A General Exponential Framework for Dimensionality Reduction

Su-Jing Wang, *Member, IEEE,* Shuicheng Yan, *Senior Member, IEEE,* Jian Yang, *Member, IEEE,* Chun-Guang Zhou and Xiaolan Fu, *Member, IEEE,*

***Abstract*—As a general framework, Laplacian embedding, based on a pairwise similarity matrix, infers low dimensional representations from high dimensional data. However, it generally suffers from three issues: 1) algorithmic performance is sensitive to the size of neighbors, 2) the algorithm encounters the well-known small sample size (SSS) problem, and 3) the algorithm de-emphasizes small distance pairs. To address these issues, here we propose Exponential Embedding using matrix exponential and provide a general framework for dimensionality reduction. In the framework, the matrix exponential can be roughly interpreted by the random walk over the feature similarity matrix, and thus is more robust. The positive definite property of matrix exponential deals with the SSS problem. The behavior of the decay function of Exponential Embedding is more significant in emphasizing small distance pairs. Under this framework, we apply matrix exponential to extend many popular Laplacian embedding algorithms, *e.g.*, Locality Preserving Projections, Unsupervised Discriminant Projections and Marginal Fisher Analysis. Experiments conducted on the synthesized data, UCI, and the Georgia Tech face databases show that the proposed new framework can well address the issues mentioned above.**

***Index Terms*—Face recognition, Manifold Learning, Matrix exponential, Laplacian embedding, Dimensionality reduction.**

I. Introduction

Real data, such as face images or fMRI scans are usually depicted in high dimensions. In order to handle high dimensional data, their dimensionality needs to be reduced. Dimensionality reduction is the transformation of high-dimensional data into a lower dimensional data space. Currently, the most extensively used dimensionality reduction methods are *subspace transformation*. Subspace transformation methods are appealing for two main reasons [1]. First, subspace transformation methods typically have a small number of parameters, and therefore can be estimated using relatively fewer samples. Subspace transformation methods are especially useful to model high-dimensional data, since learning models typically requires a large number of samples as a result of the curse-of-dimensionality. Second, many subspace transformation methods can be formulated as eigen-problems, offering great potential for efficient learning of linear and nonlinear models without local minima.

Principal Component Analysis (PCA) [2] is an extensively used linear subspace transformation method maximizing the variance of the transformed features in the projected subspace. Linear Discriminant Analysis (LDA) [3] encodes discriminant information by maximizing the between-class covariance, and meanwhile minimizing the within-class covariance in the projected subspace. Another important subspace method is the Bayesian algorithm using probabilistic subspace [4]. Wang *et al.* [5] modeled face difference with three components and used them to unify PCA, LDA and Bayesian into a general framework.

Among the three algorithms, LDA suffers from the *Small Sample Size* (SSS) problem. This stems from generalized eigen-problems with singular matrices. To tackle the SSS problem, many variants of LDA have been proposed in the recent years, such as Fisherface [3], null LDA [6], LDA/QR [7], LDA/GSVD [8], LDA/FKT [9], DLA [10][11], Direct LDA and its variants [12][13][6][14]. An *et al.* [15] unified these LDA variants in one framework: principal component analysis plus constrained ridge regression.

However, both PCA and LDA fail to discover the underlying manifold structure, in which the high dimensional image information in the real world lies. In order to uncover the essential manifold structure of the facial images, *laplacianfaces* [16] were obtained by using Locality Preserving Projections (LPP) [17] to preserve the locality of image samples. LPP is well-known as a Laplacian embedding algorithm. When transforming the samples into the projected subspace, it tries to preserve the local structure of the samples, i.e., the neighbor relationship between the samples [18][19] so that samples that were originally in close proximity in the original space remain so in the projected subspace. Torre [1] unified PCA, LDA, Canonical Correlation Analysis (CCA) [20], LPP, Spectral Clustering (SC) [21], and their kernel as well as regularized extensions into least-squares weighted kernel reduced rank regression. LPP and its variations only characterize the locality

This work was supported in part by grants from 973 Program (2011CB302201), the National Natural Science Foundation of China (61379095, 61175023), China Postdoctoral Science Foundation funded project (2012M520428) and the open project program (93K172013K04) of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University. Jian Yang was partially supported by the National Science Fund for Distinguished Young Scholars under Grant Nos. 61125305.

S.J Wang is with the State Key Laboratory of Brain and Cognitive Science, Institute of Psychology, Chinese Academy of Sciences, Beijing, 100101, China and also with the College of Computer Science and Technology, Jilin University, Changchun 130012, China. (e-mail:wangsujing@psych.ac.cn).

S.C Yan is now with the Department of Electrical and Computer Engineering at National University of Singapore, Singapore. (e-mail: eleyans@nus.edu.sg).

J Yang is now with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China. (e-mail: csjyang@mail.njust.edu.cn).

C.G Zhou is with the College of Computer Science and Technology, Jilin University, Changchun 130012, China. (e-mail:cgzhou@jlu.edu.cn).

X Fu is with the State Key Laboratory of Brain and Cognitive Science, Institute of Psychology, Chinese Academy of Sciences, Beijing, 100101, China. (e-mail:fuxl@psych.ac.cn).

of samples, so they do not guarantee a good projection for classification purposes. To address this, Unsupervised Discriminant Projections (UDP) [22] introduces the concept of nonlocality and characterizes the nonlocality of samples by using the nonlocal scatter. A concise criterion for feature extraction can be obtained by maximizing the ratio of nonlocal scatter to local scatter. Most of the above existing algorithms were unified into a general graph embedding framework proposed by Yan *et al.* [23]. And a new supervised dimensionality reduction algorithm Marginal Fisher Analysis (MFA) was proposed by them under this framework as well.

In the graph embedding framework, the neighbor relationship is measured by the artificially constructed adjacent graph. The k nearest neighbors and $\epsilon-$lneighborhood criteria are the two most popular adjacent graph construction manners. $\epsilon-$lneighborhood is geometrically intuitive but infeasible because it is hard to choose a proper neighborhood radius ϵ in practice. So k nearest neighbors is always used instead in real applications.

Once an adjacent graph is constructed, the edge weights are assigned by various strategies such as 0-1 weights and heat kernel function. Unfortunately, since the adjacent graph is artificially constructed beforehand, it does not necessarily disclose the intrinsic locality of the samples. The algorithmic performance is often sensitive to the parameter k and the performance may vary each time k changes [24][25]. Worse yet, even if k and the sample number are fixed, the performance would still fluctuate with each new set of random samples. To overcome the problem, several researchers began to investigate on how to construct the adjacent graph. Yang et al. [26] constructed Sample-dependent Graph based on samples in question to determine neighbors of each sample and similarities between sample pairs, instead of predefining the same neighbor parameter k for all samples. Zhao et al. [27] used label information to construct Locally Discriminating Projection (LDP). Qiao et al. [28] aimed to preserve the sparse reconstructive relationship of the samples, which was achieved by constructing the adjacent graph using a minimizing ℓ_1 regularization-related objective function.

Another common problem of these Laplacian Embedding algorithms (such as LPP, UDP and MFA) is that they suffer from the Small Sample Size (SSS) problem. The problem occurs when the feature dimensionality is greater than the number of samples, resulting in the singularity issue. To address the problem, *laplacianfaces* [16] uses PCA to reduce the dimension, and then applies LPP. However, a potential problem is that the PCA criterion may not be compatible with the LPP criterion, thus the PCA step may discard the valuable information for LPP. In order to address the issue, some strategies [6][12] to deal with the singular problem of LDA are used on LPP [29][30][31], but they are ineffective on LPP because the influence of the null space of $\mathbf{S}_w$ on LDA differs from that of the null space of $\mathbf{S}_D$ on LPP. Finally, Laplacian Embedding algorithms de-emphasize small distance pairs, leading to many violations of local topology preserving at small distance pairs [32].

To address the above issues, we propose a general exponential framework for dimensionality reduction, motivated by the work in [33]. In [33], Zhang *et al.* proposed exponential discriminant analysis (EDA), using matrix exponential to deal with the SSS problem of LDA. In the proposed framework, the matrix exponential can be considered as the cumulative sum of the similarity/transition matrices after the random walk over the feature similarity matrix. The random walk makes the feature similarity matrix more reliable and suppresses the sensitivity to the size of neighbors. The fact that the matrix exponential is non-singular well deals with the SSS problem. The framework uses Exponential Embedding and the relation $\prod$ to replace Laplacian Embedding and the relation $\sum$, respectively. This remedies the defect that Laplacian Embedding de-emphasizes small distance pairs. Under this new framework, we use matrix exponential to extend LPP, UDP and MFA algorithms.

The rest of this paper is organized as follows: in Section II, we briefly review Laplacian Embedding algorithms and show that they suffer from the SSS problem; in Section III, we give the background of matrix exponential, introduce the Exponential Embedding, and provide a general framework for dimensionality reduction; in Section IV, experiments are conducted on the synthesized data, UCI, and the well-known face databases to validate that the proposed new framework can well address the three issues mentioned above; finally in Section V, conclusions are drawn.

II. Laplacian Embedding: A Review

Given a matrix of N samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N], \mathbf{x}_i \in \mathbb{R}^D$, Laplacian Embedding searchs for a transformation matrix $\mathbf{W} \in \mathbb{R}^{D\times d}$ to obtain: $\mathbf{y}_i = \mathbf{W}^T\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^d$, such that $\mathbf{y}_i$ in the projected subspace represents the desirable properties of $\mathbf{x}_i$. In order to characterize the locality of samples, an adjacency graph G with N nodes is constructed often by k nearest neighbors. If node i is among the k nearest neighbors of node j or node j is among the k nearest neighbors of node i, an edge is put to connect nodes i and j. According to the adjacency graph G, a similarity matrix $\mathbf{H} \in \mathbb{R}^{N\times N}$ is defined by the following two ways:

1) 0-1 function

$$H_{ij} = \begin{cases} 1 & \text{nodes } i \text{ and } j \text{ are connected in } G \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

2) Heat kernel function

$$H_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{2t^2}} & \text{nodes } i \text{ and } j \text{ are connected in } G \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Here t is a parameter that can be determined empirically. When t is large enough, $exp(-\|x_i - x_j\|^2/t) = 1$, heat kernel becomes 0-1 ways. Obviously, 0-1 ways is a special case of the heat kernel. The similarity matrix $\mathbf{H}$ is the basic foundation of characterizing the locality of samples in many manifold learning algorithms.

A. Three Popular Laplacian Embedding Algorithms

Here, we will give a brief introduction of Locality Preserving Projections, Unsupervised Discriminant Projections and Marginal Fisher Analysis.

1) Locality Preserving Projections: LPP [17] is a classical Laplacian embedding approach. The similarity matrix $\mathbf{H}$, without using class label information, is used to characterize the locality of samples in unsupervised LPP. The objective function is designed to enforce that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, then $\mathbf{y}_i$ and $\mathbf{y}_j$ are close as well. The desired transformation matrix $\mathbf{W}$ is obtained by minimizing the following objective:

$$\begin{aligned}&\frac{1}{2}\sum_{i,j}\|\mathbf{y}_i-\mathbf{y}_j\|^2H_{ij}=\frac{1}{2}\sum_{i,j}\|\mathbf{W}^T\mathbf{x}_i-\mathbf{W}^T\mathbf{x}_j\|^2H_{ij}\\&=tr\left(\mathbf{W}^T\mathbf{X}(\mathbf{D}-\mathbf{H})\mathbf{X}^T\mathbf{W}\right)=tr\left(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w}\right),\end{aligned}\tag{3}$$

where $\mathbf{L}=\mathbf{D}-\mathbf{H}$ is the Laplacian matrix. $\mathbf{D}$ is a diagonal matrix and its entry $D_{ii}=\sum_j H_{ij}$ measures the local density around $\mathbf{x}_i$. The bigger the value D_{ii} is, the more important $\mathbf{y}_i$ will be. Therefore, we impose a constraint as follows:

$$\mathbf{Y}^T\mathbf{D}\mathbf{Y}=\mathbf{I}\Rightarrow\mathbf{W}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{W}=\mathbf{I}.\tag{4}$$

For convenience, we denote

$$\mathbf{S}_L=\mathbf{X}\mathbf{L}\mathbf{X}^T\qquad\text{and}\qquad\mathbf{S}_D=\mathbf{X}\mathbf{D}\mathbf{X}^T.\tag{5}$$

So, the criterion function of LPP is as follows:

$$\min_{\mathbf{W}}tr\left(\left(\mathbf{W}^T\mathbf{S}_D\mathbf{W}\right)^{-1}\left(\mathbf{W}^T\mathbf{S}_L\mathbf{W}\right)\right).\tag{6}$$

Finally, the transformation matrix $\mathbf{W}$ consists of the eigenvectors associated with the smallest eigenvalues of the following generalized eigenvalue decomposition problem:

$$\mathbf{S}_L\mathbf{w}_i=\lambda_i\mathbf{S}_D\mathbf{w}_i.\tag{7}$$

2) Unsupervised Discriminant Projections: LPP only characterizes the locality of samples. Based on LPP, UDP [22] also characterizes the nonlocality of samples by using the nonlocal scatter. A concise criterion for feature extraction can be obtained by maximizing the ratio of nonlocal scatter to local scatter. The local scatter matrix is defined by

$$\mathbf{S}_L=\frac{1}{2}\sum_{i,j}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^TH_{ij}.\tag{8}$$

Similarly, the nonlocal scatter matrix can be defined by

$$\mathbf{S}_N=\frac{1}{2}\sum_{i,j}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^T(1-H_{ij}).\tag{9}$$

UDP then optimizes:

$$\max_{\mathbf{W}}tr\left(\left(\mathbf{W}^T\mathbf{S}_L\mathbf{W}\right)^{-1}\left(\mathbf{W}^T\mathbf{S}_N\mathbf{W}\right)\right).\tag{10}$$

3) Marginal Fisher Analysis: Differing from LPP and UDP, MFA uses the class label information to construct two graphs based on k nearest neighbors: an intrinsic graph that characterizes the intraclass compactness and a penalty graph which characterizes the interclass separability. The intrinsic graph illustrates the intraclass point adjacency relationship, where each sample is connected to its k_1-nearest neighbors of the same class. The corresponding similarity matrix is denoted as $\mathbf{H}^c$.

$$H_{ij}^c=\begin{cases}1 & \text{if } i\in N_{k_1}^+(j) \text{ or } j\in N_{k_1}^+(i);\\ 0 & \text{otherwise;}\end{cases}\tag{11}$$

where $N_{k_1}^+(i)$ indicates the index set of the k_1 nearest neighbors of the sample $\mathbf{x}_i$ in the same class.

The penalty graph illustrates the interclass marginal point adjacency relationship and the marginal point pairs of different classes are connected. The corresponding similarity matrix is denoted as $\mathbf{H}^p$.

$$H_{ij}^p=\begin{cases}1 & \text{if } (i,j)\in P_{k_2}(c_i) \text{ or } (i,j)\in P_{k_2}(c_j);\\ 0 & \text{otherwise;}\end{cases}\tag{12}$$

where $P_{k_2}(c)$ is a set of data pairs that represent the k_2 nearest pairs among the set $\{(i,j), i\in\pi_c, j\notin\pi_c\}$. π_c denotes a set of the elements belonging to cth class. Intraclass compactness is characterized as follows:

$$\mathbf{S}_c=\frac{1}{2}\sum_{ij}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^TH_{ij}^c,\tag{13}$$

where $\mathbf{L}^c=\mathbf{D}^c-\mathbf{H}^c$ is the Laplacian matrix from the intrinsic graph. Similarly, the interclass separability is characterized by

$$\mathbf{S}_p=\frac{1}{2}\sum_{ij}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^TH_{ij}^p,\tag{14}$$

where $\mathbf{L}^p=\mathbf{D}^p-\mathbf{H}^p$ is the Laplacian matrix from the penalty graph.

MFA tries to find a transformation matrix which will make intraclass more compact while simultaneously making interclass more separable. The criterion function of MFA is as follows:

$$\min_{\mathbf{W}}tr\left(\left(\mathbf{W}^T\mathbf{S}_p\mathbf{W}\right)^{-1}\left(\mathbf{W}^T\mathbf{S}_c\mathbf{W}\right)\right).\tag{15}$$

B. Small Sample Size Problem

Generally, the number of training samples is always less than their dimensionality. This results in the consequence that LPP, UDP and MFA suffer from the SSS problem. We first investigate the SSS problem for UDP. Due to the symmetry of $\mathbf{H}$, Eq. (8) can be rewritten:

$$\begin{aligned}\mathbf{S}_L&=\frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N(H_{ij}\mathbf{x}_i\mathbf{x}_i^T+H_{ij}\mathbf{x}_j\mathbf{x}_j^T-2H_{ij}\mathbf{x}_i\mathbf{x}_j^T)\\&=\sum_{i=1}^N D_{ii}\mathbf{x}_i\mathbf{x}_i^T-\sum_{i=1}^N\sum_{j=1}^N H_{ij}\mathbf{x}_i\mathbf{x}_j^T\\&=\mathbf{X}\mathbf{D}\mathbf{X}^T-\mathbf{X}\mathbf{H}\mathbf{X}^T\\&=\mathbf{X}\mathbf{L}\mathbf{X}^T\end{aligned}\tag{16}$$

Theorem 1. *Let D and N be the dimensionality of the sample and the number of the samples, respectively. If $D>N$, then the rank of $\mathbf{S}_L$ is at most $N-1$.*

Proof: According to the definition of the Laplacian matrix and the fact that the similarity matrix is symmetrical,

$$|\mathbf{L}|=\begin{vmatrix}\sum_j H_{1j}-H_{11} & -H_{12} & \cdots & -H_{1N}\\ -H_{12} & \sum_j H_{2j}-H_{22} & \cdots & -H_{2N}\\ \vdots & \vdots & \ddots & \vdots\\ -H_{1N} & -H_{2N} & \cdots & \sum_j H_{Nj}-H_{NN}\end{vmatrix}\tag{17}$$

we add the 2nd, 3rd,... Nth rows into the 1st row, and then obtain $|\mathbf{L}| = 0$. So, the rank of $\mathbf{L}$ 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. Hence, $rank(\mathbf{S}_L) = rank(\mathbf{X}\mathbf{L}\mathbf{X}^T) \leq N-1$. ■

From Theorem 1, when SSS problem occurs, $\mathbf{S}_L$ is singular. Eq. (10) cannot be solved. So, UDP suffers from SSS problem. Using similar proof of Theorem 1, we can obtain $rank(\mathbf{S}_D) \leq N$ with SSS problem. LPP also suffers from SSS problem. So does MFA.

III. A General Exponential Framework for Dimensionality Reduction

A. Matrix Exponential

Mathematically, matrix exponential is a matrix function on square matrices analogous to the ordinary exponential function. Due to the fact that it has many desirable properties, the matrix exponential is widely used in applications such as nuclear magnetic resonance spectroscopy [34][35], control theory [36], and Markov chain analysis [37].

Definition 1. *Given an $n \times n$ square matrix $\mathbf{X}$, its exponential is denoted as $e^{\mathbf{X}}$ or $\exp(\mathbf{X})$, and it is defined as follows:*

$$e^{\mathbf{X}} = \mathbf{I} + \mathbf{X} + \frac{\mathbf{X}^2}{2!} + \cdots + \frac{\mathbf{X}^m}{m!} + \cdots \tag{18}$$

where $\mathbf{I}$ is an identity matrix with the size of $n \times n$.

The properties of matrix exponential are listed as follows:

1) $e^{\mathbf{0}} = \mathbf{I}$.
2) $e^{a\mathbf{X}}e^{b\mathbf{X}} = e^{(a+b)\mathbf{X}}$.
3) $e^{\mathbf{X}}e^{-\mathbf{X}} = \mathbf{I}$.
4) If $\mathbf{X}\mathbf{Y} = \mathbf{Y}\mathbf{X}$, then $e^{\mathbf{X}+\mathbf{Y}} = e^{\mathbf{X}}e^{\mathbf{Y}} = e^{\mathbf{Y}}e^{\mathbf{X}}$.
5) If $\mathbf{X}$ is a diagonal matrix, i.e. $\mathbf{X} = diag(x_1, x_2, \ldots, x_n)$, then its exponential can be obtained by just exponentiating every entry on the main diagonal: $e^{\mathbf{X}} = diag(e^{x_1}, e^{x_2}, \ldots, e^{x_n})$
6) If $\mathbf{Y}$ is an invertible matrix, then $e^{\mathbf{Y}^{-1}\mathbf{X}\mathbf{Y}} = \mathbf{Y}^{-1}e^{\mathbf{X}}\mathbf{Y}$.
7) $|e^{\mathbf{X}}| = e^{tr(\mathbf{X})}$.
8) $e^{(\mathbf{X}^T)} = (e^{\mathbf{X}})^T$. It holds that if $\mathbf{X}$ is symmetric then $e^{\mathbf{X}}$ is also symmetric, and that if $\mathbf{X}$ is skew-symmetric then $e^{\mathbf{X}}$ is orthogonal.

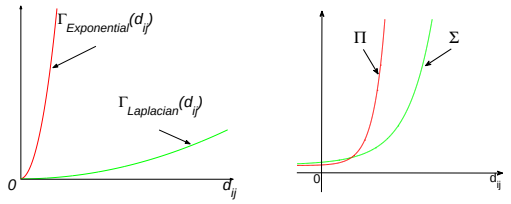
A wide variety of methods for computing $\exp(\mathbf{A})$ were analyzed in the classic paper of Moler and Van Loan [38], which was reprinted with an update in [39]. The scaling and squaring method is one of the best methods for computing the matrix exponential. For details, please refer to [39].

B. Exponential Embedding and General Framework

Observing LPP, UDP and MFA, we can derive their core functions as the following Laplacian embedding:

$$\arg\min_{\mathbf{w}} tr\left(\sum_{ij}(\mathbf{y}_i - \mathbf{y}_j)(\mathbf{y}_i - \mathbf{y}_j)^T H_{ij}\right), \tag{19}$$

where H_{ij} represents H_{ij} in LPP and UDP, but H^p_{ij} and H^c_{ij} in MFA.



(a) (b)

Fig. 1. The different behaviors of the decay functions and their relations. (a) The behaviors of the decay functions for Laplacian embedding and Exponential embedding. Abscissa denotes d_{ij} and ordinate denotes the values of the decay functions. For ease of comparison, the curve of the decay function of Exponential Embedding has been phase shifted to start from the origin. (b) The behaviors of the relations for $\sum$ and $\prod$. The red line denotes the function $\prod_{ij}\Gamma_{Laplacian}(d_{ij})$ and the green line denotes the function $\sum_{ij}\Gamma_{Laplacian}(d_{ij})$.

Without loss of generality, we assume that the dimensionality of projected subspace $d = 1$. If we define the distance between y_i and y_j as $d_{ij} = |y_i - y_j|$ and the decay function as $\Gamma(d_{ij})$, the general embedding objective can be written as follows:

$$\arg\min_{\mathbf{w}} \mathcal{R}_{ij}\Gamma(d_{ij})H_{ij}. \tag{20}$$

The objective consists of two parts: 1) the decay function $\Gamma(d_{ij})$, and 2) the relation $\mathcal{R}$, such as summation $\sum$ or product $\prod$. For example, when the decay function is $\Gamma_{Laplace}(d_{ij}) = d^2_{ij}$ and the relation $\mathcal{R}$ is summation $\sum$, Eq. (20) is then the Laplacian embedding in Eq. (19):

$$\arg\min_{\mathbf{w}} \sum_{ij}\Gamma_{Laplacian}(d_{ij})H_{ij}. \tag{21}$$

Figure 1(a) depicts the behavior of the decay function of Laplacian embedding. Laplacian embedding uses small distance pairs (namely, d_{ij} is small) to preserve the locality of samples. In other words, Laplacian embedding only takes into account the small distance pairs, because when d_{ij} is large, the corresponding H_{ij} is zero, and then $\Gamma_{Laplacian}(d_{ij})H_{ij}$ has no contribution to Eq. (21). However, when d_{ij} is small, the behavior of $\Gamma_{Laplacian}(d_{ij})$ is less steep. So Laplacian Embedding cannot characterize the locality of samples well. To address the problem, we define the decay function of Exponential embedding as $\Gamma_{Exponential}(d_{ij}) = e^{d^2_{ij}/\sigma^2}$. As shown in Figure 1(a), when d_{ij} is small, the behavior of $\Gamma_{Exponential}(d_{ij})$ is steeper, so it can characterize the locality of samples better than $\Gamma_{Laplacian}(d_{ij})$. Thus, we replace the $\Gamma_{Laplacian}(d_{ij})$ in Eq. (21) with $\Gamma_{Exponential}(d_{ij})$,

$$\arg\min_{\mathbf{w}} \sum_{ij} e^{\frac{(y_i-y_j)^2}{\sigma}}H_{ij}. \tag{22}$$

In this work, we simply set $\sigma = 1$. Because the value of H_{ij} is either 1 or 0 in our implementations, the solution of the following problem is equal to the solution of Eq. (22) plus a constant:

$$\arg\min_{\mathbf{w}} \sum_{ij} e^{(y_i-y_j)^2 H_{ij}}. \tag{23}$$

Due to $(y_i - y_j)^2 H_{ij} \geq 0$, we obtain $e^{(y_i-y_j)^2 H_{ij}} \geq 1$. This means that each term in Eq. (23) is equal to or greater than one. Figure 1(b) illustrates the behaviors of two relations $\sum$ and $\prod$. As shown in Figure 1(b), when the value of the decay function is greater than or equal to 1, the behavior of product $\prod$ is more significant than summation $\sum$. To further enhance the significance, we change the relation $\mathcal{R}$ in Eq.(23) from summation $\sum$ to product $\prod$:

$$\arg\min_{\mathbf{w}} \prod_{ij} e^{(y_i-y_j)^2 H_{ij}} = \arg\min_{\mathbf{w}} e^{\sum_{ij}(y_i-y_j)^2 H_{ij}}. \quad (24)$$

When $d > 1$, Eq. (24) can be replaced as:

$$\begin{aligned} &\arg\min_{\mathbf{W}} \left| \prod_{ij} e^{(\mathbf{y}_i-\mathbf{y}_j)(\mathbf{y}_i-\mathbf{y}_j)^T H_{ij}} \right| \\ &= \arg\min_{\mathbf{W}} \left| e^{\sum_{ij}(\mathbf{y}_i-\mathbf{y}_j)(\mathbf{y}_i-\mathbf{y}_j)^T H_{ij}} \right| \\ &= \arg\min_{\mathbf{W}} \left| e^{\mathbf{W}^T \mathbf{X}\mathbf{L}\mathbf{X}^T \mathbf{W}} \right| = \arg\min_{\mathbf{W}} e^{tr(\mathbf{W}^T \mathbf{S}_L \mathbf{W})} \end{aligned} \quad (25)$$

where $|\cdot|$ denotes the matrix determinant. Obviously, $\mathbf{W} = \mathbf{0}$ is the solution of Eq. (25). However, $\mathbf{W} = \mathbf{0}$ does not make sense for dimensionality reduction. So, the constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ is needed. From the monotonicity of the exponential function, Eq. (25) acquires the minimum, if and only if $tr\left(\mathbf{W}^T\mathbf{S}_L\mathbf{W}\right)$ obtains the minimum. The minimum of $tr\left(\mathbf{W}^T\mathbf{S}_L\mathbf{W}\right)$ can be obtained by solving the following eigenvalue problem:

$$\mathbf{S}_L\mathbf{w} = \lambda\mathbf{w}. \quad (26)$$

Theorem 2. *If $\mu_1, \mu_2, \ldots, \mu_n$ are eigenvectors of $\mathbf{X}$ that correspond to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, then $\mu_1, \mu_2, \ldots, \mu_n$ are also eigenvectors of $e^{\mathbf{X}}$ that correspond to the eigenvalues $e^{\lambda_1}, e^{\lambda_2}, \ldots, e^{\lambda_n}$.*

Proof: μ_i is the eigenvector of $\mathbf{X}$ that corresponds to the eigenvalue λ_i, i.e. $\mathbf{X}\mu_i = \lambda_i\mu_i$, because

$$\begin{aligned} &\mathbf{I}\mu_i = \mu_i \\ &\mathbf{X}\mu_i = \lambda_i\mu_i \\ &\frac{1}{2!}\mathbf{X}^2\mu_i = \frac{1}{2!}\lambda_i\mathbf{X}\mu_i = \frac{1}{2!}\lambda_i^2\mu_i \\ &\frac{1}{3!}\mathbf{X}^3\mu_i = \frac{1}{3!}\lambda_i\mathbf{X}^2\mu_i = \frac{1}{3!}\lambda_i^3\mu_i \\ &\ldots \\ &\frac{1}{m!}\mathbf{X}^m\mu_i = \frac{1}{m!}\lambda_i\mathbf{X}^{m-1}\mu_i = \frac{1}{m!}\lambda_i^m\mu_i \\ &\ldots \end{aligned} \quad (27)$$

Summing the above equations, we have

$$\begin{aligned} &(\mathbf{I} + \mathbf{X} + \frac{1}{2!}\mathbf{X}^2 + \ldots + \frac{1}{m!}\mathbf{X}^m + \ldots)\mu_i \\ &= (1 + \lambda_i + \frac{\lambda_i^2}{2!} + \ldots + \frac{\lambda_i^m}{m!} + \ldots)\mu_i. \end{aligned} \quad (28)$$

According to the definition of matrix exponential and the definition of power series of scaler λ_i: $e^{\lambda_i} = 1 + \lambda_i + \frac{\lambda_i^2}{2!} + \ldots + \frac{\lambda_i^m}{m!} + \ldots$, Eq. (28) is rewritten:

$$e^{\mathbf{X}}\mu_i = e^{\lambda_i}\mu_i. \quad (29)$$

This means that μ_i is the eigenvector of $e^{\mathbf{X}}$ that corresponds to the eigenvalue e^{λ_i}. ■

According to Theorem 2, $\mathbf{S}_L$ has the same eigenvectors as $e^{\mathbf{S}_L}$. So, Eq. (25) can be obtained by solving the following eigenvalue problem:

$$e^{\mathbf{S}_L}\mathbf{w} = \lambda\mathbf{w}. \quad (30)$$

Similarly, the constraint $\mathbf{W}^T e^{\mathbf{S}_D}\mathbf{W} = \mathbf{I}$ is imposed. The criterion function of Exponential LPP (ELPP) can be written as follows:

$$\arg\min_{\mathbf{W}} \left| \left(\mathbf{W}^T e^{\mathbf{S}_D}\mathbf{W}\right)^{-1} \left(\mathbf{W}^T e^{\mathbf{S}_L}\mathbf{W}\right) \right|. \quad (31)$$

It can be solved by the following generalized eigenvalue decomposition method:

$$e^{\mathbf{S}_L}\mathbf{w}_i = \lambda_i e^{\mathbf{S}_D}\mathbf{w}_i. \quad (32)$$

Due to the fact that $e^{\mathbf{S}_L}$ and $e^{\mathbf{S}_D}$ are positive definite, Eq. (32) has D positive eigenvalues. The solution of ELPP consists of the eigenvectors corresponding to the d ($1 \leq d \leq D$) smallest eigenvalues.

Observing LPP, UDP and MFA, the criterion functions of many dimensionality reduction algorithms can be summarized as follows,

$$\arg\max_{\mathbf{W}} \text{or} \min_{\mathbf{W}} tr\left(\left(\mathbf{W}^T\mathbf{S}_2\mathbf{W}\right)^{-1}\left(\mathbf{W}^T\mathbf{S}_1\mathbf{W}\right)\right) \quad (33)$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ differ according to different algorithms. For example, when $\mathbf{S}_1 = \mathbf{S}_L$ and $\mathbf{S}_2 = \mathbf{S}_D$, a minimization of Eq. (33) will represent the criterion function of LPP.

Following Theorem 2, $\mathbf{X}$ and $e^{\mathbf{X}}$ share the same eigenvectors. We can use $e^{\mathbf{S}_1}$ and $e^{\mathbf{S}_2}$ to replace $\mathbf{S}_1$ and $\mathbf{S}_2$ in Eq. (33) and propose a General Exponential Framework that solves the SSS problem as follows:

$$\arg\max_{\mathbf{W}} \text{or} \min_{\mathbf{W}} \left| \left(\mathbf{W}^T e^{\mathbf{S}_2}\mathbf{W}\right)^{-1} \left(\mathbf{W}^T e^{\mathbf{S}_1}\mathbf{W}\right) \right|. \quad (34)$$

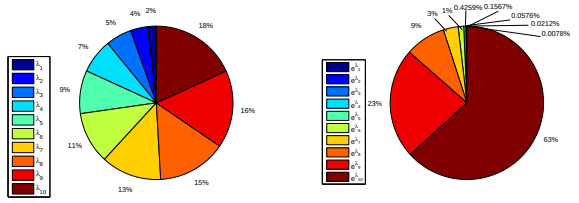
For specific algorithms, we can obtain the criterion functions of Exponential LPP (ELPP), Exponential UDP (EUDP) and Exponential MFA (EMFA) as follows:

1) ELPP: $\arg\min_{\mathbf{W}} \left| \left(\mathbf{W}^T e^{\mathbf{S}_D}\mathbf{W}\right)^{-1} \left(\mathbf{W}^T e^{\mathbf{S}_L}\mathbf{W}\right) \right|$;
2) EUDP: $\arg\min_{\mathbf{W}} \left| \left(\mathbf{W}^T e^{\mathbf{S}_N}\mathbf{W}\right)^{-1} \left(\mathbf{W}^T e^{\mathbf{S}_L}\mathbf{W}\right) \right|$;
3) EMFA: $\arg\min_{\mathbf{W}} \left| \left(\mathbf{W}^T e^{\mathbf{S}_p}\mathbf{W}\right)^{-1} \left(\mathbf{W}^T e^{\mathbf{S}_c}\mathbf{W}\right) \right|$.

Similarly, when $\mathbf{S}_1 = \mathbf{S}_b$ and $\mathbf{S}_2 = \mathbf{S}_w$, a maximization of Eq.(34) will represent the criterion function of EDA. So EDA can also be unified into the framework.

C. Justifications

From another point of view, we may treat each feature f_p ($p = 1, 2, \ldots, D$) as a vertex in the vertex set $F = \{f_1, f_2, \ldots, f_D\}$ of a graph. $\mathbf{S}$ ($\mathbf{S}$ denotes $\mathbf{S}_1$ or $\mathbf{S_2}$) can be considered as the special feature *similarity matrix*, which represents the certain relationship among the D features. Especially when each feature dimension is ℓ_2-normalized, the similarity matrix is the special affine cosine similarity matrix, and we may *virtually* and *roughly* consider this matrix as a



(a) Proportion of each λ_i to $\sum \lambda_i$ (b) Proportion of each e^{λ_i} to $\sum e^{\lambda_i}$

Fig. 2. The proportions of λ_i and e^{λ_i}.

transition matrix over the graph $G_{rw} = \{F, \mathbf{S}\}$[1]. Due to the limited number of training samples and the sensitivity to k, the similarity/transition matrix $\mathbf{S}$ is often not reliable. To get a more reliable similarity/transition matrix, we may argue the matrix in two aspects. First we may implement the *random walk* on the graph G_{rw}. The $\mathbf{S}^m$ can be considered as a new similarity/transition matrix after $(m-1)$-step random walk on the graph to consider the global correlations among features, and the resultant $\mathbf{S}^m$ is a smoothed similarity matrix. We set different weights $\frac{1}{m!}$ for the similarity matrices $\mathbf{S}^m$; the more smoothed the similarity matrix is, the fewer weights there will be. Then, we may use the prior that features are prone to independent, and then the corresponding similarity matrix is an identity matrix of $\mathbf{I}$. To balance the prior matrix of $\mathbf{I}$ and the similarity matrix $\mathbf{S}$, we may add a weight λ to $\mathbf{S}$. Then by combining all these matrices with different weights, we obtain $\sum_{m=0}^{\infty} \frac{(\lambda \mathbf{S})^m}{m!}$, which is right to be $e^{\lambda \mathbf{S}}$. By properly setting the scale of $\mathbf{S}$, we may set λ=1, and then we have $e^{\mathbf{S}}$. Obviously, $e^{\mathbf{S}}$ encodes richer information for characterizing the similarity of features than $\mathbf{S}$.

LPP, UDP, MFA and other methods where the k-nearest neighbor method is applied are sensitive to changes of parameter k. When k is different, the set $\{\lambda_1, \lambda_2, \ldots, \lambda_D\}$ generated by Eq. (7) is different. This is also the reason why the performance of LPP is sensitive to the size of neighbors k. Through matrix exponential, λ_i is changed to e^{λ_i}. Due to the nature of exponentials, the bigger the eigenvalue is, the larger its proportion will be, and the smaller the eigenvalue is, the smaller its proportion will be. Figure 2 illustrates the proportion of each λ_i and e^{λ_i} to the sum of all eigenvalues. The largest eigenvalue λ_{10} accounts for 18% of the total proportion in Figure 2(a), while its corresponding exponential $e^{\lambda_{10}}$ accounts for 63% of the total in Figure 2(b). The smallest eigenvalue λ_1 accounts for 2% in Figure 2(a), while its corresponding exponential e^{λ_1} accounts for 0.0078% in Figure 2(b). As shown in Figure 2, through matrix exponential, noises (over small eigenvalues) caused by fluctuation in parameter k have been reduced, namely because eigenvectors for small eigenvalues are reduced. In other words, eigenvectors for large eigenvalues are magnified.

Generally, for real high dimensional data, such as face images, the number of dimensions is greater than the number of training samples. This is the well-known small sample size (SSS) problem. However, there is a possibility of $\mathbf{S}_2$ in Eq. (33) becoming a singular matrix. A common strategy for SSS problem is to reduce dimensionality via PCA before using the corresponding criterion functions. However, a potential problem is that the PCA criterion may not be compatible with the subsequent criterion functions, and the PCA step may discard valuable information for these algorithms in the null space of $\mathbf{S}_2$. To address this problem, let us look back at Theorem 2, where any eigenvalue e^{λ_i} of $e^{\mathbf{X}}$ should be larger than zero. This ensures that $e^{\mathbf{X}}$ is non-singular. In addition, the eigenvectors of $\mathbf{X}$ and $e^{\mathbf{X}}$ are the same, and their corresponding eigenvalues share an exponential relationship that is strongly monotonic and unaffected by the order of eigenvalues. The relevant matrices are replaced with their corresponding exponentials into the criterion functions. This ensures $e^{\mathbf{S}_1}$ and $e^{\mathbf{S}_2}$ are also non-singular matrices. Therefore, when the SSS problem occurs, the Eq. (34) works well. Moreover, valuable information in the null space $\mathbf{S}_2$ is kept.

[1]Here, S_{ij} can be negative, and thus the explanation is just intuitive but not fully theoretically sound.

IV. Experiments

A. Experiments on Synthesized Data

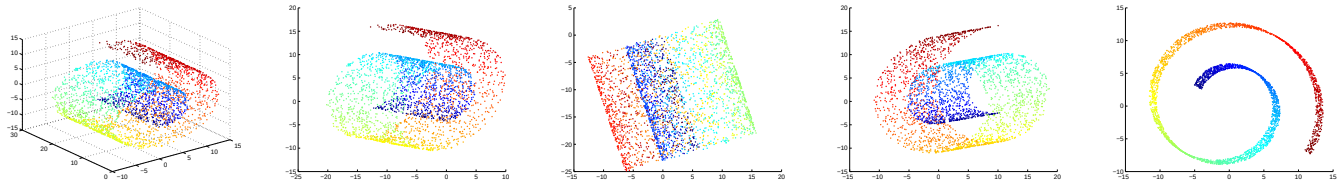
Let us first consider two well-known synthetic data sets from [40]: the s-curve and the Swiss roll. Figure 3 illustrates the 3,000 randomly sampled 3D points on the Swiss roll manifold and their respective 2D projections obtained by LPP, ELPP, UDP and EUDP. The size of nearest neighbors k is set as 12. It can be observed that the performance of LPP parallels that of UDP, and the performance of ELPP is better than that of LPP. Among the four methods, EUDP obtains the best performance, because it preserves global geometric characteristics and faithfully projects and conveys the information about how the manifold is folded in the high dimensional space.

Figure 4 illustrates the sensitivity of ELPP and EUDP with respect to random realizations of the data set for different k, respectively. We tested with several values of k ($k = 5, 10, 15, 20, 25$) and the results are illustrated in Figure 4. Notice that when the parameter k changes, the 2D projections obtained by ELPP only rotate, but the distribution of projections remains the same. This means that the performance of ELPP is less sensitive to k. For EUDP, its performance is more sensitive to k in comparison to ELPP. However, it can still reveal the intrinsic manifold structure of the Swiss roll in many cases ($k = 10, 15, 25$). In other cases ($k = 5, 20$), it unfolds and separates the samples well.

With the same experimental setting, Fig. 5 illustrates the 3,000 randomly sampled 3D points on the s-curve manifolds and the 2D projections obtained by the four methods in the s-curve data set. From the figure, it can also be seen that the distribution of LPP is similar to that of UDP and the projection of EUDP is the most separable in the four projections. In summary, EUDP has the best performance on the synthesized data.

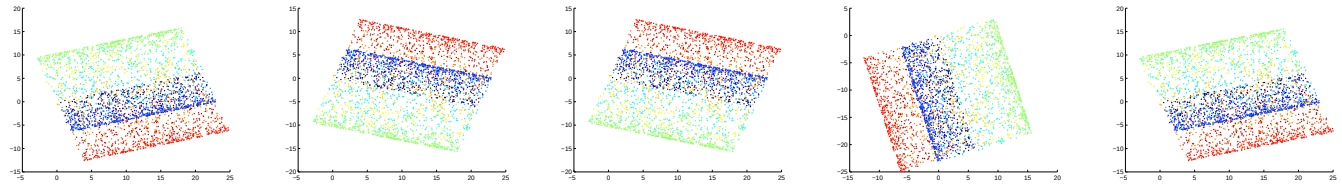
B. Experiments on the UCI Machine Learning Repository

We evaluate the recognition accuracy of ELPP, EUDP and EMFA on the Landsat Satellite data set from the UCI Machine

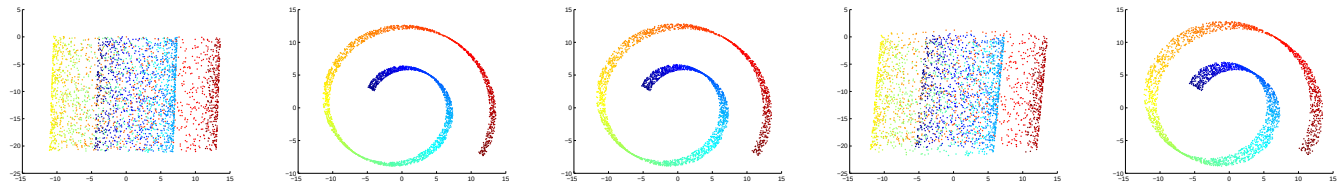


(a) swiss roll (b) LPP (c) ELPP (d) UDP (e) EUDP

Fig. 3. Results of four related methods applied to the Swiss roll example.

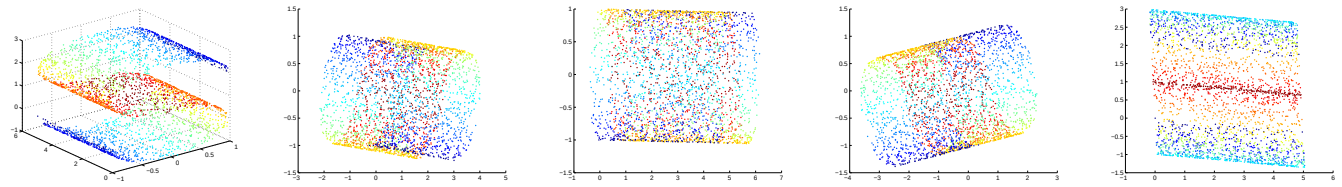


(a) ELPP k=5 (b) ELPP k=10 (c) ELPP k=15 (d) ELPP k=20 (e) ELPP k=25



(f) EUDP k=5 (g) EUDP k=10 (h) EUDP k=15 (i) EUDP k=20 (j) EUDP k=25

Fig. 4. Behavior of ELPP and EUDP under different values of k on the Swiss roll data set.



(a) s-curve (b) LPP (c) ELPP (d) UDP (e) EUDP

Fig. 5. Results of four related methods applied to the s-curve example.

Learning Repository [2]. The data set consists of 6,435 measurements with 36 attributes from six classes. We compare ELPP, EUDP and EMFA against their non-exponential versions. We randomly choose p ($p = 100, 200, 300, 400, 500$) samples from each class as the training set and the rest are used as the testing set. This process is repeated 20 times. We search for k from $\{25, 50, 75, \ldots, \lfloor \frac{N-1}{25} \rfloor \times 25\}$ for ELPP, EUDP and their corresponding non-exponential versions. As for MFA and EMFA, we set k_1 for the intrinsic graph, searching from $\{25, 50, 75, \ldots, \lfloor \frac{p}{25} \rfloor \times 25\}$ and k_2 for the penalty graph searching from $\{25, 50, 75, \ldots, \lfloor \frac{N-p}{25} \rfloor \times 25\}$. We then choose the maximal recognition accuracy as the result for each process. Note that for final classification, we used the Nearest Neighbor method. Finally, we calculate the mean value of the 20 maximal values. The results are listed in Table I, which shows that the average recognition accuracies of ELPP, EUDP and EMFA are better than those of their corresponding non-exponential versions in most cases. For EMFA, we also compare it with two discriminant dimensionality reduction algorithms: LDA and EDA. Out of four discriminant algorithms, EMFA obtains the best performance and EDA obtains the second best performance. This also shows that the general exponential framework is effective.

TABLE I
AVERAGE RECOGNITION ACCURACY ON LANDSAT SATELLITE

p	100	200	300	400	500
ELPP	**0.8508**	**0.8668**	**0.8744**	**0.8788**	**0.8854**
LPP	0.8100	0.8372	0.8531	0.8608	0.8701
EUDP	**0.8469**	**0.8626**	**0.8706**	0.8760	0.8831
UDP	0.8230	0.8564	0.8702	**0.8784**	**0.8855**
EMFA	**0.8595**	**0.8745**	**0.8836**	**0.8880**	**0.8941**
MFA	0.8157	0.8413	0.8568	0.8677	0.8781
EDA	0.8511	0.8643	0.8724	0.8758	0.8809
LDA	0.8021	0.8236	0.8338	0.8369	0.8439

[2]http://archive.ics.uci.edu/ml/

Fig. 6. Sample images of one individual from the Georgia Tech database.

C. Experiments on the Georgia Tech face database

Georgia Tech face database[3] contains images of 50 individuals taken in two or three sessions at different times. Each individual in the database is represented by 15 color images with cluttered background taken at a resolution of 640×480 pixels. The pictures show frontal and/or tilted faces with different facial expressions, lighting conditions and scales. Each image is manually cropped and resized to 32×32 pixels. The sample images for one individual of the Georgia Tech database are shown in Fig. 6. In order to avoid the matrix exponential approaching infinite, all images are normalized to $[0, 1]$.

In this experiment, the similarity matrix $\mathbf{H}$ is defined by the 0-1 function. The neighbors parameter k is searched from $\{2, 3, \ldots, N-1\}$. We randomly split the image samples so that p ($p = 2, 4, 6, 8, 10, 12$) images for each individual are used as the training set and the rest are used as the testing set. This process is repeated 20 times. Fig. 7 shows that the performances of LPP and ELPP vs. the neighborhood size k. Abscissa denotes the repeated time and ordinate denotes the neighborhood size k. The color of the patch denotes the recognition accuracy. The warmer the color is, the higher the recognition accuracy is. The difference between the patch colors may show the algorithmic sensitivity to k. The greater the difference is, the higher the sensitivity is. Comparing the corresponding columns of Fig. 7(a) and Fig. 7(b), there is very little color difference in each column of Fig. 7(a). This means that the performance of ELPP is much less sensitive to the parameter k than that of LPP.

Analytically, we define the criterion to measure the sensitivity to the parameter k as follows. The recognition accuracies are normalized to $[0, 1]$. Within the 20 random splits in our experiments, each split includes $N-1$ recognition accuracy corresponding to $N-1$ values of k. For each split, the maximum difference of recognition accuracy is obtained by subtracting the minimum accuracy from the maximum accuracy. The criterion Mean Maximum Difference (MMD) is the mean value of all maximum differences of recognition accuracy. To a certain degree, the smaller the value of MMD is, the more insensitive to k will be. The MMDs of ELPP and LPP are listed in Table II. From the table, the MMDs of ELPP are less than those of LPP. This also shows that ELPP is less sensitive to k than LPP.

[3]http://www.anefian.com/research/face_reco.htm

TABLE II
MMD OF ELPP AND LPP ON GEORGIA TECH DATABASE

p	2	4	6	8	10	12
ELPP	0.1855	0.2065	0.2393	0.3600	0.2687	0.4187
LPP	0.4632	0.6021	0.6806	0.6056	0.6620	0.6056

We compare the cost time of ELPP with LPP using various image sizes. The images are resized to 8×8, 16×16, 32×32, 64×64 and 128×128 pixels. All images are used as the training samples. The cost time is listed in Table III. From the table, it can be seen that the cost time of ELPP is smaller than that of LPP when the size of the image is greater than or equal to 32×32 pixels. In LPP, the step of PCA becomes more time consuming as image size increases. Hence, the cost time of LPP increases dramatically with the increasing image size.

TABLE III
THE COST TIME OF ELPP AND LPP (SECOND)

$size$	8×8	16×16	32×32	64×64	128×128
ELPP	3.0140	3.1090	3.4988	5.6480	32.3166
LPP	0.0121	0.2083	6.4669	9.7797	35.5853

In order to investigate the performance of ELPP, we compare ELPP with LPP. The results are illustrated in Fig. 8(a). The solid lines denote that the neighborhood size k is searched from $\{2, 3, \ldots, N-1\}$. The dot-dash lines denote k is set as 2. As shown in Fig. 8(a), the performances of ELPP are better than those of LPP in two ranges of k. This outcome stems from the PCA step before implementing LPP, where the information that might be valuable to LPP is discarded. Nevertheless, the proposed algorithm does not employ the PCA step, preserving all possible valuable information that gives ELPP an edge over LPP. In Fig. 8(a), we also find that the space between two curves of ELPP is much narrower than that of LPP. This also proves that the performance of ELPP is much less sensitive to the parameter k than that of LPP.

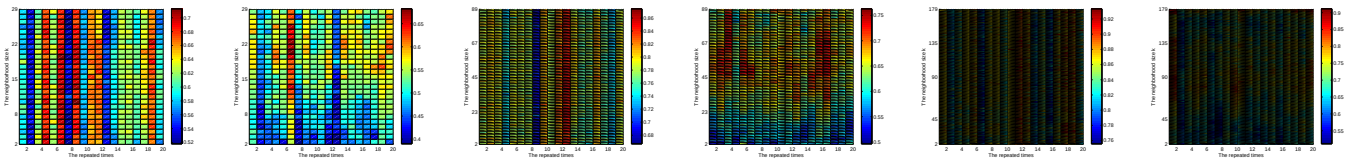
Intuitively, we project all 225 samples in Georgia Tech database to the first two axes by LPP and ELPP. The projections are illustrated in Fig. 9. From Fig. 9(b), we can see that 255 samples are projected into 4 points by LPP. This may cause difficulties for subsequent classifications. As mentioned in Section II-A1,

$$\frac{1}{2}\sum_{i,j}(\mathbf{y}_i - \mathbf{y}_j)^2 H_{ij} = \mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w}. \tag{35}$$

Consequently, if the eigenvector corresponding to a zero eigenvalue is taken as the transforming axis, the transformed result will statistically satisfy the following condition:

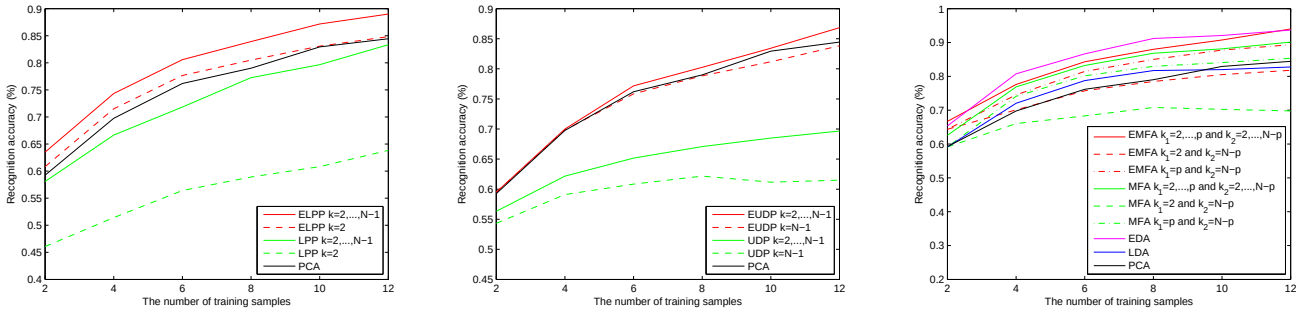
$$\frac{1}{2}\sum_{i,j}(\mathbf{y}_i - \mathbf{y}_j)^2 H_{ij} = 0. \tag{36}$$

Note that $H_{ij} \geq 0$ and $(\mathbf{y}_i - \mathbf{y}_j)^2 \geq 0$ are certain for arbitrary i and j. Hence, Eq. (36) means that for arbitrary



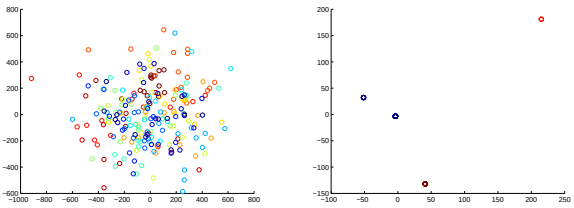
(a) ELPP (p=2) (b) LPP (p=2) (c) ELPP (p=6) (d) LPP (p=6) (e) ELPP (p=12) (f) LPP (p=12)

Fig. 7. The performances of LPP and ELPP vs. the neighborhood size k on the the Georgia Tech database.



(a) ELPP and LPP (b) EUDP and UDP (c) EMFA and MFA

Fig. 8. The performances of on the Georgia Tech database.



(a) ELPP (b) LPP

Fig. 9. The sample distribution on the first two axes using the LPP and ELPP on the Georgia Tech database.

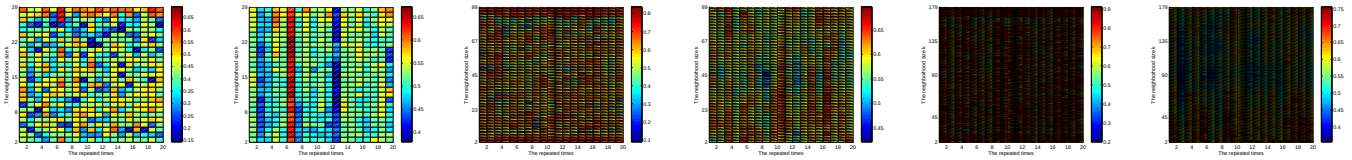
i and j, $(\mathbf{y}_i - \mathbf{y}_j)^2 H_{ij} = 0$ should be satisfied. In particular, for two neighbor samples $H_{ij} > 0$ is satisfied. As a result, $(\mathbf{y}_i - \mathbf{y}_j)^2 H_{ij} = 0$ implies that in the transform space the neighbor samples must have the same representation. This does not preserve the local structure of data as in LPP, which requires that the transformed results of neighbor samples be in close proximity rather than being the same. Therefore, 255 samples are projected into four points only on the first two axes of LPP. This cannot reveal the intrinsic manifold structure of samples. However, ELPP replaces $\mathbf{XLX}^T$ with $e^{\mathbf{XLX}^T}$ to ensure that the arbitrary eigenvalue is larger than zero. This prevents neighbor samples from being projected into the same point.

In the same way, we also compare EUDP with UDP. Fig. 10 shows that the performances of UDP and EUDP vs. the neighborhood size k. Differing from Fig. 7, there is less color difference in each row of the figure of EUDP. This means that when the neighborhood size k is fixed, the performance of the proposed algorithm is less sensitive to changes of the randomly sampled training images than that of UDP. From Fig. 10, we can also see that when k is very small or very large, EUDP obtains better performance. When k is almost $N-1$, k is less sensitive to the performance of EUDP than that of UDP. And when p is fixed, the performances of EUDP are almost identical in 20 random processes. The performances of EUDP and UDP are illustrated in Fig. 8(b). The dot-dash lines denote that k is equal to $N-1$. As is shown in Fig. 8(b), the performances of EUDP with $k = N-1$ are better than those of UDP with $k = 1, 2, \ldots, N-1$. Moreover, we also find that the space between two curves of EUDP is much narrower than that of UDP.

We also evaluate the performances of EMFA and MFA on the Georgia Tech database. The neighbors parameter k_1 for the intrinsic graph is searched from $\{2, 3, \ldots, p\}$. The neighbors parameter k_2 for the penalty graph is searched from $\{2, 3, \ldots, N-p\}$. Fig. 11 shows that the performances of EMFA and MFA vs. the neighborhood size k_2 (k_1 is fixed). From the figures, we can see that when the neighbors parameter k_2 for the penalty graph is large enough, the performance of EMFA is insensitive to the neighbors parameters. We also compare EMFA with MFA, LDA and EDA. The results are illustrated in Fig. 8(c). Although the performances of EDA are better than those of EMFA in most cases, the performance of EMFA is better than that of EDA in the case where p=2. This shows EMFA is more efficient than EDA, when the number of training samples is rather small.

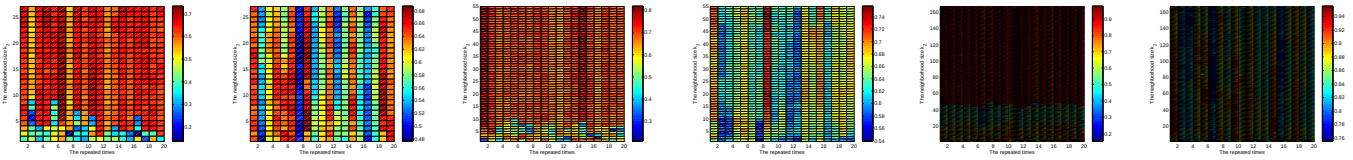
V. Conclusion

We have presented Exponential Embedding and a general framework for dimensionality reduction. Under the framework, we used matrix exponential to extend LPP, UDP and MFA algorithms. These exponential versions can deal with 1) small sample size (SSS) problem, 2) the algorithmic sensitivity to the size of neighbors k and 3) de-emphasizing small distance pairs. The experiments on the synthesized data, UCI and



(a) EUDP (p=2) (b) UDP (p=2) (c) EUDP (p=6) (d) UDP (p=6) (e) EUDP (p=12) (f) UDP (p=12)

Fig. 10. The performances of UDP and EUDP vs. the neighborhood size k on the Georgia Tech database.



(a) EMFA (p=2,k_1=2) (b) MFA (p=2,k_1=2) (c) EMFA (p=4,k_1=2) (d) MFA (p=4,k_1=2) (e) EMFA (p=12,k_1=12) (f) MFA (p=12,k_1=12)

Fig. 11. The performances of EMFA and MFA vs. the neighborhood size k_2 on the the Georgia Tech database.

the Georgia Tech face databases revealed that the proposed framework can well address above problems.

ACKNOWLEDGMENT

The authors would like to thank Xiao-Hua Liu, Shuchao Pang, Quanhong Fu and Yu-Hsin Chen for their valuable comments.

REFERENCES

[1] F. De la Torre, "A least-squares framework for component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1041–1055, 2011.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[4] B. Moghaddam, "Principal manifolds and probabilistic subspaces for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 780–788, 2002.

[5] X. Wang and X. Tang, "A unified framework for subspace face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1222–1228, 2004.

[6] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.

[7] J. Ye and Q. Li, "A two-stage linear discriminant analysis via QR-decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 929–941, 2005.

[8] P. Howland and H. Park, "Generalizing discriminant analysis using the generalized singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 995–1006, 2004.

[9] S. Zhang and T. Sim, "Discriminant subspace analysis: A Fukunaga-Koontz approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1732–1745, 2007.

[10] T. Zhang, D. Tao, and J. Yang, "Discriminative locality alignment," *Computer Vision–ECCV 2008*, vol. 5320, pp. 725–738, 2008.

[11] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1299–1313, 2009.

[12] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data-with application to face recognition," *Pattern Recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.

[13] J. Yang and J. Yang, "Why can LDA be performed in PCA transformed space?" *Pattern recognition*, vol. 36, no. 2, pp. 563–566, 2003.

[14] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "Discriminative common vectors for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 4–13, 2005.

[15] S. An, W. Liu, S. Venkatesh, and H. Yan, "Unified formulation of linear discriminant analysis methods and optimal parameter selection," *Pattern Recognition*, vol. 44, no. 2, pp. 307 – 319, 2011.

[16] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[17] X. He and P. Niyogi, "Locality preserving projections," in *Advances in neural information processing systems*, vol. 16. The MIT Press, 2004, pp. 153–160.

[18] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in neural information processing systems*, vol. 1, pp. 585–592, 2002.

[19] X. He, D. Cai, and W. Min, "Statistical and computational analysis of locality preserving projection," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 281–288.

[20] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[21] B. Mohar, *Some applications of Laplace eigenvalues of graphs*. Springer, 1997.

[22] J. Yang, D. Zhang, J. Yang, and B. Niu, "Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 650–664, 2007.

[23] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[24] M. Balasubramanian and E. Schwartz, "The isomap algorithm and topological stability," *Science*, vol. 295, no. 5552, pp. 7–7, 2002.

[25] R. Wang, S. Shan, X. Chen, J. Chen, and W. Gao, "Maximal linear embedding for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1776–1792, 2011.

[26] B. Yang and S. Chen, "Sample-dependent graph construction with application to dimensionality reduction," *Neurocomputing*, vol. 74, no. 1-3, pp. 301–314, 2010.

[27] H. Zhao, S. Sun, Z. Jing, and J. Yang, "Local structure based supervised feature extraction," *Pattern Recognition*, vol. 39, no. 8, pp. 1546–1550, 2006.

[28] L. Qiao, S. Chen, and X. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognition*, vol. 43, no. 1, pp. 331–341, 2010.

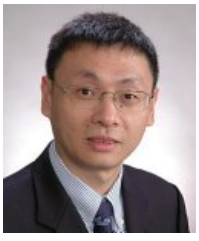
[29] G. Feng, D. Hu, and Z. Zhou, "A Direct Locality Preserving Projections (DLPP) Algorithm for Image Recognition," *Neural Processing Letters*, vol. 27, no. 3, pp. 247–255, 2008.

[30] J. Chen, B. Li, and B. Yuan, "Face recognition using direct LPP algorithm," in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*. IEEE, 2008, pp. 1457–1460.

[31] R. Li, Z. Luo, and G. Han, "Pseudo-inverse Locality Preserving Pro-

jections," in *Computational Intelligence and Security, 2009. CIS'09. International Conference on*, vol. 1. IEEE, 2010, pp. 363–367.
[32] D. Luo, C. Ding, F. Nie, and H. Huang, "Cauchy graph embedding," *the Proceedings of the 28th International Conference on Machine Learning*, pp. 553–560, 2011.
[33] T. Zhang, B. Fang, Y. Y. Tang, Z. Shang, and B. Xu, "Generalized discriminant analysis: A matrix exponential approach," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 40, no. 1, pp. 186–197, 2010.
[34] T. Havel, I. Najfeld, and J. Yang, "Matrix decompositions of two-dimensional nuclear magnetic resonance spectra," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 91, no. 17, pp. 7962–7966, 1994.
[35] I. Najfeld and T. Havel, "Derivatives of the matrix exponential and their computation," *Advances in Applied Mathematics*, vol. 16, no. 3, pp. 321–375, 1995.
[36] G. Franklin, D. Powell, and M. Workman, "Digital control of dynamic systems," 1997.
[37] R. Sidje and W. Stewart, "A numerical study of large sparse matrix exponentials arising in markov chains," *Computational statistics & data analysis*, vol. 29, no. 3, pp. 345–368, 1999.
[38] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM review*, vol. 20, no. 4, pp. 801–836, 1978.
[39] ——, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM review*, vol. 45, no. 1, pp. 3–49, 2003.
[40] L. Saul and S. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *The Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

Su-Jing Wang (M'12) received the Master's degree from the Software College of Jilin University, Changchun, China, in 2007. He received the Ph.D. degree from the College of Computer Science and Technology of Jilin University in 2012. He is a postdoctoral researcher in Institute of Psychology, Chinese Academy of Sciences. He has published more than 30 scientific papers. He is One of Ten S-electees of the Doctoral Consortium at International Joint Conference on Biometrics 2011. He was called as *Chinese Hawkin* by the Xinhua News Agency. His research was published in IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, Neurocomputing, etc. His current research interests include pattern recognition, computer vision and machine learning. He also reviews for several top journals, such as IEEE Transactions on Pattern Analysis and Machine Intelligence and IEEE Transactions on Neural Networks and Learning Systems. For details, please refer to his homepage http://sujingwang.name.



Shuicheng Yan (M'06-SM'09) is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, and the Founding Lead of the Learning and Vision Research Group. Dr. Yan's research areas include computer vision, multimedia and machine learning, and he has authored or coauthored over 300 technical papers over a wide range of research topics, with Google Scholar citation ¿9400 times and H-index-42. He is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and ACM TRANSACTIONS ON INTELLIGENT SYSTEMS AND TECHNOLOGY, and has been serving as the Guest Editor of the special issues for TMM and CVIU. He received the Best Paper Awards from ACM MM in 2012 (demo), PCM in 2011, ACM MM in 2010, ICME in 2010, and ICIMCS in 2009, the winner prizes of the classification task in PASCAL VOC from 2010 to 2012, the winner prize of the segmentation task in PASCAL VOC in 2012, the Honourable Mention Prize of the detection task in PASCAL VOC in 2010, the 2010 TCSVT Best Associate Editor Award, the 2010 Young Faculty Research Award, the 2011 Singapore Young Scientist Award, and the 2012 NUS Young Researcher Award.

Jian Yang (M'08) received the BS degree in mathematics from the Xuzhou Normal University in 1995. He received the MS degree in applied mathematics from the Changsha Railway University in 1998 and the PhD degree from the Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a professor in the School of Computer Science and Technology of NUST. He is the author of more than 80 scientific papers in pattern recognition and computer vision. His journal papers have been cited more than 1800 times in the ISI Web of Science, and 4000 times in the Web of Scholar Google. His research interests include pattern recognition, computer vision and machine learning. Currently, he is an associate editor of Pattern Recognition Letters and IEEE Trans. Neural Networks and Learning Systems, respectively.

Chun-Guang Zhou PhD, professor, PhD supervisor, Dean of Institute of Computer Science of Jilin University. He is Jilin-province-management Expert, Highly Qualified Expert of Jilin Province, One-hundred Science-Technique elite of Changchun. And he is awarded the Governmental Subsidy from the State Department. He has many pluralities of national and international academic organizations. His research interests include related theories, models and algorithms of artificial neural networks, fuzzy systems and evolutionary computations, and applications of machine taste and smell, image manipulation, commercial intelligence, modern logistic, bioinformatics, and biometric identification based on computational intelligence. he has published over 168 papers in Journals and conferences and he published 1 academic book.

Xiaolan Fu (M'13) received her Ph. D. degree in 1990 from Institute of Psychology, Chinese Academy of Sciences. Currently, she is a Senior Researcher at Cognitive Psychology. Her research interests include visual and computational cognition: (1) attention and perception, (2) learning and memory, and (3) affective computing. At present, she is the director of Institute of Psychology, Chinese Academy of Sciences and Vice Director, State Key Laboratory of Brain and Cognitive Science.