IEICE TRANS. INF. & SYST., VOL.E101–D, NO.5 MAY 2018

PAPER

Exponential Neighborhood Preserving Embedding for Face Recognition

Ruisheng RAN[†,††a)], **Bin FANG**[†b)], ***Nonmembers, and*** **Xuegang WU**[†††c)], ***Member***

SUMMARY Neighborhood preserving embedding is a widely used manifold reduced dimensionality technique. But NPE has to encounter two problems. One problem is that it suffers from the small-sample-size (SSS) problem. Another is that the performance of NPE is seriously sensitive to the neighborhood size k. To overcome the two problems, an exponential neighborhood preserving embedding (ENPE) is proposed in this paper. The main idea of ENPE is that the matrix exponential is introduced to NPE, then the SSS problem is avoided and low sensitivity to the neighborhood size k is gotten. The experiments are conducted on ORL, Georgia Tech and AR face database. The results show that, ENPE shows advantageous performance over other unsupervised methods, such as PCA, LPP, ELPP and NPE. Another is that ENPE is much less sensitive to the neighborhood parameter k contrasted with the unsupervised manifold learning methods LPP, ELPP and NPE.

key words: *neighborhood preserving embedding, matrix exponential, face recognition, the small-sample-size problem, manifold learning*

1. Introduction

The classical principal component analysis (PCA) [1] and linear discriminant analysis (LDA) [2] are two important and effective approaches in pattern recognition. However, PCA and LDA aim only to preserve the global structures of the image samples and cannot uncover the essential manifold structure of the image.

Recently, many of manifold learning algorithms have been developed, such as locally linear embedding (LLE) [3], Isomap [4], Laplacian eigenmaps (LE) [5], local tangent space alignment (LTSA) [6]. These manifold learning algorithms discover the intrinsic geometry structure of a data set and have been widely used in the past decade. Unfortunately, all of these algorithms suffer from the out-of-sample problem [7]. To address this problem, a linearization procedure is developed, which constructs a linear map from the original data space to new low-dimensionality space. Representative ones are local preserving projection (LPP) [8] and neighborhood preserving embedding (NPE) [9]. LPP is a linearization version of LE [5] and NPE is a linearization version of LLE [3]. In addition, stochastic neighbor embedding (SNE) [10] and its variation, *t*-distributed stochastic neighbor embedding (t-SNE) [11], are also popular embedding methods.

The main idea of NPE is that it introduces a linear transformation matrix into LLE. NPE inherits LLE's neighborhood preserving property. Due to its simplicity and effectiveness, NPE has become a popular method in computer vision field. Recently, NPE has been investigated in many literatures, such as [12]–[14]. However, NPE has to encounter two problems. One problem of NPE is the fact that, like LDA, it also suffers from the small-sample-size (SSS) problem. Because that, in most cases, the dimension of the sample is much larger than the number of the samples, the generalized eigenvalue problem may be unsolvable. Another problem of NPE is that, the neighbor relationship is measured by the artificially constructed adjacent graph. Usually, the k nearest neighbor or ε-neighborhood criteria is used to construct adjacent graph. The performance of NPE is seriously sensitive to the neighborhood size k, which is observed in the experiments presented in the Sect. 5 of this paper.

As an effective method, exponential discriminant analysis (EDA) [15] is proposed to overcome the SSS problem of classical LDA. The main idea of EDA is that the matrix exponential is introduced. It replaces the between-class scatter matrix $\boldsymbol{S}_B$ with the matrix exponential $\exp(\boldsymbol{S}_B)$, and replaces the within-class scatter matrix $\boldsymbol{S}_W$ with the matrix exponential $\exp(\boldsymbol{S}_W)$, and so avoid the singularity of the matrix $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$.

After the release of EDA, it is also introduced to solve the SSS problem, especially in the manifold learning field. Many of manifold learning algorithms, such as Local Preserving Projection (LPP) [8], Discriminant Locality Preserving Projection (DLPP) [16], Local Discriminant Embedding (LDE) [17] and Semi-supervised Discriminant Embedding (SDE) [18], have to suffer from the SSS problem. Then, the EDA method is introduced to solve this problem. The exponential LPP (ELPP) [19], the exponential DLPP method (EDLPP) [20], the exponential LDE method (ELDE) [21] and the exponential SDE method (ESDE) [22] are proposed. They are the exponential versions of the corresponding methods. They avoid the SSS problem and show better performance in face recognition.

Inspired by the idea of EDA, an exponential neighborhood preserving embedding (ENPE) is proposed in this paper. The main advantages of the proposed ENPE are two

Manuscript received August 13, 2017.
Manuscript revised December 27, 2017.
Manuscript publicized January 23, 2018.

[†]The authors are with Chongqing University, Chongqing, 400044, China.
[††]The author is with Chongqing Normal University, Chongqing, 401331, China.
[†††]The author is with Yangtse Normal University, Chongqing, 408100, China.
a) E-mail: rshran@163.com
b) E-mail: fb@cqu.edu.cn (Corresponding author)
c) E-mail: xgwu@cqu.edu.cn

DOI: 10.1587/transinf.2017EDP7259

aspects. One is that ENPE avoids the SSS problem of NPE and shows advantageous performance over NPE in face recognition. Another is that, ENPE is much less sensitive to the neighborhood parameter k, and it can get stable recognition performance when the parameter k varies.

NPE and the proposed ENPE are unsupervised methods. In general, unsupervised and supervised method are two different methods. In face recognition, the performance of the two methods are different because that the class-specific information is used or not. This paper focus on the unsupervised technique. Unsupervised learning is a vast field. It is usually applied in such case where class label of data or other guidance for training is not available. Compared with supervised learning, unsupervised learning is generally much difficult due to the lack of label information [23], [24]. In manifold learning filed, the original manifold learning algorithms are generally unsupervised, and then these algorithms are extended to supervised cases.

The rest of this paper is organized as follows. In Sect. 2, the neighborhood preserving embedding method is reviewed. In Sect. 3, exponential neighborhood preserving embedding (ENPE) is presented. Section 4 provides the theoretical analysis of the proposed ENPE method. Experimental results are shown in Sect. 5. Finally, Sect. 6 concludes this study.

2. Review of NPE

Neighborhood preserving embedding (NPE) [9] is unsupervised manifold reduced dimension technique proposed in recent years. NPE embeds the original data to a low dimensional space, in which the local neighborhood structure on the data manifold is preserved.

Let $\boldsymbol{X} = \{\boldsymbol{x}_i \in R^D \,|\, i = 1, 2, \cdots, N\}$ represents the input data in R^D space. NPE aims to seek an optimal transformation matrix $\boldsymbol{A}$ to map the D-dimensional data point $\boldsymbol{x}_i$ onto a d-dimensional data point $\boldsymbol{y}_i$, $\{\boldsymbol{y}_i \in R^d \,|\, i = 1, 2, \cdots, N\}$ ($d \ll D$), namely, $\boldsymbol{y}_i = \boldsymbol{A}^T\boldsymbol{x}_i$, in which the local neighborhood structure of the original data set $\boldsymbol{X}$ can be preserved.

NPE first finds the neighbors of each data point in space R^D, then constructs an adjacency graph on the input data set.

Let the weights $\boldsymbol{W}_{ij}$ be the coefficients that best reconstruct $\boldsymbol{x}_i$ from its neighbors $j = 1, 2, \cdots, k$, and $\boldsymbol{W} = (\boldsymbol{W}_{ij})$ be the reconstruct matrix. The matrix $\boldsymbol{W}$ can be calculated by minimizing the objective function:

$$\phi(\boldsymbol{W}) = \sum_i \left\| \boldsymbol{x}_i - \sum_j \boldsymbol{W}_{ij}\boldsymbol{x}_j \right\|,$$

with constraints $\sum_j \boldsymbol{W}_{ij} = 1$ ($j = 1, 2, \cdots, N$).

NPE believes that if the data points $\boldsymbol{x}_i$ in space R^D can be reconstructed by $\boldsymbol{W}_{ij}$, then the corresponding point $\boldsymbol{y}_i$ in low dimension space R^d can also be reconstructed by $\boldsymbol{W}_{ij}$. Therefore, the optimal mapping transformation matrix $\boldsymbol{A}_{opt}$ can be obtained by solving the minimization problem:

$$\boldsymbol{A}_{opt} = \arg\min \left[\sum_i \left\| \boldsymbol{y}_i - \sum_j \boldsymbol{W}_{ij}\boldsymbol{y}_j \right\|^2 \right].$$

With the algebraic transformation, the above minimization problem may be reduced as:

$$\boldsymbol{A}_{opt} = \arg\min_{\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}=1} \boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T\boldsymbol{A}. \tag{1}$$

And then the optimal mapping vectors are the solution of the generalized eigenvalue problem:

$$\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T\boldsymbol{a} = \lambda\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{a}. \tag{2}$$

The optimal mapping transformation matrix $\boldsymbol{A}_{opt}$ is composed of the optimal mapping transformation vectors, which are arranged in the order of the corresponding eigenvalues from small to large. Where

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2 \cdots, \boldsymbol{x}_N),\ \boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{W})^T(\boldsymbol{I} - \boldsymbol{W})$$

and $\boldsymbol{I}$ is an identity matrix.

3. Exponential Neighborhood Preserving Embedding

3.1 Matrix Exponential

In this section, we firstly introduce the following definition and theorems of matrix exponential [15]. Given an $n \times n$ square matrix A, its exponential is defined as follows:

$$\exp(\boldsymbol{A}) = \boldsymbol{I} + \boldsymbol{A} + \frac{\boldsymbol{A}^2}{2!} + \cdots + \frac{\boldsymbol{A}^k}{k!} + \cdots,$$

where $\boldsymbol{I}$ is the identity matrix with the size of $n \times n$. The properties of matrix exponential are listed as follows.

1) $\exp(\boldsymbol{A})$ is a finite matrix.

2) $\exp(\boldsymbol{A})$ is a full-rank matrix.

3) If square matrix $\boldsymbol{A}$ commutes with $\boldsymbol{B}$, i.e., $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}\boldsymbol{A}$, then

$$\exp(\boldsymbol{A} + \boldsymbol{B}) = \exp(\boldsymbol{A})\exp(\boldsymbol{B}).$$

4) For an arbitrary square matrix $\boldsymbol{A}$, there exists the inverse of $\exp(\boldsymbol{A})$. This is given by

$$(\exp(\boldsymbol{A}))^{-1} = \exp(-\boldsymbol{A}).$$

5) If T is a nonsingular matrix, then

$$\exp(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}) = \boldsymbol{T}^{-1}\exp(\boldsymbol{A})\boldsymbol{T}.$$

6) If $v_1, v_2, \cdots, v_n$ are eigenvectors of $\boldsymbol{A}$ that are related to the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$, then $v_1, v_2, \cdots, v_n$ are also eigenvectors of $\exp(\boldsymbol{A})$ that are related to the eigenvalues $e^{\lambda_1}, e^{\lambda_2}, \cdots, e^{\lambda_n}$ of $\exp(\boldsymbol{A})$.

3.2 The SSS Problem of NPE

Denote $\boldsymbol{S}_M = \boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T$, $\boldsymbol{S}_I = \boldsymbol{X}\boldsymbol{X}^T$, then generalized eigenvalue problem (2) can be rewritten as follows:

$$\boldsymbol{S}_M\boldsymbol{a} = \lambda\boldsymbol{S}_I\boldsymbol{a}.$$

About the ranks of $\boldsymbol{S}_M$ and $\boldsymbol{S}_I$, the following theorem

holds:

Theorem 1. Let N is the number of the samples, and D is dimension of the samples, if $D > N$, then the rank of $\boldsymbol{S}_M$ is at most $N-1$, and the rank of $\boldsymbol{S}_I$ is at most N, i.e.

$$rank(\boldsymbol{S}_M) \leq N-1 \text{ and } rank(\boldsymbol{S}_I) \leq N.$$

Proof. We have $\boldsymbol{S}_M = \boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T$, where $\boldsymbol{M} = (\boldsymbol{I}-\boldsymbol{W})^T(\boldsymbol{I}-\boldsymbol{W})$. Note that $\boldsymbol{W}$ is an N-order matrix, and the row elements of the matrix $\boldsymbol{W}$ are with constraint $\sum_j \boldsymbol{W}_{ij} = 1$ $(j = 1, 2, \cdots, N)$, so it easy to know that the rank of the matrix $\boldsymbol{I}-\boldsymbol{W}$ is at most $N-1$. It is known that the maximum possible rank of the product of two matrices is smaller than or equal to the smaller of the ranks of the two matrices. The rank of the matrix $\boldsymbol{M}$ is at most $N-1$, and then the rank of $\boldsymbol{S}_M$ is at most $N-1$, i.e.

$$rank(\boldsymbol{S}_M) \leq rank(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T) \leq N-1.$$

Similarly, it is easy to know that the rank of $\boldsymbol{S}_I$ is at most N, i.e.

$$rank(\boldsymbol{S}_I) \leq rank(\boldsymbol{X}\boldsymbol{X}^T) \leq N.$$

According to the above Theorem 1, both the matrices $\boldsymbol{S}_M = \boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T$ and $\boldsymbol{S}_I = \boldsymbol{X}\boldsymbol{X}^T$ can be singular. It is from the fact that, in most cases, the number of images in the training set N is much smaller than the dimensionality of the image D, i.e. $N \ll D$. This is known as SSS problem and NPE suffers from the difficulty.

3.3 Exponential NPE

According to the above Theorem 1, both the matrices $\boldsymbol{S}_M = \boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T$ and $\boldsymbol{S}_I = \boldsymbol{X}\boldsymbol{X}^T$ can be singular in most cases, it is from fact that there are some 0 eigenvalues. The exponential NPE (ENPE) is proposed to address the problem. It replaces the matrices $\boldsymbol{S}_M$ and $\boldsymbol{S}_I$ with the exponential of $\boldsymbol{S}_M$ and $\boldsymbol{S}_I$ respectively. According to Eq. (1), the criterion function of NPE can be rewritten as:

$$\min_{\boldsymbol{A}^T\boldsymbol{S}_I\boldsymbol{A}=1} \boldsymbol{A}^T\boldsymbol{S}_M\boldsymbol{A}. \tag{3}$$

Let $\boldsymbol{\Phi}_M = (\boldsymbol{\varphi}_{m1}, \boldsymbol{\varphi}_{m2}, \cdots, \boldsymbol{\varphi}_{mD})$ is the eigenvector matrix of $\boldsymbol{S}_M$ that corresponds to eigenvalue of $\boldsymbol{S}_M$:

$$\boldsymbol{\Lambda}_M = diag(\lambda_{m1}, \lambda_{m2}, \cdots, \lambda_{mD}),$$

and $\boldsymbol{\Phi}_I = (\boldsymbol{\varphi}_{i1}, \boldsymbol{\varphi}_{i2}, \cdots, \boldsymbol{\varphi}_{iD})$ is the eigenvector matrix of $\boldsymbol{S}_I$ that corresponds to eigenvalue of $\boldsymbol{S}_I$:

$$\boldsymbol{\Lambda}_I = diag(\lambda_{i1}, \lambda_{i2}, \cdots, \lambda_{iD}).$$

The criterion of NPE (3) can be rewritten as:

$$\min_{\boldsymbol{A}^T\boldsymbol{\Phi}_I^T\boldsymbol{\Lambda}_I\boldsymbol{\Phi}_T\boldsymbol{A}=1} \boldsymbol{A}^T\boldsymbol{\Phi}_M^T\boldsymbol{\Lambda}_M\boldsymbol{\Phi}_M\boldsymbol{A}. \tag{4}$$

In the criterion function (4), we replace the eigenvalue λ_{mj} of $\boldsymbol{S}_M$ with $\exp(\lambda_{mj})$, and λ_{ij} of $\boldsymbol{S}_I$ with $\exp(\lambda_{ij})$, and denote

$$\exp(\boldsymbol{\Lambda}_M) = diag(e^{\lambda_{m1}}, e^{\lambda_{m2}}, \cdots, e^{\lambda_{mD}}),$$
$$\exp(\boldsymbol{\Lambda}_I) = diag(e^{\lambda_{i1}}, e^{\lambda_{i2}}, \cdots, e^{\lambda_{iD}}).$$

The criterion of NPE can be transformed to:

$$\min_{\boldsymbol{A}^T\boldsymbol{\Phi}_I^T\exp(\boldsymbol{\Lambda}_I)\boldsymbol{\Phi}_T\boldsymbol{A}=1} \boldsymbol{A}^T\boldsymbol{\Phi}_M^T\exp(\boldsymbol{\Lambda}_M)\boldsymbol{\Phi}_M\boldsymbol{A},$$

i.e.,

$$\min_{\boldsymbol{A}^T\exp(\boldsymbol{S}_I)\boldsymbol{A}=1} \boldsymbol{A}^T\exp(\boldsymbol{S}_M)\boldsymbol{A}.$$

The columns vector of optimal transformation matrix $\boldsymbol{A}$, i.e., the optimal projection axes, can be obtained by solving the following generalized eigenvalue problem:

$$\exp(\boldsymbol{S}_M)\boldsymbol{a} = \lambda\exp(\boldsymbol{S}_I)\boldsymbol{a}. \tag{5}$$

However, in this paper, an important goal of selecting the optimal projected axes is to make that the method is low sensitivity to the neighborhood parameter k. Unfortunately, with some experiments, we find that the recognition performance is sensitive to the neighborhood size k. And so the following processing will be made.

Because the matrix $\exp(\boldsymbol{S}_I)$ is full-rank matrix, the eigenvalue problem (5) can be written as:

$$(\exp(\boldsymbol{S}_I))^{-1}\exp(\boldsymbol{S}_M)\boldsymbol{a} = \lambda\boldsymbol{a}. \tag{6}$$

Then, by the matrix exponential property 4), we have:

$$\exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M)\boldsymbol{a} = \lambda\boldsymbol{a}. \tag{7}$$

By the matrix exponential property 3), if the matrix equation $\boldsymbol{S}_M\boldsymbol{S}_I = \boldsymbol{S}_I\boldsymbol{S}_M$ holds, we have

$$\exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M) = \exp(\boldsymbol{S}_M - \boldsymbol{S}_I).$$

But it is easily to prove that there is $\boldsymbol{S}_M\boldsymbol{S}_I = (\boldsymbol{S}_I\boldsymbol{S}_M)^T$, not $\boldsymbol{S}_M\boldsymbol{S}_I = \boldsymbol{S}_I\boldsymbol{S}_M$, so the above equation does not hold.

However, in fact, the difference of $\exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M)$ and $\exp(\boldsymbol{S}_M - \boldsymbol{S}_I)$ is little. Let

$$\boldsymbol{S}_M\boldsymbol{S}_I = \boldsymbol{S}_I\boldsymbol{S}_M + \Delta\boldsymbol{T}, \tag{8}$$
$$\exp(\boldsymbol{S}_M - \boldsymbol{S}_I) = \exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M) + \Delta\boldsymbol{S}. \tag{9}$$

According to Theorem presented in Appendix, we have

$$\|\Delta\boldsymbol{S}\| \leq \frac{1}{2}\exp(\|\boldsymbol{S}_M\| + \|\boldsymbol{S}_I\|)\|\Delta\boldsymbol{T}\|. \tag{10}$$

The detailed proof of the estimate of the $\|\Delta\boldsymbol{S}\|$ may be referred in the Appendix.

In generally, the matrix exponential $\exp(\boldsymbol{A})$ usually expands the matrix $\boldsymbol{A}$. So we must normalize the matrices $\boldsymbol{S}_M$ and $\boldsymbol{S}_I$ because $\exp(\boldsymbol{S}_M)$ and $\exp(-\boldsymbol{S}_I)$ may involve larger values. And then, $\Delta\boldsymbol{T}$ in Eq. (8) is generally a little value matrix. So, in the Eq. (9), contrasted with the matrix $\exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M)$ and the matrix $\exp(\boldsymbol{S}_M - \boldsymbol{S}_I)$, $\Delta\boldsymbol{S}$ is a

Table 1 The measurement of the three matrices for different k (where $\boldsymbol{A} = \exp(\boldsymbol{S}_M - \boldsymbol{S}_I)$, $\boldsymbol{B} = \exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M)$)

The parameter k	2	12	22	32	42	52	62
$\|\boldsymbol{A}\|_F$	31.98534	31.98512	31.9865	31.98476	31.9832	31.98465	31.98431
$\|\boldsymbol{B}\|_F$	31.98529	31.98506	31.9851	31.98494	31.9849	31.98487	31.98483
$\|\Delta\boldsymbol{S}\|_F\ (\times 10^{-5})$	28.7	13.2	9.23	6.90	5.20	4.12	3.19

matrix whose element values are very little. So, it is feasible to replace the matrix $\exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M)$ with the matrix $\exp(\boldsymbol{S}_M - \boldsymbol{S}_I)$, and the error is very little.

An experiment has also been made to prove that. The experiment is made in ORL face database. In this experiment, for the different neighborhood parameter k, we compute $\exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M)$, $\exp(\boldsymbol{S}_M - \boldsymbol{S}_I)$ and $\Delta\boldsymbol{S}$, and use the Frobenius norm to measure the matrices. The experiment results are list in Table 1. We find that the Frobenius norm of $\Delta\boldsymbol{S}$, i.e., $\|\Delta\boldsymbol{S}\|_F$, is much little. Contrasted to the matrix $\exp(-\boldsymbol{S}_I)\exp(\boldsymbol{S}_M)$ and the matrix $\exp(\boldsymbol{S}_M - \boldsymbol{S}_I)$, $\Delta\boldsymbol{S}$ occupies a very small proportion.

And then the eigenvalue problem (7) may be replaced with the eigenvalue problem:

$$\exp(\boldsymbol{S}_M - \boldsymbol{S}_I)\boldsymbol{a} = \lambda\boldsymbol{a}. \tag{11}$$

Let the column vectors $\boldsymbol{a}_0, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_{d-1}$ be the solutions of Eq. (11), ordered by their corresponding eigenvalues, namely, $\lambda_0 \le \lambda_1 \le \cdots \le \lambda_{d-1}$, and let $\boldsymbol{A} = (\boldsymbol{a}_0, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_{d-1})$, then $\boldsymbol{A}$ is the optimal transformation matrix. This is ENPE method proposed in this paper.

4. Theoretical Analysis of ENPE

From the Sect. 3.2, NPE method suffers from the SSS problem. To overcome the complication of singular matrices, PCA can be adopted to reduce the dimensionality of the original image. But, due to the processing PCA step, some of significant information, which is contained in the original data, may be lost in the low-dimensional mapped data $\boldsymbol{y}$. The neighbor relationship of NPE is measured by the artificially constructed adjacent graph. Usually, k nearest neighbor criteria is adopted to construct the adjacent graph based on the preset neighborhood size k. However, for the different preset neighborhood size k, the eigenvalues gotten from the generalized eigenvalue problem (2) are different and the optimal projected axes are different greatly. This causes that NPE method is seriously sensitive to the neighborhood size k.

The proposed ENPE method may address the above problems effectively. On the one hand, obviously, the ENPE method has no the small-sample-size (SSS) problem. On the other hand, importantly, based on the eigenvalue problem (11), the ENPE method has low sensitivity to the neighborhood parameter k. Let us to discuss the problem in this section.

Table 2 The norm of the eigenvector difference for two different parameter k

the first seven	1	2	3	4	5	6	7
NPE	3.48	30.95	48.37	37.89	35.10	54.64	68.31
ENPE	0.001	1.999	0.065	0.159	0.159	0.086	0.162

According to the matrix exponential property 6), the matrix $\exp(\boldsymbol{S}_M - \boldsymbol{S}_I)$ and the matrix $\boldsymbol{S}_M - \boldsymbol{S}_I$ have the same eigenvectors. Consider the eigenvalue problem:

$$(\boldsymbol{S}_M - \boldsymbol{S}_I)\boldsymbol{a} = \lambda\boldsymbol{a}. \tag{12}$$

Based on the above discussion, the eigenvectors of the eigenvalue problem (11), i.e., the projection axes of ENPE, are from the eigenvectors of the eigenvalue problem (12) essentially.

And, in fact, the projection axes of NPE are from the eigenvectors of the eigenvalue problem:

$$\boldsymbol{S}_M\boldsymbol{a} = \lambda\boldsymbol{S}_I\boldsymbol{a}. \tag{13}$$

The eigenvalues of the Eq. (12) is from the solution of the eigen-polynomial:

$$\det(\lambda\boldsymbol{I} - \boldsymbol{S}_M - \boldsymbol{S}_I) = 0. \tag{14}$$

And the eigenvalues of the Eq. (13) is from the solution of the eigen-polynomial:

$$\det(\boldsymbol{S}_M - \lambda\boldsymbol{S}_I) = 0. \tag{15}$$

When the neighborhood parameter k varies, the difference of the eigenvalues gotten from the Eq. (14) is little, and the difference of the eigenvectors gotten is little. But the difference of the eigenvalues gotten from the Eq. (15) is large, and the difference of the eigenvectors gotten is large. To explain the problem, an experiment about the difference of the eigenvectors from two different parameter k is made to contrast the two methods. The experiment results are listed in the Table 2.

The experiment process is as follows. Two neighborhood parameters k_1 and k_2, are randomly preset. For the parameter k_1, we firstly compute the eigenvalues $\lambda_i^{NPEk_1}$ $(i = 1, 2, \cdots, t)$ of NPE. Where, as an example, the first seven eigenvectors, i.e., $t = 7$, are considered. Then get the eigenvector $\boldsymbol{v}_i^{NPEk_1}$ corresponding to the eigenvalues $\lambda_i^{NPEk_1}$ of NPE. And then, for the parameter k_2, we may get the eigenvector $\boldsymbol{v}_i^{NPEk_2}$. And then to compute the Frobenius norms of the vectors $\boldsymbol{v}_i^{NPEk_1} - \boldsymbol{v}_i^{NPEk_2}$, i.e.,

$$\left\|\boldsymbol{v}_i^{NPEk_1} - \boldsymbol{v}_i^{NPEk_2}\right\|_F \ (i = 1, 2, \cdots, t).$$

The results are listed on the second row of Table 2.

Similarly, for ENPE method, the Frobenius norms of vector differences for parameter k_1 and k_2 are computed, i.e.,

$$\left\|\boldsymbol{v}_i^{ENPEk_1} - \boldsymbol{v}_i^{ENPEk_2}\right\|_F \ (i = 1, 2, \cdots, t).$$

The results are listed on the third row of Table 2.

Based on the above discussion, for the different parameter k, the difference of the eigenvectors from ENPE is much smaller than that of NPE. The good property make that ENPE is less sensitive than NPE when the neighborhood size k varies.

5. Experiment Results

The experiments are made on the ORL, Georgia Tech and AR face database respectively. For each experiment, it is made from two aspects. One is to measure the sensitivity of ENPE method to the neighborhood parameter k, another is to evaluate the face recognition performance of the proposed ENPE method. In the experiments, the proposed ENPE method is compared with the classical PCA, and the unsupervised manifold learning methods, including LPP [8], exponential LPP (ELPP) [19] and NPE [9]. For the methods suffering from the SSS problem (LPP and NPE), PCA technique is firstly used to reduce the dimension of the original image vector.

5.1 Face Database

1) ORL face database

The ORL database contains 400 images of 40 persons (10 images per person). For some subjects, the images were taken at different times, varying the lighting, facial expressions (open or closed eyes, smiling or not smiling) and facial details (glasses or no glasses). In our experiment, each image is manually cropped and resized to 32×32 pixels.

2) Georgia Tech face database

Georgia Tech face database contains images of 50 people taken in two or three sessions between 06/01/99 and 11/15/99 at the Center for Signal and Image Processing at Georgia Institute of Technology. All people in the database are represented by 15 color JPEG images with cluttered background. In our experiment, each image is manually cropped and resized to 32×32 pixels.

3) AR face database

This face database contains over 4000 color images corresponding to 126 people's faces. Images feature frontal view faces with different facial expressions, illumination conditions and occlusions. The pictures were taken under strictly controlled conditions. In our experiment, each image is manually cropped and resized to 40×40 pixels.

5.2 Experiment Results on ORL Face Database

The experiment is conducted on ORL face Database. A random subset with p (p = 2, 3, 4, 5, 6) images for each individual was taken to form the training set, and the remaining were used as the testing set. For a fixed p, the above random split process is repeated 20 times to obtain stable recognition results. And for a fixed training sample p and a fixed split, the neighborhood parameter k is searched from $\{2, 3, \cdots, N-1\}$ and with step size = 5, i.e., $k = \left\{2, 7, 12 \cdots, 2 + \left[\frac{N-1}{5}\right] \times 5\right\}$, where N is the training sample number. For convenience, denote $l = \left[\frac{N-1}{5}\right] + 1$.

Table 3 The MMDs of four methods on ORL face database

Method	2 trains	3 trains	4 trains	5 trains	6 trains
LPP	5.688	9.716	14.834	19.000	22.248
ELPP	6.064	7.928	5.830	5.800	4.252
NPE	4.186	8.070	10.836	10.600	12.124
ENPE	2.934	2.572	1.500	0.900	1.378

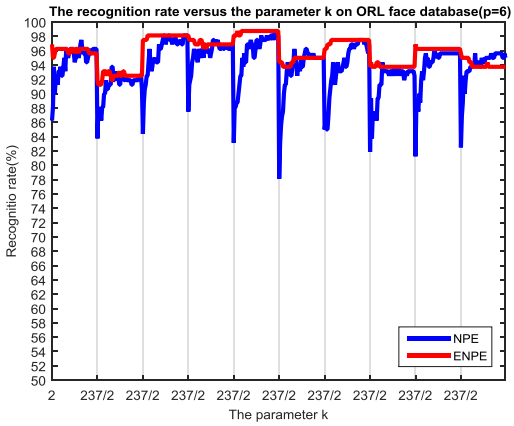

Fig. 1 The comparison of NPE and ENPE versus the parameter k for ten random splits on ORL face database

Firstly, we evaluate the face recognition sensitivity to the parameter k for four local manifold learning algorithms: LPP, ELPP, NPE and ENPE. A criterion to measure the sensitivity to the parameter k, mean maximum difference (MMD), is presented in [19]. For a fixed p, there are 20 random splits in the experiments and each split includes l recognition ratios versus l values of k. So, there are maximum value and minimum value in the l recognition accuracies for each split. The maximum difference of recognition accuracy is obtained by subtracting the minimum accuracy from the maximum accuracy for each split. The mean maximum difference criterion (MMD) is the mean value of 20 maximum differences from the 20 random splits. According the definition, the smaller value of MMD means that the algorithm is more stable and more insensitive corresponding to the parameter k.

The experiment results are listed in Table 3, where the face subspace dimension is 40. For the other subspace dimensions, the similar results can be gotten. From the Table 3, we observe the following facts: 1) the MMDs of ENPE are less than that of other methods and this shows that the performance of ENPE is much less sensitive to k than that of other methods, 2) the matrix exponential methods ELPP and ENPE are much less sensitive to k than that of LPP and NPE.

Furth more, the following Fig. 1 is used to compare the sensitivity to the neighborhood parameter k of NPE and

Table 4 The performance comparison of five methods on ORL database

Method	2 trains(%)	3 trains(%)	4 trains(%)	5 trains(%)
PCA	84.06(60)	85.39(80)	87.89(70)	88.78(55)
LPP	90.00(10)	87.71(40)	91.92(45)	93.10(40)
ELPP	87.81(25)	88.99(50)	94.08(35)	95.81(45)
NPE	90.25(40)	90.21(50)	94.33(45)	95.42(55)
ENPE	90.94(40)	91.14(70)	94.50(55)	95.73(45)

ENPE methods. In the Fig. 1, ten recognition rate curves versus the parameter k are plotted for ten times random splits when the train sample $p = 6$. The horizontal axis shows the neighborhood parameter k for 10 times random splits. The vertical line is used to separate the recognition rate curve of each random split. Between two vertical lines, the value of k is from 2 to maximum. Where, for $p = 6$, the maximum of k is 237, so the value of k is from 2 to 237.

From Fig. 1, NPE shows a fluctuant trend and ENPE shows a more stable trend. This means that the performance of ENPE is much less sensitive to the parameter k than that of NPE. And in most cases, the recognition accuracies of ENPE are better than that of NPE.

Secondly, we evaluate the performance of the five unsupervised methods: PCA, LPP, ELPP, NPE and the proposed ENPE. In this experiment, for each training sample p ($p = 2, 3, 4, 5$), 20 random splits have been made. There are l recognition accuracies corresponding to the l values of k for each split. For each random split, we report the top-1 recognition accuracy from the best parameter k configuration. And so, there 20 maximal recognition accuracies, then we get the average value of the 20 maximal recognition accuracies and regard it as the recognition rate of the corresponding method. In general, the recognition performance varies with the dimension of the face subspace. In the experiment, let the dimension is from a range of dimensions. For every subspace dimension, the above process is repeated to calculate the recognition rate. The best average performance obtained by the five methods as well as the corresponding dimension is summarized in Table 4. The number appearing in parenthesis is the optimal subspace dimension. From Table 4, the recognition rates of ENPE are better than that of PCA, LPP, ELPP and NPE.

5.3 Experiment Results on Georgia Tech Face Database

Georgia Tech face database is more complex than ORL database, because it contains various pose faces with different expressions on cluttered background. In this experiment, a random subset with p ($p = 2, 4, 6, 8, 10$) images for everyone was taken to form the training set, and the remaining were used as the testing set. For a fixed p, the above random split process is repeated 20 times to obtain stable recognition results. And for a fixed p and a fixed split, the neighborhood parameter k is searched from $\{2, 3, \cdots, N-1\}$ and the step size = 5, i.e., $k = \left\{2, 7, 12 \cdots, 2 + \left[\frac{N-1}{5}\right] \times 5\right\}$.

Table 5 The MMDs of four methods on Georgia Tech face database

Method	2 trains	4 trains	6 trains	8 trains	10 trains
LPP	6.248	28.764	38.002	40.800	48.680
ELPP	7.014	9.710	9.380	10.486	12.320
NPE	11.262	24.290	36.402	38.318	45.760
ENPE	3.478	2.510	1.956	1.886	1.160

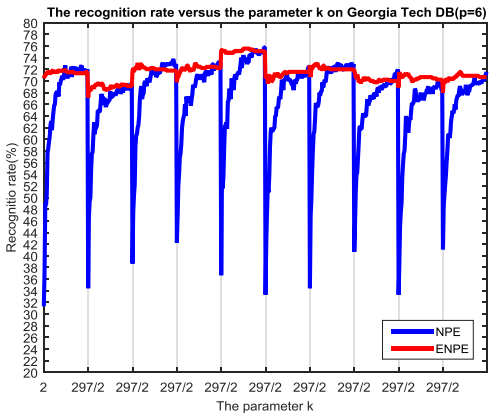

Fig. 2 The comparison of NPE and ENPE versus the parameter k for ten random splits on Georgia Tech face database

In the same way, the mean maximum difference (MMD) criterion is used to measure the face recognition sensitivity with respect to the neighborhood parameter k. As an illustration, the face subspace dimension is set to 40. The experiment results are shown in Table 5. As shown, the performance of ENPE is less sensitive to the neighborhood parameter k than that of other methods.

The recognition rate of NPE and ENPE versus the parameter k for ten times random splits when $p = 6$ are shown in Fig. 2. The legend is the same as the description in the Sect. 5.2. The horizontal axis shows the neighborhood parameter k for ten times random splits. The value of k is from 2 to maximum for each split. This also means that the performance of ENPE is much less sensitive to the parameter k than that of NPE. And in most cases, the recognition accuracies of ENPE are better than that of NPE.

In this experiment, we also evaluate the performance of the five methods: PCA, LPP, ELPP, NPE and ENPE. The subspace dimension is from a range of dimensions. The best average performance corresponding dimension is summarized in the Table 6. From the Table 6, the recognition rates of ENPE are better than that of other four methods in the complex face database.

5.4 Experiment Results on AR Face Database

AR face database is more larger and complex face database. In this experiment, a random subset with p ($p = 3, 4, 5, 6, 7$) images for everyone was taken to form the training set, and the remaining were used as the testing set. For a fixed p, the above random split process is repeated

Table 6 The performance comparison of five methods on Georgia Tech face database

Method	2 trains(%)	4 trains(%)	6 trains(%)	8 trains(%)
PCA	53.08(80)	67.78(30)	73.14(60)	76.92(85)
LPP	49.85(15)	55.34(60)	58.75(40)	66.21(30)
ELPP	57.35(15)	70.32(100)	75.29(25)	80.35(25)
NPE	56.98(15)	69.10(100)	74.93(65)	81.77(45)
ENPE	57.97(15)	69.42(100)	75.20(25)	82.85(80)

Table 7 The MMDs of four methods on AR face database

Method	3 trains	4 trains	5 trains	6 trains	7 trains
LPP	18.894	20.766	25.390	26.522	30.046
ELPP	17.562	14.802	11.722	10.980	7.144
NPE	23.472	25.154	31.388	33.960	41.024
ENPE	0.621	0.530	0.518	0.453	0.410

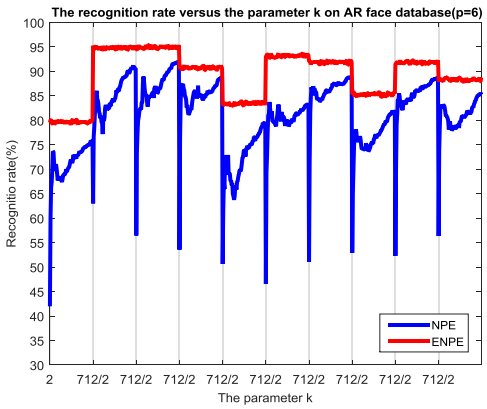

Fig. 3 The comparison of NPE and ENPE versus the parameter k for ten random splits on AR face database

10 times to obtain stable recognition results. And for a fixed p and a fixed split, the neighborhood parameter k is searched from $\{2, 3, \cdots, N-1\}$ and the step size = 10, i.e., $k = \left\{2, 12, \cdots, 2 + \left[\frac{N-1}{10}\right] \times 10\right\}$.

The mean maximum difference (MMD) criterion is also used to measure the face recognition sensitivity with respect to the neighborhood parameter k.

The experiment results are shown in Table 7. From Table 7, the MMD values of ENPE are the smallest than that of LPP, ELPP and NPE. This means that the performance of ENPE is less sensitive to k than that of other methods. And interestingly, the recognition performance of ENPE is almost constant for all the training samples. Where, the face subspace dimension is set to 40.

In fact, the similar results may be gotten for the other subspace dimension. Figure 3 shows the recognition rate versus the parameter k for 10 times random splits when the training sample $p = 6$. Obviously, ENPE shows a more stable trend than that of NPE. This also proves that the performance of ENPE is much less sensitive to the parameter k than that of NPE. And in most cases, the recognition accuracies of ENPE are better than that of NPE.

Table 8 The performance comparison of five methods on AR database

Method	3 trains(%)	4 trains(%)	5 trains(%)	6 trains(%)
PCA	79.02(70)	83.48(90)	83.58(80)	84.74(80)
LPP	81.69(100)	83.97(80)	84.79(60)	85.23(50)
ELPP	82.55(90)	84.14(50)	85.96(90)	85.76(50)
NPE	84.37(100)	88.30(50)	86.05(90)	86.99(50)
ENPE	85.05(100)	89.45(80)	90.25(70)	90.59(60)

In this experiment, we also evaluate the performance of the five methods: PCA, LPP, ELPP, NPE and ENPE. The best average performance corresponding subspace dimension is summarized in Table 8. The experiment on AR face database also illustrates that ENPE is more effective than other four methods.

6. Conclusions

In this paper, an exponential neighborhood preserving embedding (ENPE) is proposed to improve NPE method. The main idea of ENPE is that the matrix exponential is introduced to NPE. Unlike the NPE method, the proposed ENPE method has no the small-sample-size problem, and shows much stable recognition performance when the neighborhood parameter varies. The experiments are conducted on three public face databases: ORL, Georgia Tech and AR. In our experiments, ENPE is compared with the methods without discriminant information: PCA, LPP, the improved LPP with matrix exponential (ELPP) and NPE. The experiment results prove that ENPE has two superiorities: 1) NPE shows advantageous performance over the above methods in face recognition. 2) Compared with the manifold learning methods LPP, ELPP and NPE, ENPE is much less sensitive to the neighborhood parameter k.

Acknowledgements

This research was supported by National Nature Science Foundation of China (61472053, 91420102), the Natural Science Foundation Project of CQ CSTC of China (cstc2016jcyjA0419) and the School Fund Project of CQNU (16XLB006, 16XZH07).

References

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol.3, no.1, pp.71–86, 1991.
[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no.7, pp.711–720, 1997.
[3] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," Science., vol.290, no.5500, pp.2323–2326, 2000.
[4] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," Science, vol.290, no.5500, pp.2319–2323, 2000.

[5] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," Neural Computation, vol.15, no.6, pp.1373–1396, 2003.
[6] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," SIAM Journal on Scientific Computing, vol.26, no.1, pp.313–338, 2005.
[7] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Roux, and M. Ouimet, "Out-of sample extensions for LLE, ISOMAP, MDS, Eigenmaps, and spectral clustering," Advances in Neural Information Processing Systems (NIPS), Cambridge, pp.177–184, 2003.
[8] X.F. He and P. Niyogi, "Locality preserving projections," Advances in Neural Information Processing Systems (NIPS), Vancouver, vol.16, pp.153–160, 2004.
[9] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," Proc. International Conference on Computer Vision (ICCV), Beijing, pp.1208–1213, 2005.
[10] G.E. Hinton and S.T. Roweis, "Stochastic Neighbor Embedding," Advances in Neural Information Processing Systems, vol.15, pp.833–840, The MIT Press, Cambridge, MA, USA, 2002.
[11] V. Maaten and G. Hinton, "Visualizing data using t-SNE," J. of Machine Learning Research, vol.9, pp.2579–2605, 2008.
[12] Y.P. Zhou, Y.L. Ding, Y.F. Luo, and H.L. Ren, "Sparse Neighborhood Preserving Embedding via L2,1-Norm Minimization," 2016 9th International Symposium on Computational Intelligence and Design (ISCID), vol.2, pp.378–382, 2016.
[13] B. Song, S. Tan, and H.B. Shi, "Process monitoring via enhanced neighborhood preserving embedding," Control Engineering Practice, vol.50, pp.48–56, 2016.
[14] X.Q. Zhao and T. Wang, "Tensor dynamic neighborhood preserving embedding algorithm for fault diagnosis of batch process," Chemometrics and Intelligent Laboratory Systems, vol.162, pp.94–103, 2017.
[15] T.P. Zhang, B. Fang, Y.Y. Tang, Z. Shang, and B. Xu, "Generalized discriminant analysis: A matrix exponential approach," IEEE Trans. Syst. Man, Cybern. B, Cybern., vol.40, no.1, pp.186–197, 2010.
[16] W.W. Yu, X.L. Teng, and C.Q. Liu, "Face recognition using discriminant locality preserving projections," Image Vis. Comput., vol.24, no.3, pp.239–248, 2006.
[17] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., San Diego, pp.846–853, 2005.
[18] H. Huang, J. Liu, and Y. Pan, "Semi-supervised marginal fisher analysis for hyperspectral image classification," ISPRS Ann. Photogrammetry-Remote Sens. Spatial Inf. Sci., vol.I-3, pp.377–382, 2012.
[19] S.-J. Wang, H.-L. Chen, X.-J. Peng, and C.-G. Zhou, "Exponential locality preserving projections for small sample size problem," Neurocomputing, vol.74, no.17, pp.3654–3662, 2011.
[20] S.C Huang and L. Zhuang, "Exponential Discriminant Locality Preserving Projection for face recognition," Neurocomputing, vol.208, pp.373–377, 2016.
[21] F. Dornaika and A. Bosaghzadeh, "Exponential Local Discriminant Embedding and Its Application to Face Recognition," IEEE Trans. Cybern., vol.43, no.3, pp.921–934, 2013.
[22] F. Dornaika and Y. El Traboulsi, "Matrix exponential based semi-supervised discriminant embedding for image classification," Pattern Recognition, vol.61, pp.92–103, 2017.
[23] P. Wittek, Quantum Machine Learning, Chapter 5, pp.57–62, Academic Press, Cambridge, Massachusetts, USA, 2014.
[24] J. Yao, Q. Mao, S. Goodison, V. Mai, and Y.J. Sun, "Feature selection for unsupervised learning through local learning," Pattern Recognition Letters, vol.53, pp.100–107, 2015.

Appendix:

According to the matrix exponential property 3), if square matrix $\boldsymbol{A}$ commutes with $\boldsymbol{B}$, i.e., $\boldsymbol{AB} = \boldsymbol{BA}$, then

$$\exp(\boldsymbol{A} + \boldsymbol{B}) = \exp(\boldsymbol{A})\exp(\boldsymbol{B}).$$

But, if $\boldsymbol{AB} \neq \boldsymbol{BA}$, the above equation does not hold. Let

$$\exp(\boldsymbol{A} + \boldsymbol{B}) = \exp(\boldsymbol{A})\exp(\boldsymbol{B}) + \Delta\boldsymbol{E},$$

where $\Delta\boldsymbol{E}$ is the error matrix. In the Appendix, it is proved that $\|\Delta\boldsymbol{E}\|$ is much little, so the error matrix $\Delta\boldsymbol{E}$ is a little value matrix, and then $\exp(\boldsymbol{A} + \boldsymbol{B})$ may be approximated by $\exp(A)\exp(B)$.

Denote

$$\boldsymbol{AB} = \boldsymbol{BA} + \Delta\boldsymbol{C},$$

where $\Delta\boldsymbol{C}$ is the error matrix between the matrix $\boldsymbol{AB}$ and $\boldsymbol{BA}$. And we made the following rule:

$$\text{if } n < m,\ \sum_{i=m}^{n}(*) = 0.$$

Additionally, let the norm $\|\bullet\|$ used in this paper be compatible norm.

By the definition of the matrix exponential, we have

$$\exp(\boldsymbol{A} + \boldsymbol{B}) = \sum_{k=0}^{\infty}\frac{(\boldsymbol{A} + \boldsymbol{B})^k}{k!} = \sum_{k=0}^{\infty}\frac{1}{k!}\boldsymbol{P}_k, \tag{A·1}$$

for convenience, denote $\boldsymbol{P}_k = (\boldsymbol{A} + \boldsymbol{B})^k$.

On the other hand, it is easy to know that the series:

$$\sum_{i=0}^{\infty}\frac{\boldsymbol{A}^i}{i!} = \boldsymbol{I} + \boldsymbol{A} + \frac{\boldsymbol{A}^2}{2!} + \cdots + \frac{\boldsymbol{A}^k}{k!} + \cdots$$

is an absolutely convergent series. And then, by the multiplication theorem of absolutely convergent series, we have

$$\begin{aligned}\exp(\boldsymbol{A})\exp(\boldsymbol{B}) &= \sum_{i=0}^{\infty}\frac{\boldsymbol{A}^i}{i!}\sum_{j=0}^{\infty}\frac{\boldsymbol{B}^j}{j!} = \sum_{k=0}^{\infty}\left[\sum_{l=0}^{k}\frac{\boldsymbol{A}^{k-l}}{(k-l)!}\frac{\boldsymbol{B}^l}{l!}\right] \\ &= \sum_{k=0}^{\infty}\left[\frac{1}{k!}\sum_{l=0}^{k}\binom{k}{l}\boldsymbol{A}^{k-l}\boldsymbol{B}^l\right] = \sum_{k=0}^{\infty}\left[\frac{1}{k!}\boldsymbol{Q}_k\right],\end{aligned} \tag{A·2}$$

for convenience, denote $\boldsymbol{Q}_k = \sum_{l=0}^{k}\binom{k}{l}\boldsymbol{A}^{k-l}\boldsymbol{B}^l$.

Because of $\boldsymbol{AB} \neq \boldsymbol{BA}$, we have $\boldsymbol{P}_k \neq \boldsymbol{Q}_k$. About the error between the matrix $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$, the following conclusion holds.

Lemma. Let $\Delta\boldsymbol{H}_k$ be the error matrix between the matrix $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$, i.e.,

$$\boldsymbol{P}_k = \boldsymbol{Q}_k - \Delta\boldsymbol{H}_k, \tag{A·3}$$

then the error matrix $\Delta\boldsymbol{H}_k$ may be formulated as:

$$\Delta\boldsymbol{H}_k = \sum_{j=0}^{k-1}\binom{k-1}{j}\sum_{r=1}^{k-1-j}\boldsymbol{A}^{k-1-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^j$$

$$
\begin{aligned}
&+ (\boldsymbol{A}+\boldsymbol{B})\sum_{j=0}^{k-2}\binom{k-2}{j}\sum_{r=1}^{k-2-j}\boldsymbol{A}^{k-2-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j}\\
&+ \cdots\cdots\\
&+ (\boldsymbol{A}+\boldsymbol{B})^{k-2}\Delta\boldsymbol{C}\\
=&\sum_{i=0}^{k-2}\left((\boldsymbol{A}+\boldsymbol{B})^{i}\sum_{j=0}^{k-1-i}\binom{k-1-i}{j}\sum_{r=1}^{k-1-i-j}\boldsymbol{A}^{k-1-i-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j}\right)
\end{aligned}
\tag{A·4}
$$

Proof. Proof by mathematical induction.

1) when $k = 0, 1, 2$,

If $k = 0$, $\Delta\boldsymbol{H}_k = 0$, Eq. (A·3) and Eq. (A·4) hold.

If $k = 1$, $\Delta\boldsymbol{H}_k = 0$, Eq. (A·3) and Eq. (A·4) hold.

If $k = 2$, $\Delta\boldsymbol{H}_k = \Delta\boldsymbol{C}$, Eq. (A·3) and Eq. (A·4) hold.

2) Let Eq. (A·3) and Eq. (A·4) hold when $k = m$.

3) when $k = m + 1$, we have

$$
\begin{aligned}
&\boldsymbol{P}_{m+1} = (\boldsymbol{A}+\boldsymbol{B})^{m+1} = (\boldsymbol{A}+\boldsymbol{B})\boldsymbol{P}_m\\
&= (\boldsymbol{A}+\boldsymbol{B})(\boldsymbol{Q}_m - \Delta\boldsymbol{H}_m)\\
&= (\boldsymbol{A}+\boldsymbol{B})\sum_{l=0}^{m}\binom{m}{l}\boldsymbol{A}^{m-l}\boldsymbol{B}^{l} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \boldsymbol{A}\sum_{l=0}^{m}\binom{m}{l}\boldsymbol{A}^{m-l}\boldsymbol{B}^{l} + \boldsymbol{B}\sum_{j=0}^{m}\binom{m}{j}\boldsymbol{A}^{m-j}\boldsymbol{B}^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \sum_{l=0}^{m}\binom{m}{l}\boldsymbol{A}^{m-l+1}\boldsymbol{B}^{l} + \sum_{j=0}^{m}\binom{m}{j}\boldsymbol{B}\boldsymbol{A}^{m-j}\boldsymbol{B}^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \boldsymbol{A}^{m+1} + \sum_{l=1}^{m}\binom{m}{l}\boldsymbol{A}^{m-l+1}\boldsymbol{B}^{l} + \sum_{j=0}^{m}\binom{m}{j}\boldsymbol{B}\boldsymbol{A}^{m-j}\boldsymbol{B}^{j}\\
&\quad - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m.
\end{aligned}
\tag{A·5}
$$

The item $\sum_{j=0}^{m}\binom{m}{j}\boldsymbol{B}\boldsymbol{A}^{m-j}\boldsymbol{B}^{j}$ of the above formula may be written as:

$$
\begin{aligned}
&\sum_{j=0}^{m}\binom{m}{j}\boldsymbol{B}\boldsymbol{A}^{m-j}\boldsymbol{B}^{j}\\
&= \sum_{j=0}^{m}\binom{m}{j}(\boldsymbol{A}\boldsymbol{B} - \Delta\boldsymbol{C})\boldsymbol{A}^{m-j-1}\boldsymbol{B}^{j}\\
&= \sum_{j=0}^{m}\binom{m}{j}(\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}^{m-j-1} - \Delta\boldsymbol{C}\boldsymbol{A}^{m-j-1})\boldsymbol{B}^{j}\\
&= \sum_{j=0}^{m}\binom{m}{j}(\boldsymbol{A}(\boldsymbol{A}\boldsymbol{B} - \Delta\boldsymbol{C})\boldsymbol{A}^{m-j-2} - \Delta\boldsymbol{C}\boldsymbol{A}^{m-j-1})\boldsymbol{B}^{j}\\
&= \sum_{j=0}^{m}\binom{m}{j}(\boldsymbol{A}^{2}\boldsymbol{B}\boldsymbol{A}^{m-j-2} - \boldsymbol{A}\Delta\boldsymbol{C}\boldsymbol{A}^{m-j-2} - \Delta\boldsymbol{C}\boldsymbol{A}^{m-j-1})\boldsymbol{B}^{j}\\
&\cdots\cdots\\
&= \sum_{j=0}^{m}\binom{m}{j}(\boldsymbol{A}^{m-j}\boldsymbol{B} - \boldsymbol{A}^{m-j-1}\Delta\boldsymbol{C} - \boldsymbol{A}^{m-j-2}\Delta\boldsymbol{C}\boldsymbol{A} - \cdots\\
&\quad - \Delta\boldsymbol{C}\boldsymbol{A}^{m-j-1})\boldsymbol{B}^{j}\\
&= \sum_{j=0}^{m}\binom{m}{j}\left(\boldsymbol{A}^{m-j}\boldsymbol{B} - \sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\right)\boldsymbol{B}^{j}\\
&= \sum_{j=0}^{m}\binom{m}{j}\boldsymbol{A}^{m-j}\boldsymbol{B}^{j+1} - \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j}.
\end{aligned}
$$

So, Eq. (A·5) becomes

$$
\begin{aligned}
\boldsymbol{P}_{m+1} &= \boldsymbol{A}^{m+1} + \sum_{l=1}^{m}\binom{m}{l}\boldsymbol{A}^{m-l+1}\boldsymbol{B}^{l} + \sum_{j=0}^{m}\binom{m}{j}\boldsymbol{A}^{m-j}\boldsymbol{B}^{j+1}\\
&\quad - \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}B^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \boldsymbol{A}^{m+1} + \sum_{l=1}^{m}\binom{m}{l}\boldsymbol{A}^{m-l+1}\boldsymbol{B}^{l} + \sum_{l=1}^{m+1}\binom{m}{l-1}\boldsymbol{A}^{m-l+1}\boldsymbol{B}^{l} \qquad (l = j+1)\\
&\quad - \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \boldsymbol{A}^{m+1} + \sum_{l=1}^{m}\binom{m}{l}\boldsymbol{A}^{m-l+1}\boldsymbol{B}^{l}\\
&\quad + \sum_{l=1}^{m}\binom{m}{l-1}\boldsymbol{A}^{m-l+1}\boldsymbol{B}^{l} + \boldsymbol{B}^{m+1}\\
&\quad - \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \boldsymbol{A}^{m+1} + \sum_{l=1}^{m}\left(\binom{m}{l} + \binom{m}{l-1}\right)\boldsymbol{A}^{m+1-l}\boldsymbol{B}^{l} + \boldsymbol{B}^{m+1}\\
&\quad - \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \boldsymbol{A}^{m+1} + \sum_{l=1}^{m}\binom{m+1}{l}\boldsymbol{A}^{m+1-l}\boldsymbol{B}^{l} + \boldsymbol{B}^{m+1} \qquad \text{(Pascal's law)}\\
&\quad - \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \sum_{l=0}^{m+1}\binom{m+1}{l}\boldsymbol{A}^{m+1-l}\boldsymbol{B}^{l}\\
&\quad - \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m\\
&= \boldsymbol{Q}_{m+1} - \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j} - (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m.
\end{aligned}
$$

Denote

$$
\Delta\boldsymbol{H}_{m+1} = \sum_{j=0}^{m}\binom{m}{j}\sum_{r=1}^{m-j}\boldsymbol{A}^{m-j-r}\Delta\boldsymbol{C}\boldsymbol{A}^{r-1}\boldsymbol{B}^{j} + (\boldsymbol{A}+\boldsymbol{B})\Delta\boldsymbol{H}_m.
$$

When $k = m$, Eq. (A·3) and Eq. (A·4) hold, replace $\Delta\boldsymbol{H}_m$ of

the above equation with the expression of $\Delta \boldsymbol{H}_m$, then

$$\begin{aligned}
&\Delta \boldsymbol{H}_{m+1} \\
&= \sum_{j=0}^{m} \binom{m}{j} \sum_{r=1}^{m-j} \boldsymbol{A}^{m-j-r} \Delta \boldsymbol{C} \boldsymbol{A}^{r-1} \boldsymbol{B}^j \\
&\quad + (\boldsymbol{A}+\boldsymbol{B}) \sum_{j=0}^{m-1} \binom{m-1}{j} \sum_{r=1}^{m-1-j} \boldsymbol{A}^{m-1-j-r} \Delta \boldsymbol{C} \boldsymbol{A}^{r-1} \boldsymbol{B}^j \\
&\quad + \cdots\cdots \\
&\quad + (\boldsymbol{A}+\boldsymbol{B})^{m-1} \Delta \boldsymbol{C} \\
&= \sum_{i=0}^{m-1} \left((\boldsymbol{A}+\boldsymbol{B})^i \sum_{j=0}^{m-i} \binom{m-i}{j} \sum_{r=1}^{m-i-j} \boldsymbol{A}^{m-i-j-r} \Delta \boldsymbol{C} \boldsymbol{A}^{r-1} \boldsymbol{B}^j \right)
\end{aligned}$$

□

Based on the Lemma, the norm $\|\Delta \boldsymbol{E}\|$ of the error matrix $\Delta \boldsymbol{E}$ may be measured.

Theorem. If $\boldsymbol{AB} \neq \boldsymbol{BA}$, let

$$\exp(\boldsymbol{A}+\boldsymbol{B}) = \exp(\boldsymbol{A}) \exp(\boldsymbol{B}) + \Delta \boldsymbol{E},$$

then the norm of the error matrix ΔE may be estimated as:

$$\|\Delta \boldsymbol{E}\| \leq \frac{1}{2} \exp(\|\boldsymbol{A}\| + \|\boldsymbol{B}\|) \|\Delta \boldsymbol{C}\|.$$

Proof. According to Eq. (A· 1) and Eq. (A· 2),

$$\begin{aligned}
\Delta \boldsymbol{E} &= \exp(\boldsymbol{A}+\boldsymbol{B}) - \exp(\boldsymbol{A}) \exp(\boldsymbol{B}) \\
&= \sum_{k=0}^{\infty} \frac{1}{k!} (\boldsymbol{P}_k - \boldsymbol{Q}_k) \\
&= \sum_{k=0}^{\infty} \frac{1}{k!} \Delta \boldsymbol{H}_k \\
&= \sum_{k=0}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} \left((\boldsymbol{A}+\boldsymbol{B})^i \sum_{j=0}^{k-1-i} \binom{k-1-i}{j} \sum_{r=1}^{k-1-i-j} \boldsymbol{A}^{k-1-i-j-r} \Delta \boldsymbol{C} \boldsymbol{A}^{r-1} \boldsymbol{B}^j \right),
\end{aligned}$$

then, we have

$$\begin{aligned}
\|\Delta E\| &\leq \sum_{k=2}^{\infty} \frac{1}{k!} \left\| \sum_{i=0}^{k-2} \left((\boldsymbol{A}+\boldsymbol{B})^i \sum_{j=0}^{k-1-i} \binom{k-1-i}{j} \sum_{r=1}^{k-1-i-j} \boldsymbol{A}^{k-1-i-j-r} \Delta \boldsymbol{C} \boldsymbol{A}^{r-1} \boldsymbol{B}^j \right) \right\| \\
&\leq \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} \left(\left\| (\boldsymbol{A}+\boldsymbol{B})^i \right\| \left\| \sum_{j=0}^{k-1-i} \binom{k-1-i}{j} \sum_{r=1}^{k-1-i-j} \boldsymbol{A}^{k-1-i-j-r} \Delta \boldsymbol{C} \boldsymbol{A}^{r-1} \boldsymbol{B}^j \right\| \right) \\
&\leq \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} \left((\|\boldsymbol{A}+\boldsymbol{B}\|)^i \sum_{j=0}^{k-1-i} \binom{k-1-i}{j} \sum_{r=1}^{k-1-i-j} \left\| \boldsymbol{A}^{k-1-i-j-r} \right\| \|\Delta \boldsymbol{C}\| \left\| \boldsymbol{A}^{r-1} \right\| \left\| \boldsymbol{B}^j \right\| \right) \\
&\leq \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} \left((\|\boldsymbol{A}+\boldsymbol{B}\|)^i \sum_{j=0}^{k-1-i} \binom{k-1-i}{j} \sum_{r=1}^{k-1-i-j} (\|\boldsymbol{A}\|)^{k-2-i-j} \|\Delta \boldsymbol{C}\| (\|\boldsymbol{B}\|)^j \right) \\
&= \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} \left((\|\boldsymbol{A}+\boldsymbol{B}\|)^i \sum_{j=0}^{k-1-i} \binom{k-1-i}{j} (k-1-i-j) (\|\boldsymbol{A}\|)^{k-2-i-j} (\|\boldsymbol{B}\|)^j \|\Delta \boldsymbol{C}\| \right) \\
&= \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} \left((\|\boldsymbol{A}+\boldsymbol{B}\|)^i \sum_{j=0}^{k-2-i} \binom{k-1-i}{j} (k-1-i-j) (\|\boldsymbol{A}\|)^{k-2-i-j} (\|\boldsymbol{B}\|)^j \|\Delta \boldsymbol{C}\| \right) \\
&= \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} \left((\|\boldsymbol{A}+\boldsymbol{B}\|)^i (k-1-i) \left(\sum_{j=0}^{k-2-i} \binom{k-2-i}{j} (\|\boldsymbol{A}\|)^{k-2-i-j} (\|\boldsymbol{B}\|)^j \right) \|\Delta \boldsymbol{C}\| \right) \\
&\leq \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} \left(((\|\boldsymbol{A}\|+\|\boldsymbol{B}\|)^i (k-1-i) (\|\boldsymbol{A}\|+\|\boldsymbol{B}\|)^{k-2-i}) \|\Delta \boldsymbol{C}\| \right) \\
&= \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} ((k-1-i)(\|\boldsymbol{A}\| + \|\boldsymbol{B}\|)^{k-2}) \|\Delta \boldsymbol{C}\| \\
&= \sum_{k=2}^{\infty} \frac{1}{k!} \sum_{i=0}^{k-2} (k-1-i)(\|\boldsymbol{A}\| + \|\boldsymbol{B}\|)^{k-2} \|\Delta \boldsymbol{C}\| \\
&= \left(\sum_{k=2}^{\infty} \frac{1}{k!} \left(\frac{k(k-1)}{2} \right) (\|\boldsymbol{A}\| + \|\boldsymbol{B}\|)^{k-2} \right) \|\Delta \boldsymbol{C}\| \\
&= \frac{1}{2} \sum_{k=2}^{\infty} \frac{(\|\boldsymbol{A}\| + \|\boldsymbol{B}\|)^{k-2}}{(k-2)!} \|\Delta \boldsymbol{C}\| \\
&= \frac{1}{2} \sum_{n=0}^{\infty} \frac{(\|\boldsymbol{A}\| + \|\boldsymbol{B}\|)^n}{n!} \|\Delta \boldsymbol{C}\| \qquad (\text{denote } n = k-2) \\
&= \frac{1}{2} \exp(\|\boldsymbol{A}\| + \|\boldsymbol{B}\|) \|\Delta \boldsymbol{C}\|.
\end{aligned}$$

□

Ruisheng Ran received the B.S. degree in mathematics from Chongqing Normal University, Chongqing, China, the M.S. degree in computational mathematics from University of Electronic Science and Technology of China, Chengdu, China, and the Ph.D. degree in computer application technology from the University of Electronic Science and Technology of China, Chengdu, China. He is currently as a post-doctor with College of Computer science, Chongqing University, and a Professor with the College of computer and information science, Chongqing Normal University, Chongqing, China. He is doing pattern recognition, image processing research.

Bin Fang received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, the M.S. degree in electrical engineering from Sichuan University, Chengdu, China, and the Ph.D. degree in electrical engineering from the University of Hong Kong, Hong Kong. He is currently a Professor with the Department of Computer Science, Chongqing University, Chongqing, China. His research interests include computer vision, pattern recognition, medical image processing, biometrics applications, and document analysis. He has published more than 100 technical papers and is an Associate Editor of the International Journal of Pattern Recognition and Artificial Intelligence.

Xuegang Wu received the B.S. degree in computer science from Shenyang University, Liaoning, China, in 1998, Ph.D. degree in the College of Computer Science of Chongqing University, Chongqing, China in 2014. And now, as a post-doctor in College of Communication Engineering, Chongqing University, he is doing pattern recognition, image processing and computer network research.