	0.0101 (82.42 ± 2.93%)	0.0140 (86.48 ± 4.66%)	0.0190 (88.11 ± 5.27%)
Dimension $d = 100 \times 100$				
EDA	166.21 (86.67 ± 1.94%)	166.19 (88.33 ± 2.75%)	165.56 (90.76 ± 4.40%)	164.62 (92.44 ± 4.94%)
FEDA (Alg. 2)	0.0170 (86.67 ± 1.94%)	0.0220 (88.33 ± 2.75%)	0.0280 (90.76 ± 4.40%)	0.0370 (92.44 ± 4.94%)
Arnoldi-EDA	0.0820 (86.44 ± 2.21%)	0.0675 (88.25 ± 3.08%)	0.0755 (90.00 ± 5.28%)	0.0809 (91.56 ± 5.11%)
Lanczos-EDA	0.0660 (86.00 ± 2.50%)	0.0641 (88.00 ± 3.36%)	0.0690 (90.29 ± 4.93%)	0.0766 (91.22 ± 6.55%)

Table 1 (continued)

Algorithms	CPU time/s (Recognition rate ± SD%)			
(Methods)	*L=2*	*L=3*	*L=4*	*L=5*
PCA + LDA LDA(99%)	0.0230 (86.89 ± 2.21%)	0.0250 (89.50 ± 4.48%)	0.0250 (91.90 ± 5.88%)	0.0324 (91.89 ± 5.90%)
PCA + EDA($n-1$)	0.0210 (86.81 ± 1.59%)	0.0230 (88.00 ± 2.89%)	0.0285 (90.76 ± 4.51%)	0.0340 (92.33 ± 5.14%)
EMFA	165.33 (86.00 ± 1.96%)	164.34 (88.42 ± 2.02%)	164.36 (90.10 ± 3.81%)	164.01 (91.00 ± 4.24%)
FEMFA (Alg. 2)	0.0310 (86.00 ± 1.96%)	0.0523 (88.42 ± 2.02%)	0.0738 (90.10 ± 3.81%)	0.1081 (91.00 ± 4.24%)
PCA + LDA MFA (99%)	0.0221 (87.04 ± 2.40%)	0.0256 (89.42 ± 4.30%)	0.0263 (92.28 ± 5.62%)	0.0306 (92.22 ± 5.54%)
PCA + LDA EMFA ($n-1$)	0.0183 (86.07 ± 1.81%)	0.0214 (88.42 ± 2.06%)	0.0310 (90.38 ± 3.55%)	0.0360 (91.00 ± 4.24%)
ELDE	167.11 (85.78 ± 1.84%)	168.10 (88.42 ± 2.34%)	167.61 (90.10 ± 4.05%)	167.00 (90.22 ± 4.44%)
FELDE (Alg.2)	0.0204 (85.78 ± 1.84%)	0.0266 (88.42 ± 2.34%)	0.0366 (90.10 ± 4.05%)	0.0480 (90.22 ± 4.44%)
PCA + LDA LDE(99%)	0.0202 (86.96 ± 2.30%)	0.0225 (89.17 ± 4.63%)	0.0244 (92.48 ± 5.84%)	0.0325 (91.89 ± 5.62%)
PCA + LDA ELDE($n-1$)	0.0178 (85.93 ± 1.71%)	0.0253 (88.17 ± 2.57%)	0.0289 (89.71 ± 4.30%)	0.0311 (90.33 ± 4.38%)
ELPP	167.31 (86.37 ± 2.13%)	167.72 (89.92 ± 3.92%)	167.14 (91.81 ± 5.28%)	167.13 (92.56 ± 5.39%)
FELPP (Alg. 2)	0.0265 (86.37 ± 2.13%)	0.0304 (89.92 ± 3.92%)	0.0356 (91.81 ± 5.28%)	0.0440 (92.56 ± 5.39%)
PCA + LDA LPP(99%)	0.0181 (86.59 ± 2.48%)	0.0192 (89.42 ± 4.29%)	0.0255 (91.90 ± 5.70%)	0.0293 (92.44 ± 5.76%)
PCA + LDA ELPP($n-1$)	0.0204 (85.93 ± 1.88%)	0.0229 (89.92 ± 3.80%)	0.0252 (91.43 ± 5.10%)	0.0324 (93.00 ± 4.89%)

Table 2 Example Sect. 5.1 (2): experiments on CMU PIE database, $d = 311040$, $K = 40$, $n = KL$, $r = 39$. Here "O.M." denotes out-of-memory

Algorithms	CPU time/s (recognition rate ± SD%)			
(Methods)	*L=2*	*L=3*	*L=4*	*L=5*
EDA	O.M. (–)	O.M. (–)	O.M. (–)	O.M. (–)
FEDA (Alg. 2)	1.7083 (72.00±21.3%)	2.4792 (79.24±19.9%)	3.7462 (78.75±19.5%)	4.9418 (81.88±20.9%)
Arnoldi-EDA	8.4444 (71.43±22.1%)	11.305 (78.44±20.8%)	13.798 (77.58±20.3%)	16.226 (81.77±20.7%)
Lanczos-EDA	6.7376 (71.81±22.1%)	9.0366 (79.29±20.0%)	11.535 (78.92±19.5%)	13.821 (81.71±20.9%)
PCA + LDA LDA (99%)	1.1239 (71.17±21.6%)	1.5628 (79.51±19.5%)	1.9895 (80.92±17.9%)	2.4454 (82.85±20.0%)
PCA + LDA ELDA($n-1$)	0.9825 (72.00±21.3%)	1.3942 (79.24±19.9%)	1.8260 (78.75±19.5%)	2.3591 (81.88±20.9%)
EMFA	O.M. (–)	O.M. (–)	O.M. (–)	O.M. (–)
FEMFA (Alg. 2)	5.6579 (69.25±21.2%)	11.321 (77.40±20.1%)	19.850 (76.84±20.0%)	30.187 (80.02±20.9%)
PCA + LDA MFA (99%)	0.9422 (70.76±21.7%)	1.2714 (79.09±19.9%)	1.6149 (80.90±17.8%)	1.9988 (82.82±19.9%)
PCA + LDA EMFA ($n-1$)	0.8579 (69.53±20.9%)	1.1771 (77.40±20.2%)	1.5000 (76.84±20.0%)	1.8891 (80.02±21.0%)
ELDE	O.M. (–)	O.M. (–)	O.M. (–)	O.M. (–)
FELDE (Alg.2)	2.3392 (69.88±21.1%)	3.7234 (77.22±19.9%)	5.8922 (77.48±19.1%)	7.8286 (80.45±20.6%)
PCA + LDA LDE(99%)	0.9543 (70.85±21.6%)	1.2769 (78.90±20.0%)	1.5977 (80.83±17.8%)	1.9926 (82.97±19.8%)
PCA + LDA ELDE($n-1$)	0.8631 (69.76±21.3%)	1.1675 (77.22±20.0%)	1.4838 (77.48±19.1%)	1.8892 (80.45±20.6%)
ELPP	O.M. (–)	O.M. (–)	O.M. (–)	O.M. (–)
FELPP (Alg. 2)	2.0077 (71.40±22.0%)	2.6778 (80.50±19.7%)	3.8968 (80.69±18.9%)	5.1021 (82.57±20.7%)
PCA + LDA LPP(99%)	0.9637 (70.56±21.8%)	1.2634 (79.78±19.5%)	1.5946 (81.06±18.6%)	1.9732 (82.97±20.5%)
PCA + LDA ELPP($n-1$)	0.8700 (70.99±22.1%)	1.1676 (80.06±20.0%)	1.4874 (80.30±19.1%)	1.9040 (82.33±20.8%)

We consider the CMU PIE (Pose, Illumination, Expression) database. It is taken from CMU 3D Room, which includes over 40000 facial images of 68 individuals. For each individual, it has 13 different poses, under 43 different illumination and 4 different expressions. We choose total 800 images of 40 people as the subset, and for each individual we select 20 images with different illuminations, poses and expressions. The size of images is $d = 486 \times 640$ pixels. We randomly choose $L = 2,\ 3,\ 4,\ 5$ images from each class as the training set, and use the rest as the testing set.

In the CMU PIE database, the data dimension $d = 311040$, the number of classes $K = 40$, and we choose the reducing dimension $r = K - 1 = 39$. We run the eighteen algorithms on this problem, and the numerical results are listed in Table 2. It is seen that EDA, EMFA, ELDE and ELPP do not work at all for this problem, because they have to deal with matrix exponential problems of size 311040×311040. Indeed, all of them suffer from the difficulty of *out-of-memory* (abbreviated as "O.M." in Table 2). As a comparison, the four fast algorithms and two inexact Krylov-type algorithms perform quite well, and FEDA is about four times faster than Arnoldi-EDA and Lanczos-EDA. So Algorithm 2 can deal with data with very high dimensionality.

For the CMU PIE database, we see that the recognition rates from all the algorithms are comparable. On the other hand, Algorithm 2 is slower than the other two PCA-processing algorithms. Indeed, the PCA-preprocessing algorithms construct the adjacency matrices by using the "reduced" data matrices whose size is much smaller than the "original" data matrices. Furthermore, FEMFA (Alg. 2) is slower than the other three fast exponential-based algorithms for this problem. This is due to the fact that the overhead for constructing scatter matrices in the MFA-type algorithm is a little higher than that for the other three algorithms.

(3) *We demonstrate that our fast algorithms can be more powerful than original PCA plus graph embedding methods and other popular LDA-based methods for dimensionality reduction.*

We consider the AR database, which contains 1680 face images of 120 individuals, with 14 images per people. In the AR database, all images are cropped and scaled to 50×40. We randomly select $L = 2,\ 3,\ 4,\ 5$ images from each class as the training set, and the rest are used as the testing set. The data dimension $d = 2000$, the number of classes $K = 120$, and we choose the reducing dimension $r = 50$.

To illustrate the competitiveness of Algorithm 2, we compare it with some PCA plus graph embedding methods. PCA preprocessed algorithms are popular for dimensionality reduction [2, 33]. The algorithms for comparison include PCA + LDA LDA, PCA + LDA MFA, PCA + LDA LDE, and PCA + LDA LPP, in which we preserve 99% energy in the PCA stage, followed by the corresponding algorithms. Also we run some popular LDA-based methods for dimensionality reduction including the regularized LDA (RLDA) (the regularized parameter is chosen as 0.001) [11], and the null space LDA (NLDA) [5], QRLDA [44], GSVDLDA [18]. The numerical results are given in Table 3.

It is seen from Table 3 that the recognition rates obtained from the fast algorithms based on Algorithm 2 (FEDA, FEMFA, FELPP, FELDE) are (much) higher than the original PCA pre-processed methods (PCA + LDA LDA, PCA + LDA MFA,

Table 3 Example Sect. 5.1 (3): experiments on AR database, $d = 2000$, $K = 120$, $n = KL$, $r = 50$

Algorithms	CPU time/s (recognition rate ± SD%)			
(Methods)	*L=2*	*L=3*	*L=4*	*L=5*
EDA	2.1559 (83.29±10.1%)	2.1495 (88.78±8.63%)	2.2477 (93.33±8.17%)	2.2029 (93.30±9.13%)
FEDA (Alg. 2)	0.0597 (83.29±10.1%)	0.1070 (88.78±8.63%)	0.1772 (93.33±8.17%)	0.2582 (93.30±9.13%)
PCA + LDA LDA(99%)	0.0670 (79.57±9.06%)	0.1139 (82.56±9.33%)	0.1061 (88.13±9.33%)	0.1451 (89.90±10.8%)
RLDA	0.9531 (83.95±10.0%)	0.9251 (88.98±8.53%)	0.9362 (92.57±8.19%)	0.9719 (92.41±9.73%)
NLDA	0.1856 (83.79±10.1%)	0.2449 (88.81±8.68%)	0.2982 (92.22±8.24%)	0.3979 (92.19±9.76%)
Arnoldi-EDA	0.1655 (83.72±10.3%)	0.2107 (89.11±8.60%)	0.2621 (93.39±8.18%)	0.3510 (93.56±9.20%)
Lanczos-EDA	0.1419 (84.22±10.1%)	0.1967 (89.99±8.09%)	0.2355 (93.76±7.65%)	0.3292 (94.02±8.87%)
QRLDA	0.0452 (81.06±9.61%)	0.0486 (88.48±8.72%)	0.0563 (93.31±8.12%)	0.0733 (93.19±9.47%)
GSVDLDA	0.1543 (74.98±9.82%)	0.2433 (79.80±9.40%)	0.3290 (85.24±8.70%)	0.4510 (84.65±12.0%)
EMFA	2.5767 (85.19±9.35%)	2.9845 (90.87±7.43%)	3.7299 (94.68±6.90%)	4.5306 (94.78±7.97%)
FEMFA (Alg. 2)	0.4654 (85.19±9.35%)	0.9655 (90.87±7.43%)	1.6372 (94.68±6.90%)	2.4937 (94.78±7.97%)
PCA + LDA MFA (99%)	0.1608 (80.37±9.33%)	0.3276 (86.27±8.11%)	0.5008 (90.96±7.93%)	0.7721 (91.71±9.84%)
ELDE	2.1854 (85.08±9.42%)	2.2059 (90.71±7.54%)	2.2292 (94.46±6.93%)	2.3713 (94.52±7.99%)
FELDE (Alg.2)	0.0796 (85.08±9.42%)	0.1405 (90.71±7.54%)	0.2637 (94.46±6.93%)	0.3883 (94.52±7.99%)
PCA + LDA LDE(99%)	0.0825 (80.22±9.10%)	0.1638 (86.52±8.17%)	0.1465 (91.36±7.78%)	0.2229 (92.15±9.52%)
ELPP	2.1685 (83.82±11.1%)	2.1496 (89.51±8.42%)	2.1872 (93.06±8.21%)	2.2762 (92.57±9.81%)
FELPP (Alg. 2)	0.0703 (83.82±11.1%)	0.1231 (89.51±8.42%)	0.2138 (93.06±8.21%)	0.3025 (92.57±9.81%)
PCA + LDA LPP(99%)	0.0374 (73.76±8.75%)	0.0640 (81.43±8.97%)	0.1139 (88.18±9.29%)	0.1607 (89.85±11.0%)

PCA + LDA LPP, PCA + LDA LDE), especially when L is small. Meanwhile, the standard deviations of the FEDA, FEMFA, FELPP, FELDE (Alg.2) are often lower than those of other original PCA pre-processed methods (PCA + LDA LDA, PCA + LDA MFA, PCA + LDA LPP, PCA + LDA LDE). Again, all the fast algorithms based on Algorithm 2 run much faster than the original matrix exponential-based algorithms. We see that the CPU time used in Algorithm 2 is comparable to the PCA pre-processed algorithm, and at the same time, Algorithm 2 shares the same recognition rates and standard derivations as the original matrix exponential-based algorithms. In conclusion, our new algorithms possess the advantages of both the original matrix exponential-based algorithms and the PCA pre-processed algorithms, and meanwhile diminish the disadvantages of those two algorithms.

In addition, we find that the recognition rates of FEDA (Alg. 2) are higher than those of PCA + LDA LDA and GSVDLDA. Though the recognition rates and standard deviations obtained form FEDA (Alg. 2), EDA, RLDA, NLDA, Arnoldi-EDA, Lanczos-EDA and QRLDA are similar, QRLDA runs faster than FEDA (Alg. 2), and FEDA(Alg.2) run much faster than other algorithms. This demonstrates that our fast algorithms FEDA (Alg. 2) is more powerful than these LDA-based methods, expect for QRLDA. Although our algorithm FEDA (Alg. 2) is weaker than QRLDA in terms of CPU time, in next subsection, we will show that FEDA (Alg. 2) is more stable than QRLDA.

5.2 Stability of the proposed algorithm

In this subsection, we aim to illustrate the stability of Algorithm 2, and show the effectiveness of the theoretical analysis given in Sect. 4. The test set is the FERET database, which consists of 14051 eight-bit grayscale images of human faces with views ranging from frontal to left and right profiles.

To show the stability of Algorithm 2, we perturb the training set X as $\underline{X} = X + E$, where E is a Gaussian matrix generated by using the MATLAB command `normrnd(0, l ,[d,n])`, i.e., a d-by-n zero-mean Gaussian distributed matrix with variance l:

$$\mathtt{E} = \mathtt{normrnd}(0, \mathtt{l}, [\mathtt{d}, \mathtt{n}]); \quad \mathtt{E} = \varepsilon * \mathtt{norm}(\mathtt{X}, '\mathtt{fro}') * \mathtt{E}/\mathtt{norm}(\mathtt{E}, '\mathtt{fro}').$$

Here ε is a user-described parameter, which is chosen as $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-4} in the experiment, moreover, $\|E\|_F / \|X\|_F = \varepsilon$.

We first run the four fast algorithms FEDA (Alg. 2), FEMFA (Alg. 2), FELDE(Alg.2) and FELPP (Alg. 2) on this problem. We compare them with the four corresponding PCA plus algorithms PCA + LDA LDA, PCA + LDA MFA, PCA + LDA LDE and PCA + LDA LPP, where 99% energy is preserved in the PCA stage. Table 4 lists the recognition rates of the algorithms with and without (i.e., $\varepsilon = 0$, $l = 0$) perturbation. All the experiments are repeated for 10 times, and the numerical results are the mean from the 10 runs.

Again, we see from Table 4 that the recognition rates of the four fast exponential-based algorithms are much higher than those from the four PCA pre-processed algorithms. This again demonstrates the efficiency of the exponential-based methods

Table 4 Example Sect. 5.2: recognition rates of the algorithms with and without perturbation

Variance	Algorithms	Recognition rates				Original
l	(Methods)	$\varepsilon = 10^{-1}$ (%)	$\varepsilon = 10^{-2}$ (%)	$\varepsilon = 10^{-3}$ (%)	$\varepsilon = 10^{-4}$ (%)	ε, $l = 0$ (%)
	FEDA (Alg. 2)	60.90	62.20	62.16	62.18	62.18
	PCA + LDA LDA(99%)	23.43	32.58	32.35	32.43	35.59
	FEMFA (Alg. 2)	60.14	60.36	60.51	60.55	60.54
$l = 0.1$	PCA + LDA MFA (99%)	22.79	37.85	37.34	37.21	37.15
	FELDE (Alg.2)	63.06	63.70	63.66	63.54	63.54
	PCA + LDA LDE(99%)	23.51	37.75	37.32	37.26	37.37
	FELPP (Alg. 2)	59.69	58.76	58.60	58.61	58.61
	PCA + LDA LPP(99%)	11.31	22.46	22.73	22.68	22.59
	FEDA (Alg. 2)	61.01	62.23	62.18	62.16	62.18
	PCA + LDA LDA(99%)	24.64	33.88	32.05	32.75	35.59
	FEMFA (Alg. 2)	60.00	60.38	60.45	60.55	60.54
$l = 0.01$	PCA + LDA MFA (99%)	22.96	36.89	37.06	37.29	37.15
	FELDE (Alg.2)	62.68	63.35	63.61	63.54	63.54
	PCA + LDA LDE(99%)	24.19	37.07	37.21	37.39	37.37
	FELPP (Alg. 2)	59.62	58.79	58.64	58.61	58.61
	PCA + LDA LPP(99%)	11.01	23.04	22.91	22.84	22.59
	FEDA (Alg. 2)	61.10	62.36	62.16	62.18	62.18
	PCA + LDA LDA(99%)	24.18	33.04	32.65	32.59	35.59
	FEMFA (Alg. 2)	59.91	60.31	60.54	60.55	60.54
$l = 0.001$	PCA + LDA MFA (99%)	21.71	37.14	37.37	37.16	37.15
	FELDE (Alg.2)	62.81	63.73	63.65	63.54	63.54
	PCA + LDA LDE(99%)	22.46	36.89	37.20	37.20	37.37
	FELPP (Alg. 2)	59.58	58.75	58.64	58.61	58.61
	PCA + LDA LPP(99%)	10.65	22.88	22.76	22.95	22.59

The FERET database, $d = 6400$, $K = 200$, $L = 3$, $n = KL$, $r = 50$, $\sigma_{\min}(X) \approx 0.0358$

for face recognition. In terms of recognition accuracy, we see that the four fast exponential-based algorithms are much more stable than the four PCA pre-processed algorithms.

In order to show this more precisely, we define "variation" of recognition rates as follows:

Table 5 Example Sect. 5.2: variation of recognition rates

Variance	Algorithms	Variation of recognition rates			
l	(Methods)	$\varepsilon = 10^{-1}$ (%)	$\varepsilon = 10^{-2}$ (%)	$\varepsilon = 10^{-3}$ (%)	$\varepsilon = 10^{-4}$ (%)
$l = 0.1$	FEDA (Alg. 2)	2.05	0.04	0.02	0.00
	PCA + LDA LDA(99%)	34.18	8.47	9.10	8.89
	FEMFA (Alg. 2)	0.66	0.29	0.04	0.02
	PCA + LDA MFA (99%)	38.66	1.88	0.50	0.17
	FELDE (Alg.2)	0.75	0.26	0.20	0.00
	PCA + LDA LDE(99%)	37.09	1.00	0.13	0.30
	FELPP (Alg. 2)	1.83	0.26	0.02	0.00
	PCA + LDA LPP(99%)	49.92	0.55	0.61	0.39
$l = 0.01$	FEDA (Alg. 2)	1.87	0.08	0.00	0.02
	PCA + LDA LDA(99%)	30.77	4.81	9.94	7.97
	FEMFA (Alg. 2)	0.89	0.27	0.14	0.02
	PCA + LDA MFA (99%)	38.19	0.71	0.24	0.37
	FELDE (Alg.2)	1.36	0.30	0.12	0.00
	PCA + LDA LDE(99%)	35.28	0.80	0.43	0.03
	FELPP (Alg. 2)	1.73	0.30	0.04	0.00
	PCA + LDA LPP(99%)	51.25	1.99	1.44	1.11
$l = 0.001$	FEDA (Alg. 2)	1.73	0.30	0.02	0.00
	PCA + LDA LDA(99%)	32.07	7.17	8.25	8.43
	FEMFA (Alg. 2)	1.03	0.37	0.00	0.02
	PCA + LDA MFA (99%)	41.55	0.03	0.61	0.03
	FELDE (Alg.2)	1.14	0.30	0.18	0.00
	PCA + LDA LDE(99%)	39.90	1.30	0.47	0.47
	FELPP (Alg. 2)	1.64	0.23	0.04	0.00
	PCA + LDA LPP(99%)	52.85	1.27	0.77	1.60

The FERET database, $d = 6400$, $K = 200$, $L = 3$, $n = KL$, $r = 50$, $\sigma_{\min}(X) \approx 0.0358$

$$\texttt{Variation} = \frac{|\texttt{Perturbed Recognition Rate} - \texttt{Original Recognition Rate}|}{\texttt{Original Recognition Rate}},$$

where "Perturbed recognition rate" and "Original recognition rate" stand for the recognition rates of algorithms "with" and "without" perturbation, respectively. Figure 1 (the upper figure) plots the figure of variation for the eight algorithms when $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-4} with $l = 0.01$. It is obvious to see from Table 5 and Fig. 1 (the upper figure) that the exponential-based algorithms are much more stable than the PCA plus algorithms, because the variation values of the former are much smaller than those of the latter.

To interpret this more precisely, we plot all the singular values of the training set X in Fig. 2, and the smallest nonzero singular value is about 0.0358, where those close to zero are caused by rounding off errors. By Theorem 4.3, the distance between V and $\underline{V}$ are bounded by $\mathcal{O}\big(\varepsilon/\sigma_{\min}(X)\big)$, which will be much less than 1 as ε

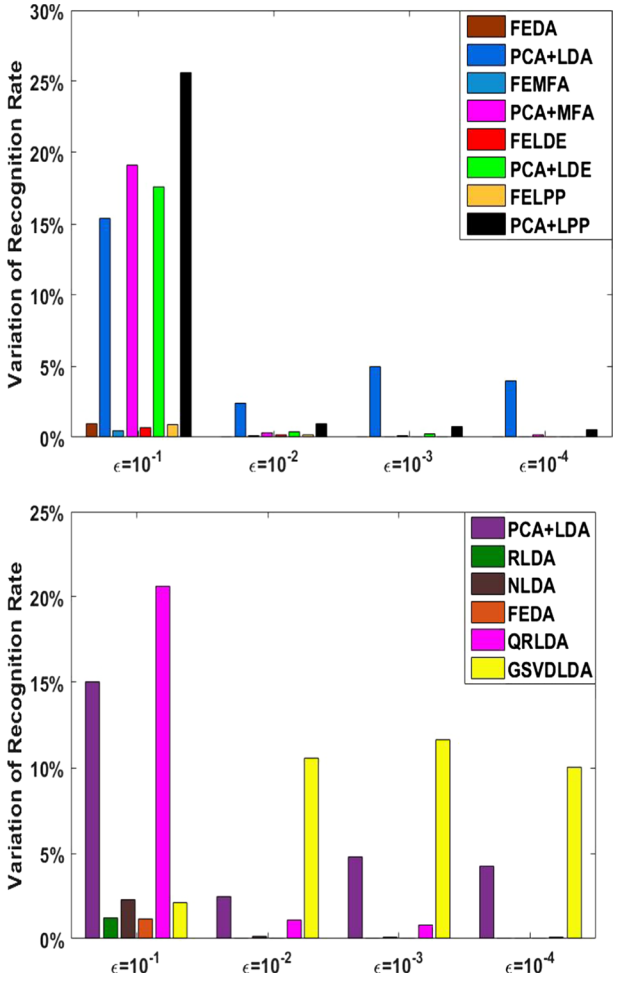

Fig. 1 Variation of the recognition rates on the FERET database, $d = 6400$, $K = 200$, $L = 3$, and $n = KL$, $l = 0.01$. The upper is for comparing our fast algorithms (FEDA, FEMFA, FELPP, FELDE) with the four corresponding PCA plus algorithms (PCA + LDA LDA, PCA + LDA MFA, PCA + LDA LPP, PCA + LDA LDE). The lower is for comparing the six LDA-based methods including PCA + LDA LDA, FEDA, RLDA, NLDA, QRLDA and GSVDLDA

is small enough, say, $\varepsilon = 10^{-4}$. In terms of Theorem 4.1, the recognition rates from the perturbed data are about the same as those from the unperturbed one. This shows the effectiveness of our theoretical results. Interestingly, we see that the four fast exponential-based algorithms are still very robust even when ε is much larger than the smallest singular value, say, $\varepsilon = 10^{-1}$.

Next, we compare FEDA (Alg. 2) with some popular discriminant analysis methods including NLDA [5], QRLDA [44], GSVDLDA [18], and RLDA [11], where the regularized parameter in RLDA is set to be 0.001. The recognition rates with and without perturbation are listed on Table 6, and the variation of the recognition rates

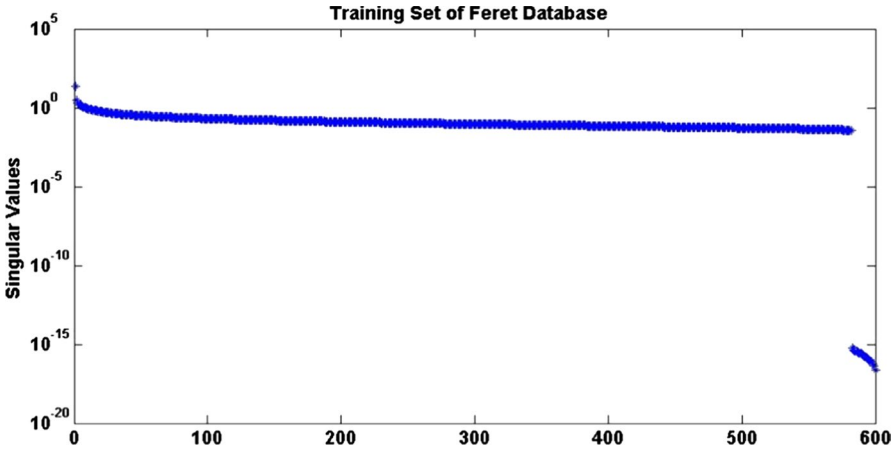

Fig. 2 Singular values of the training set X, the FERET database, $d = 6400$, $K = 200$, $L = 3$, $n = KL$

Table 6 Example Sect. 5.2: recognition rates of the algorithms with and without perturbation

Variance	Algorithms	Recognition rates				Original
l	(Methods)	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon,\ l = 0$
$l = 0.1$	PCA + LDA LDA(99%)	25.54%	33.84%	33.04%	32.28%	35.59%
	RLDA	61.54%	59.73%	59.70%	59.71%	59.71%
	NLDA	61.13%	58.23%	58.16%	58.06%	58.09%
	FEDA (Alg. 2)	61.12%	62.11%	62.18%	62.18%	62.18%
	QRLDA	23.35%	39.43%	39.92%	39.35%	40.10%
	GSVDLDA	10.76%	9.70%	9.55%	9.75%	11.88%
$l = 0.01$	PCA + LDA LDA(99%)	24.89%	33.81%	32.18%	32.56%	35.59%
	RLDA	61.16%	59.73%	59.73%	59.71%	59.71%
	NLDA	60.78%	58.25%	58.00%	58.11%	58.09%
	FEDA (Alg. 2)	60.71%	62.17%	62.18%	62.18%	62.18%
	QRLDA	23.56%	39.23%	39.46%	39.24%	40.10%
	GSVDLDA	11.38%	9.36%	9.11%	9.49%	11.88%
$l = 0.001$	PCA + LDA LDA(99%)	24.41%	33.74%	31.91%	32.05%	35.59%
	RLDA	61.26%	59.60%	59.67%	59.71%	59.71%
	NLDA	60.79%	58.06%	58.06%	58.11%	58.09%
	FEDA (Alg. 2)	60.93%	62.13%	62.20%	62.16%	62.18%
	QRLDA	23.28%	39.36%	39.11%	39.34%	40.10%
	GSVDLDA	11.77%	10.18%	9.31%	9.83%	11.88%

The FERET database, $d = 6400$, $K = 200$, $L = 3$, $n = KL$, $r = 50$

is given in Table 7. We see from Table 6 that the recognition rates of FEDA (Alg. 2) are higher than those of the others in most cases, which demonstrates the superiority of the exponential discriminant methods for recognition. Furthermore, it is observed from Table 7 and Fig. 1 (the lower figure) that FEDA (Alg. 2) is the most stable one

Table 7 Example Sect. 5.2: variation of recognition rates

Variance	Algorithms	Variation of recognition rates			
l	(Methods)	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$
$l = 0.1$	PCA + LDA LDA(99%)	28.24%	4.92%	7.17%	9.31%
	RLDA	3.06%	0.02%	0.02%	0.00%
	NLDA	5.23%	0.24%	0.13%	0.04%
	FEDA (Alg. 2)	1.69%	0.10%	0.00%	0.00%
	QRLDA	41.77%	1.68%	0.44%	1.87%
	GSVDLDA	9.37%	18.32%	19.58%	17.89%
$l = 0.01$	PCA + LDA LDA(99%)	30.07%	4.99%	9.59%	8.50%
	RLDA	2.43%	0.02%	0.02%	0.00%
	NLDA	4.63%	0.28%	0.15%	0.04%
	FEDA (Alg. 2)	2.35%	0.00%	0.00%	0.00%
	QRLDA	41.24%	2.18%	1.59%	2.15%
	GSVDLDA	4.21%	21.16%	23.26%	20.11%
$l = 0.001$	PCA + LDA LDA(99%)	31.40%	5.20%	10.33%	9.94%
	RLDA	2.60%	0.19%	0.06%	0.00%
	NLDA	4.65%	0.04%	0.04%	0.04%
	FEDA (Alg. 2)	2.01%	0.08%	0.04%	0.02%
	QRLDA	41.96%	1.84%	2.46%	1.90%
	GSVDLDA	0.84%	14.32%	21.58%	17.26%

The FERET database, $d = 6400$, $K = 200$, $L = 3$, $n = KL$, $r = 50$

compared with others. Thus, Algorithm 2 is both fast and stable, and it is a competitive candidate for high dimensionality reduction and recognition.

6 Conclusion

Exponential discriminant analysis methods can be utilized to settle the small-sample-size problem arising in dimensionality reduction, and they often have more discriminant power than their original counterparts. However, one has to solve large-scale matrix exponential eigenproblems which are the bottleneck in this type of methods.

The first contribution of this paper is to propose a fast implementation framework on exponential discriminant analysis methods. To this aim, we first reformulate large-scale matrix exponential of size d to a concise form, and then reduce the large-scale exponential eigenproblem to a small-sized one of size n, where d is dimension of the data and n is the number of samples. Our new algorithm runs much faster than its original counterpart, with no recognition rate lost.

The second contribution of this paper is to show the stability of the exponential discriminant analysis methods from a matrix perturbation point of view. The key result indicates that, unlike conventional discriminant analysis methods, the exponential discriminant methods refrain from the difficulty of (possible) small Crawford

number; see (4.17). Numerical experiments illustrate the numerical behavior of the new algorithm and demonstrate the effectiveness of the theoretical results. Furthermore, we point out that our framework can be generalized to all exponential-based matrix transformation methods, and it is favorable for data with high dimension and small number of training samples.

Acknowledgements We would like to express our sincere thanks to the anonymous referees and our editor for insightful comments and suggestions that greatly improved the representation of this paper.

References

1. Ahmed, N.: Exponential discriminant regularization using nonnegative constraint and image descriptor. In: IEEE 9th international conference on emerging technologies, pp. 1–6 (2013)
2. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs Fisherface: recognition using class-specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 711–720 (1997)
3. Chang, X., Paige, C.C., Stewart, G.W.: Preturbation analysis for the QR factorization. SIAM J. Matrix Anal. Appl. **18**, 775–791 (1997)
4. Chen, H., Chang, H., Liu, T.: Local discriminant embedding and its variants. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 846–853 (2005)
5. Chen, L., Liao, H., Ko, M., Lin, J., Yu, G.: A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognit. **33**, 1713–1726 (2000)
6. Chu, D., Thye, G.: A new and fast implementation for null space based linear discriminant analysis. Pattern Recognit. **43**, 1373–1379 (2010)
7. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**, 21–27 (1967)
8. Dornaika, F., Bosaghzadeh, A.: Exponential local discriminant embedding and its application to face recognition. IEEE Trans. Syst. Man Cybern. Part B Cybern. **43**, 921–934 (2013)
9. Dornaika, F., El Traboulsi, Y.: Matrix exponential based semi-supervised discriminant embedding for image classification. Pattern Recognit. **61**, 92–103 (2017)
10. Duda, R., Hart, P., Stock, D.: Pattern Classification, 2nd edn. Wiley, New York (2004)
11. Friedman, J.: Regularized discriminant analysis. J. Am. Stat. Assoc. **84**, 165–175 (1989)
12. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Elsevier Academic Press, London (1999)
13. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. The Johns Hopkins university Press, Baltimore (2013)
14. Higham, N.J.: Functions of Matrices: Theory and Computation. SIAM, Philadelphia (2008)
15. He, J., Ding, L., Cui, M., Hu, Q.: Marginal Fisher analysis based on matrix exponential transformation. Chin. J. Comput. **10**, 2196–2205 (2014). (in Chinese)
16. He, X., Niyogi, P.: Locality preserving projections. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems, vol. 16, pp. 153–160. MIT Press, Cambridge (2004)
17. He, X., Yan, S.C., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 328–340 (2005)
18. Howland, P., Park, H.: Generalizing discriminant analysis using the generalized singular value decomposition. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 995–1006 (2004)
19. Jolliffe, I.: Principal Component Analysis. Springer, New York (1986)
20. Kokiopoulou, E., Chen, J., Saad, Y.: Trace optimization and eigenproblems in dimension reduction methods. Numer. Linear Algebra Appl. **18**, 565–602 (2011)
21. Kokiopoulou, E., Saad, Y.: Orthogonal neighborhood preserving projections. In: Houston, T.X., Han, J., et al. (eds.) IEEE 5th International Conference on Data Mining (ICDM05), pp. 234–241. IEEE, New York (2005)
22. Kokiopoulou, E., Saad, Y.: Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 2143–2156 (2007)
23. Kramer, R., Young, A., Burton, A.: Understanding face familiarity. Cognition **172**, 46–58 (2018)
24. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. The MIT Press, Cambridge (2012)

25. Park, C., Park, H.: A comparision of generalized linear discriminant analysis algorithms. Pattern Recognit. **41**, 1083–1097 (2008)
26. Qiao, L., Chen, S., Tan, X.: Sparsity preserving projections with applications to face recognition. Pattern Recognit. **43**, 331–341 (2010)
27. Seghouane, A., Shokouhi, N., Koch, I.: Sparse principal component analysis with preserved sparsity pattern. IEEE Trans. Image Process. **28**, 3274–3285 (2019)
28. Shi, W., Wu, G.: Why PCA plus graph embedding methods can be unstable for extracting classification features? (submitted for publication)
29. Stewart, G.W.: Perturbation bounds for the definite generalized eigenvalue problem. Linear Algebra Appl. **23**, 69–85 (1979)
30. Sun, J.: The perturbation bounds for eigenspaces of a definite matrix-pair. Numer. Math. **41**, 321–343 (1983)
31. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
32. Torgerson, W.: Multidimensional scaling: I. Theory and method. Psychometrika **17**, 401–419 (1952)
33. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cognit. Neurosci. **3**, 71–86 (1991)
34. Van Loan, C.F.: The sensitivity of the matrix exponential. SIAM J. Numer. Anal. **14**, 971–981 (1977)
35. Wang, J.: Geometric Structure of High-Dimensional Data and Dimensionality Reduction. Higher Education Press, Beijing (2011)
36. Wang, S., Chen, H., Peng, X., Zhou, C.: Exponential locality preserving projections for small sample size problem. Neurocomputing **74**, 36–54 (2011)
37. Wang, S., Yan, S., Yang, J., et al.: A general exponential framework for dimensionality reduction. IEEE Trans. Image Process. **23**, 920–930 (2014)
38. Webb, A.: Statistical Pattern Recognition, 2nd edn. Wiley, Hoboken (2002)
39. Wu, G., Feng, T., Zhang, L., Yang, M.: Inexact implementation using Krylov subspace methods for large scale exponential discriminant analysis with applications to high dimensionality reduction problems. Pattern Recogniti. **66**, 328–341 (2017)
40. Wu, G., Xu, W., Leng, H.: Inexact and incremental bilinear Lanczos components algorithms for high dimensionality reduction and image reconstruction. Pattern Recognit. **48**, 244–263 (2015)
41. Yan, L., Pan, J.: Two-dimensional exponential discriminant analysis and its application to face recognition. In: International Conference on Computational Aspects of Social Networks (CASoN), pp. 528–531 (2010)
42. Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 40–51 (2007)
43. Yang, J., Zhang, D., Yang, J., Niu, B.: Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 650–664 (2007)
44. Ye, J., Li, Q.: LDA/QR: an efficient and effective dimension reduction algorithm and its theoretical foundation. Pattern Recognit. **37**, 851–854 (2004)
45. Yuan, S., Mao, X.: Exponential elastic preserving projections for facial expression recognition. Neurocomputing **275**, 711–724 (2018)
46. Zhang, T., Fang, B., Tang, Y., Shang, Z., Xu, B.: Generalized discriminant analysis: a matrix exponential approach. IEEE Trans. Syst. Man Cybern. Part B Cybern. **40**, 186–197 (2010)

Affiliations

Wenya Shi[1] · Youwei Luo[1] · Gang Wu[1]

Wenya Shi
shiwenyaer@163.com

Youwei Luo
wpdatamining@163.com

[1] School of Mathematics, China University of Mining and Technology, Xuzhou 221116, Jiangsu, People's Republic of China