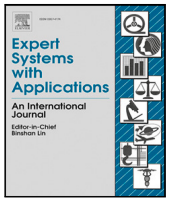
Expert Systems With Applications 242 (2024) 122750

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Locality Preserving Projections with Autoencoder

Ruisheng Ran [a,*], Ji Feng [a], Zheng Li [a], Jinping Wang [a], Bin Fang [b]

[a] College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China
[b] The College of Computer Science, Chongqing University, Chongqing 400044, China

ARTICLE INFO

Keywords:
Locality preserving projections
Autoencoder
Dimensionality reduction
Manifold learning

ABSTRACT

Locality Preserving Projections (LPP) is a popular dimensionality reduction method in the manifold learning field. However, LPP and all its variants only consider the one-way mapping from the high-dimensional space to the low-dimensional space and have no reverse verification, resulting in inaccurate low-dimensional embeddings. In this paper, we propose a new LPP method, called LPPAE (Locality Preserving Projections with Autoencoder), based on the linear Autoencoder. It constructs a two-way mapping: at the encoding stage, the conventional projection of LPP is viewed as a mapping from the high-dimensional space to the low-dimensional space. At the decoding stage, the low-dimensional embeddings are mapped back to the original high-dimensional space. The main contributions of the new method are: (1) This design not only preserves the neighborhood relationship of the data but more importantly, ensures that the low-dimensional embeddings can more accurately "represent" the original data, thus significantly improving the performance of LPP. Experimental results on Handwritten Alphadigits, COIL-20, Yale, AR datasets show that the recognition rates of LPPAE are 26.06, 10.09, 5.40, and 8.86% higher than those of the original LPP respectively. On the MNIST dataset, compared to some of the latest improvements of LPP, including LPPMDC, LAPP, LPP+TR, and DNLPP, the recognition rate of LPPAE has been improved by 12.50, 38.10, 9.10, and 2.61%, respectively. (2) LPPAE regards the conventional LPP as an encoder, which is a new perspective. The idea of LPPAE can be used as a general framework and then extended to other manifold learning methods, and then a series of new methods can be developed.

1. Introduction

Locality Preserving Projections (LPP) is a manifold-based dimensionality reduction method that was proposed by He et al. in 2003 (He & Niyogi, 2004), which is the linearization version of the well-known Laplacian Eigenmaps (LE) (Belkin & Niyogi, 2001). Based on the neighborhood graph constructed, LPP finds a linear mapping from the high-dimension space to a subspace and optimally preserves local neighborhood information. In comparison with manifold-based methods, classical principal component analysis (PCA) (Turk & Pentland, 1991) and linear discriminant analysis (LDA) (Fisher, 1936; Rao, 1948) are global structure-based dimensionality reduction methods.

In machine learning and computer vision, LPP has been paid much attention by researchers. A considerable amount of literature has been published on LPP in the past decade.

Since the number of training samples is often less than the dimensionality of the samples, LPP encounters difficulties in solving the generalized eigenvalue problem, which is called the small sample size (SSS) problem of LPP. To solve the SSS problem, many methods have been proposed. Regularization is a way to solve this problem. The oldest regularization method for LPP is regularization LPP (RLPP) (Cai, He and Han, 2007). Then the parametric regularized locality preserving projections method (PRLPP) has been proposed (Lu & Tan, 2009). The latest regularization method is the TSVD regularization method (LPP+TSVD) and the Tikhonov regularization method (LPP+TR) (Wei, Dai, & Liang, 2020). Matrix exponential was recently introduced into LPP to solve the SSS problem. For example, the exponential locality preserving projection (ELPP) method (Wang, Chen, Peng, & Zhou, 2011) and a general exponential framework for manifold learning (Wang, Yan, Yang, Zhou, & Fu, 2014). Inspired by the idea of matrix exponential, a more general matrix function is used to map the scatter matrices, and then a general matrix function dimensionality reduction framework is proposed in Ran, Feng, Zhang, and Fang (2020), and thus a function LPP (FLPP) method is presented, which also solves the SSS problem of LPP.

The classical LPP is based on the L2-norm distance criterion. However, this distance criterion is very sensitive to the outliers for the

* Corresponding author.
E-mail addresses: rshran@cqnu.edu.cn (R. Ran), fengji@cqnu.edu.cn (J. Feng), 2020210516075@stu.cqnu.edu.cn (Z. Li), 2021110516071@stu.cqnu.edu.cn (J. Wang), fb@cqu.edu.cn (B. Fang).

https://doi.org/10.1016/j.eswa.2023.122750
Received 17 May 2023; Received in revised form 8 November 2023; Accepted 25 November 2023
Available online 27 November 2023

Table 1
The review and analysis of the improvement of LPP.

Improved routes	Motivation	Sub-routes	Shortcoming
Solving the SSS problem	Since the number of the training samples is often less than the dimensionality of the samples, LPP encounters difficulties in solving the generalized eigenvalue problem.	(1) With regularization; (2) Matrix exponential	One-way mapping
Different distance metric	The classical LPP is based on the L2-norm distance, but it is sensitive to the outliers, so a different distance metric is made. criterion	(1) L2-norm distance metric; (2) L1-norm; (3) F-norm	One-way mapping
Discriminant LPP	LPP does not consider the class label of the sample, Discriminant Lpp considers the class label information	(1) For the SSS problem of DLPP; (2) With sparse representation; (3) With different distance metric	One-way mapping
Two-dimensional LPP	LPP is based on one-dimensional image vectors, which takes a lot of time, so a two-dimensional method is proposed.	(1) 2DLPP with regularization; (2) 2DLPP with sparse representation; (3) 2DLPP with different distance metric	One-way mapping
Other			One-way mapping

square operation. To ease the negative impact of outliers, many improvements of LPP based on the other distance criterion have been proposed. For example, the LPP-L1 method with L1-norm distance metric (Pang & Yuan, 2010), LPP with L1-norm minimization (Yu et al., 2018), the low-rank LPP model (Yin et al., 2021), the F-norm based models (F-LPP) (Hu, Sun, Gao, Hu, & Yin, 2018).

The LPP method is an unsupervised method that does not consider the class label of the sample. By considering the class information of the sample, discriminant locality preserving projections (DLPP) (Yu, Teng, & Liu, 2006) and many improvements have been presented. For example, exponential DLPP (EDLPP) (Lu, Wang, Zou, & Wang, 2018), DLPP based on the polynomial matrix function (Ran, Ren, Zhang, & Fang, 2021) are for the small-sample-size (SSS) problem of DLPP. With sparse representation, there is sparse locality preserving discriminative projections for face recognition (SLPDP) (Zhang, Wang, & Cai, 2017). With different distance criteria, there is the DLPP-L1 method (Wang, Gao, Xie, Gao, & Wang, 2016) the L2,1-DLPP method (Liu, Gao, Gao, & Shao, 2018), the DLPP-L2, p method (Du, Li, Wang, & Zhang, 2019).

The two-dimensional locality preserving projection (2DLPP) (Lu, Liu, & Liu, 2012) is also another extension way of LPP. These methods are all based on two-dimensional image matrices rather than one-dimensional image vectors. The 2DLPP method is extended in regularization, sparse representation, and different distance criteria, for example, robust 2DRLPP with regularization (2DRLPP) (Chen, Li, Shao, Zhang, & Deng, 2019), sparse 2DDLPP (S2DDLPP) (Shikkenawis & Mitra, 2015), 2DLPP based on the nuclear-norm (NN2DLPP) (Lu et al., 2017).

Recently, some improved methods of LPP have been proposed. Based on a deep neural network, a new LPP method called DNLPP is proposed (Long, Gao, Yang, Hu, & Yin, 2019). This method employs a suitable deep neural network to map the original sample layer and the low-dimensional embedding layer and has better classification ability than the conventional LPP. In addition, the locality adaptive preserving projections (LAPP) method (Wang et al., 2020) is proposed, which adaptively determines the neighbors and their relationships in the optimal subspace. Based on the approximate maximum difference criterion, the LPPMDC method is also proposed in Ran, Qin, Zhang, and Fang (2022).

Based on the above literature, we have analyzed the improvement of LPP in terms of the improved routes, motivations, improved sub-routes, and shortcomings, and then report them in Table 1.

The objective of these methods is to find the optimal linear mapping from the high-dimensional data space to the low-dimensional embedding space. This is achieved by adding conditions such as regularization, sparse representation, distance criterion, etc. to the original LPP while preserving the local neighborhood information of the original data. However, LPP and all its variants only consider a one-way mapping from the high-dimensional space to the low-dimensional space and do not reverse-verify the low-dimensional embeddings. This may lead to inaccurate low dimensional embeddings that cannot effectively represent the original high-dimensional data. This is a significant drawbacks of LPP and all its variants. In fact, many dimensionality reduction methods have this drawback, such as Isomap projection (IsoP) (Cai, He, Han and et al., 2007), neighborhood preserving embedding (NPE) (He, Cai, Yan, & Zhang, 2005), linear LTSA (LLTSA) (Zhang, Yang, Zhao, & Ge, 2007) etc.

Motivated to address the above drawbacks of LPP and all its variants, in this work, we present a new LPP method based on the encoder–decoder paradigm of Autoencoder, which is named LPPAE (LPP with Autoencoder). The innovation and superiority of LPPAE are:

LPP and its variants only consider one-way mapping from high-dimensional space to low-dimensional space, which may result in loss of information from the original data. To address this issue, we propose a new LPP method based on the encoder–decoder paradigm, named LPPAE (LPP with Autoencoder). With the linear autoencoder, LPPAE constructs a two-way mapping: at the encoding stage, the conventional projection of LPP is viewed as a mapping from the high-dimensional space to the low-dimensional space. At the decoding stage, the projected low-dimensional data is mapped or reconstructed back to the original high-dimensional data, which allows the model to reverse-verify the low-dimensional embeddings and make them more accurate.

The two-way mapping of LPPAE is designed to preserve the neighborhood relationship of the original data, which is obtained by the conventional mapping of LPP. Additionally, the resulting low-dimensional embedded data retains more information about the original data, and then more accurately represents it. Therefore, the obtained low-dimensional embeddings are more representative, which significantly improves the performance of LPP. This can be seen from the experimental results in Section 5. Furthermore, LPPAE can be used as a general framework and then extended to variants of LPP and other manifold learning methods. As examples, we improved two variant versions of LPP, namely Spectral LPP (Cai, He, Han, 2007) and Exponential LPP (ELPP) (Wang et al., 2011, 2014), as RLPPAE (RLPP with Autoencoder) and ELPPAE (ELPP with Autoencoder) in this paper. The main contribution of this paper is summarized as follows:

1. A novel LPP method based on the autoencoder named LPP with autoencoder (LPPAE) is proposed, which can obtain more accurate low-dimensional embedding by considering two-way mapping.
2. The concept of two-way mapping has been abstracted into a fundamental framework for data dimensionality reduction. This framework serves as a foundation for extending to various enhancements of LPP.
3. Comprehensive experiments are conducted to verify of the new method LPPAE, and the experimental results prove that the effectiveness of this method, which indicates that LPPAE is a significant improvement on the LPP method.

The rest of this paper proceeds as follows: in Section 2, we review the LPP method and autoencoder. In Section 3, we propose the novel LPP method with the encoder–decoder paradigm. In Section 4, as examples, we apply the proposed paradigm to some variants of LPP. In Section 5, we conduct experiments to evaluate the new methods. In Section 6, we make discussion about the study and finally conclude this paper in Section 7.

2. Related works

The topic of this study is about the improvement of LPP. Therefore, we will briefly review the LPP method in this section. Since LPP is a linearization version of Laplacian Eigenmaps (LE), we will first give a brief introduction to the LE method and then to LPP. In addition, since the structure of AutoEncoder will also be used, we will also briefly introduce Autoencoder and its development.

2.1. Locality preserving projections

LPP is based on Laplacian Eigenmaps (LE), so a brief introduction to LE is first presented. Suppose there are l original samples $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_l]$ in $\mathbb{R}^n$ space. For the LE method, an adjacency graph $\boldsymbol{G}$ is firstly constructed, which puts an edge between nodes i and j if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close". Usually, the k nearest neighborhood method or the ϵ -neighborhood method is used to construct the graph $\boldsymbol{G}$. Secondly, the weights S_{ij} of any two neighbors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in graph $\boldsymbol{G}$ are set. The heat kernel function $S_{ij} = exp(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2)/t$ is usually used to define the weight, where t is a parameter. And, a similarity matrix $\boldsymbol{S}$ is defined as $\boldsymbol{S} = (S_{ij})$. Thirdly, to make eigenmaps. LE is designed to embed the n-dimensional data point into the m-dimensional ($m \ll n$) data point $\boldsymbol{y}_i$ and simultaneously to make $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are "close" if the nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close", which is to minimize:

$$\frac{1}{2}\sum_{ij}\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 S_{ij} = tr\left(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T\right), \tag{1}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_1, \dots, \boldsymbol{y}_l]$, $\boldsymbol{L}$ is the Laplacian matrix, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$, and $\boldsymbol{D}$ is a diagonal matrix where $D_{ij} = \sum_j S_{ij}$. Usually, the constraint $\boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^T = \boldsymbol{I}$ is imposed on Eq. (1) to remove an arbitrary scaling factor in the embedding.

LE is a nonlinear dimensionality reduction method and it is only defined on the training dataset. The LPP method aims to seek a linear mapping:

$$\boldsymbol{y}_i = \boldsymbol{W}^T\boldsymbol{x}_i \tag{2}$$

to map the data in high-dimensional space into low-dimensional space. Considering the linear mapping (2) in Eq. (1), the objective function of LPP becomes:

$$\min_{\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{W}=\boldsymbol{I}} tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}), \tag{3}$$

where $\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{W} = \boldsymbol{I}$ is a constraint. The desired projection matrix $\boldsymbol{W}$ is composed of the eigenvectors corresponding to the first few minimum eigenvalues of the following generalized eigenvalue problem:

$$\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{w} = \lambda\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{w}. \tag{4}$$

Based on the design, LPP can be defined anywhere in ambient space, not just on the training dataset.

2.2. Autoencoder

The standard autoencoder is a two-layer fully connected neural network, including the input layer, hidden layer, and output layer. The input layer and the hidden layer constitute the encoder, and the hidden layer and the output layer constitute the decoder. The encoder encodes the input data into a new feature representation, and the decoder decodes the feature representation to obtain the reconstruction of the input data. The Autoencoder trains the weight parameters by minimizing the error between the reconstruction and the original input data to get the optimal feature representation of the input data. Autoencoder has been researched in recent years, and various autoencoders have been proposed one after another. These improvements include Stacked Autoencoder (Hinton & Salakhutdinov, 2006), Sparse Autoencoder (Bengio, Lamblin, Popovici, & Larochelle, 2006), Variational Autoencoder (Kingma & Welling, 2022), etc. For autoencoder, if the number of hidden layer nodes is less than the number of input layer nodes, it is called an undercomplete model; otherwise, it is called an overcomplete model. If the activation function of the hidden layer is linear, the Autoencoder is called a linear Autoencoder. In this paper, the proposed method is based on a linear undercomplete Autoencoder with only one hidden layer.

3. The locality preserving projections with autoencoder

With the idea of making the low-dimensional embeddings maintain the neighborhood relationship of the original high-dimensional data, LPP constructs a mapping from high-dimensional space to low-dimensional space to make the LE method work well. However, the mapping is one-way and does not reverse-verify the low-dimensional embeddings, which may lose some information about the original data and cannot represent the original data accurately. Other improvements of LPP focus on optimizing this one-way mapping from different approaches. In this paper, with the linear Autoencoder, the LPPAE method is proposed to address this problem. In this section, with the linear Autoencoder, the LPPAE method is proposed to address this problem. The framework of LPPAE is firstly given, then the objective function is constructed and the optimization method is performed. Finally, we describe how to apply the LPPAE model.

3.1. Framework

With the linear Autoencoder, the proposed LPPAE constructs a two-way mapping: at the encoding stage, the conventional projection of the LPP can be considered as a mapping from a high-dimensional space to a low-dimensional space, which has been designed to maintain the neighborhood relationships of the original data. At the decoding stage, the projected low-dimensional data is mapped or reconstructed back to the original high-dimensional data. The framework of the proposed LPPAE method is shown in Fig. 1.

Suppose there are l original samples $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_l]$ in $\mathbb{R}^n$ space, an adjacency graph $\boldsymbol{G}$ is firstly constructed.

The encoding stage is the conventional projection model of LPP. The model maps the high-dimension data point $\boldsymbol{x}_i$ to the low-dimensional data point $\boldsymbol{y}_i$ with the linear mapping $\boldsymbol{y}_i = \boldsymbol{W}^T\boldsymbol{x}_i$. From the perspective of a linear autoencoder, this mapping can be regarded as encoding each high-dimensional data point into a low-dimensional data point. And, by the design of LPP, the mapping simultaneously preserves local neighborhood information of the original samples, i.e., if the original samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close", then the embedded points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are also "close", which can be ensured with

$$min\frac{1}{2}\sum_{ij}\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 S_{ij}.$$

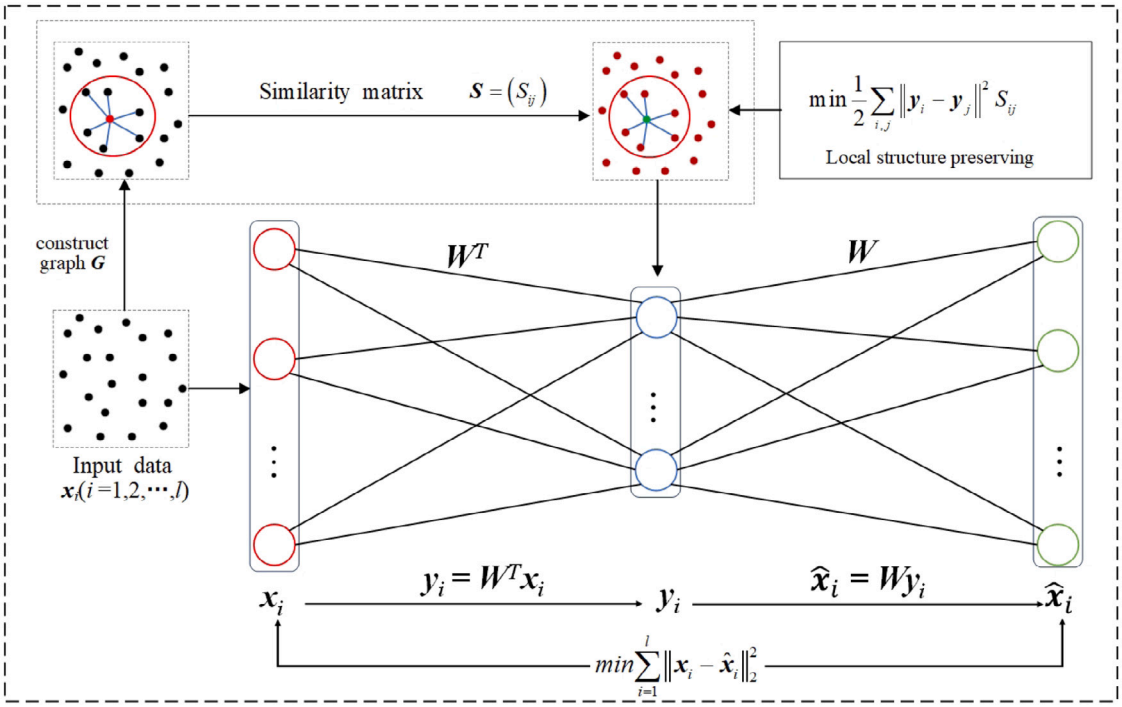


Fig. 1. The framework of the proposed LPPAE method.

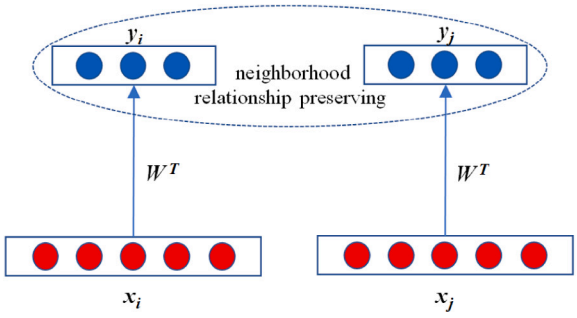


Fig. 2. The illustration of neighborhood preserving.

The idea of the low-dimensional embeddings maintaining the neighborhood relationship of the original data can be illustrated in Fig. 2:

The decoding stage is to decode the low-dimensional embedded point $\boldsymbol{y}_i$ to the original data with the linear autoencoder. Let $\hat{\boldsymbol{x}}_i$ be the reconstruction of the original data point $\boldsymbol{x}_i$, $\boldsymbol{W}^*$ be the weight matrix of the decoder. Mathematically, the decoding process may be formulated as:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{W}^* \boldsymbol{y}_i.$$

To simplify the model, the weights of the encoder and decoder in Autoencoder can be tied as introduced in Ref. Ranzato, Boureau, Cun, et al. (2007), i.e., $\boldsymbol{W}^* = (\boldsymbol{W}^T)^T = \boldsymbol{W}$. Thus, the decoding process may be rewritten as:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{W} \boldsymbol{y}_i.$$

And then, we also want to minimize the error between the input data $\boldsymbol{x}_i$ and its reconstruction $\hat{\boldsymbol{x}}_i$, i.e.,

$$min \sum_{i=1}^{l} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|_2^2.$$

so that the low-dimensional embedded point $\boldsymbol{y}_i$ "represents" the original sample $\boldsymbol{x}_i$ more accurately and effectively.

Thus, the new LPP method has two advantages. One is that it still preserves the neighborhood structure information of the original samples. The other is that the low-dimensional embedding data "represent" the original samples more accurately and effectively due to the additional reconstruction. Because the new LPP method is based on the encoder–decoder paradigm, it is named LPPAE (Locality Preserving Projections with Autoencoder).

3.2. The objective function

Following the framework, we present the objective function of the LPPAE method in this section. The first stage of LPPAE is the conventional projection model of LPP and preserves local neighborhood information of the samples. It can be formulated as minimizing Eq. (3), so it is used as the first item of the objective function of the new method, i.e.,

$$\mathcal{L}_{1st} = \frac{1}{2}\sum_{ij}\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 S_{ij} = tr\left(\boldsymbol{W}^T \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T \boldsymbol{W}\right). \tag{5}$$

And the constraint imposed on Eq. (3), i.e., $\boldsymbol{W}^T \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{W} = \boldsymbol{I}$ can be relaxed as:

$$\mathcal{L}_{2nd} = tr(\boldsymbol{W}^T \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{W}) - m. \tag{6}$$

Eq. (6) is used as the second item of the objective function of the new method. Where m is the dimensionality of the low-dimensional space, see Section 2.1.

The second stage of LPPAE is to reconstruct the original data $\boldsymbol{x}_i$ and minimize the error between the input data $\boldsymbol{x}_i$ and its reconstruction $\hat{\boldsymbol{x}}_i$. It can be formulated as:

$$\mathcal{L}_{3rd} = \sum_{i=1}^{l} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|_2^2.$$

It is used as the third item of the objective function of the new method.

Note that the linear mapping of LPP is $\boldsymbol{y}_i = \boldsymbol{W}^T \boldsymbol{x}_i$, so one has:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{W} \boldsymbol{y}_i = \boldsymbol{W} \boldsymbol{W}^T \boldsymbol{x}_i.$$

Thus, $\mathcal{L}_{3rd}$ can be rewritten as:

$$\begin{aligned}\mathcal{L}_{3rd} &= \sum_{i=1}^{l}\left\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_i\right\|_2^2 \\ &= \sum_{i=1}^{l}\left\|\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)\boldsymbol{x}_i\right\|_2^2 \\ &= tr\left(\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)^T\right).\end{aligned} \tag{7}$$

Combine Eqs. (5), (6), and (7), the new method is to find such a projection matrix W and minimize the following objective function:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{1st} + \lambda\mathcal{L}_{2nd} + \gamma\mathcal{L}_{3rd} \\ &= tr\left(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}\right) + \\ &\lambda\left(tr\left(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{W}\right) - m\right) + \\ &\gamma tr\left(\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)^T\right),\end{aligned} \tag{8}$$

where λ and γ are the balance parameters, which reflect the importance of the corresponding item.

It is worth mentioning that $\|\boldsymbol{W}\|_F^2$ regularization has not been considered in our model. It is unnecessary because the weights of the encoder and decoder are tied, i.e., $\boldsymbol{W}^* = \boldsymbol{W}$. If the norm $\|\boldsymbol{W}\|_F^2$ is large, the low-dimensional projection produced by the encoder will have large values; and then, in the decoding stage, after the low-dimensional projection is multiplied by the matrix $\boldsymbol{W}$, bad reconstruction will be produced. That is, the $\|\boldsymbol{W}\|_F^2$ regularization has been automatically handled by the reconstruction constraints (Ranzato et al., 2007).

The proposed LPPAE method has two free parameters: λ and γ. If $\gamma = 0$, Eq. (8) is reduced to:

$$\mathcal{L} = tr\left(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}\right) + \lambda\left(tr\left(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{W}\right) - m\right) \tag{9}$$

Eq. (9) is the expression of applying the Lagrange multiplier method to Eq. (3) with the constraint of Eq. (4). At this point, Eq. (8) is the objective function of the LPP method.

In Eq. (8), the first item $\mathcal{L}_{1st}$ can be regarded as a local structure information preserving item, and its coefficient can be regarded as 1. The third item $\mathcal{L}_{3rd}$ can be regarded as the reconstruction item and the parameter γ reflects the proportion of reconstruction. If $0 < \gamma < 1$, the proportion of the reconstruction item is less than that of the local structure information preserving item. If $\gamma \geq 1$, the proportion of the reconstruction item is equal to or greater than that of the local structure information preserving item.

3.3. Optimization

Unlike the LPP method, the objective function of the proposed LPPAE method, Eq. (8), is nonlinear, so it is difficult to solve directly. However, we may use the gradient descent method to solve Eq. (8). In order to speed up the learning process of the gradient descent method, we can use the Stochastic Gradient Descent with Nesterov momentum to find the optimal projection matrix $\boldsymbol{W}$, the parameter λ and γ. This method is also called Nesterov Accelerated Gradient (NAG) method (Nesterov, 1983). Note that the objective function of LPPAE is a multivariate function with respect to $\boldsymbol{W}$, the parameter λ and γ, so we use the multivariate NAG method.

In the multivariate NAG method, the first step is to calculate the gradients of the objective function (8), and the formulas can be expressed as follows:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{W}} &= \frac{d\mathcal{L}_{1st}}{d\boldsymbol{W}} + \lambda\frac{d\mathcal{L}_{2nd}}{d\boldsymbol{W}} + \gamma\frac{d\mathcal{L}_{3rd}}{d\boldsymbol{W}}, \\ \frac{\partial\mathcal{L}}{\partial\lambda} &= tr\left(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{W}\right) - m, \\ \frac{\partial\mathcal{L}}{\partial\gamma} &= tr\left(\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)^T\right).\end{aligned} \tag{10}$$

Where,

$$\begin{aligned}\frac{d\mathcal{L}_{1st}}{d\boldsymbol{W}} &= 2\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}, \\ \frac{d\mathcal{L}_{2nd}}{d\boldsymbol{W}} &= 2\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{W}, \\ \frac{d\mathcal{L}_{3rd}}{d\boldsymbol{W}} &= -4\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}.\end{aligned}$$

Then, with the gradient (10), the multivariate NAG method is used to find the optimal projection matrix $\boldsymbol{W}$, the parameter λ and γ. The optimization process can be expressed as follows. Introducing a velocity variable υ in the conventional Stochastic Gradient Descent method, then randomly initializing velocity variable υ, the parameters $\boldsymbol{W}$, λ, and γ. Taking the parameter $\boldsymbol{W}$ as an example, the NAG method can be summarized in three steps: the parameter $\boldsymbol{W}$ is firstly updated with the last velocity $\boldsymbol{v}_{t-1}$; the velocity variable $\boldsymbol{v}_t$ is updated using the gradient after exponentially weighted averaging; and then the parameter $\boldsymbol{W}_{t+1}$ is updated. Iterate the above three steps in a loop until the optimal value of $\boldsymbol{W}$ is found. The optimal values of the other two parameters λ and γ can be similarly obtained. Based on this, the updating of the parameters $\boldsymbol{W}$, λ, and γ can be summarized in the following three sets of formulas:

$$\begin{aligned}&(\tilde{\boldsymbol{W}} = \boldsymbol{W}_t + \rho\boldsymbol{v}_{t-1}, \boldsymbol{v}_t = \rho\boldsymbol{v}_{t-1} - \alpha\frac{\partial\mathcal{L}}{\partial\boldsymbol{W}}(\tilde{\boldsymbol{W}}, \lambda, \gamma), \\ &\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \boldsymbol{v}_t.) \\ &\tilde{\lambda} = \lambda_t + \rho\boldsymbol{v}_{t-1}, \boldsymbol{v}_t = \rho\boldsymbol{v}_{t-1} - \alpha\frac{\partial\mathcal{L}}{\partial\lambda}(\boldsymbol{W}_t, \tilde{\lambda}, \gamma), \lambda_{t+1} = \lambda_t + \boldsymbol{v}_t. \\ &\tilde{\gamma} = \gamma_t + \rho\boldsymbol{v}_{t-1}, \boldsymbol{v}_t = \rho\boldsymbol{v}_{t-1} - \alpha\frac{\partial\mathcal{L}}{\partial\gamma}(\boldsymbol{W}_t, \lambda, \tilde{\gamma}), \gamma_{t+1} = \gamma_t + \boldsymbol{v}_t.\end{aligned} \tag{11}$$

The optimization process of the parameters $\boldsymbol{W}$, λ, and γ can be organized as Algorithm 1. In Section 2.1, we first construct the original data matrix $\boldsymbol{X}$ and the similarity matrix $\boldsymbol{S}$. Then, initializing the velocity variable υ, the parameters $\boldsymbol{W}$, λ, γ with random values; setting the learning rate $\alpha = 5 * 10^{-3}$, the threshold for the change of loss function $\varepsilon = 0.05$; the momentum coefficient $\rho = 0.9$. Then execute the loop body of Algorithm 1 until the change of objective function is less than the threshold value ε. Finally, we can obtain the optimal projection matrix $\boldsymbol{W}$. In this way, the training phase of the LPPAE method is completed.

Algorithm 1: multivariate NAG method for training the parameters $\boldsymbol{W}$, λ and γ

Input: the original data matrix $\boldsymbol{X}$, the similarity matrix $\boldsymbol{S} = (S_{ij})$
Initialize: initialize velocity variable υ, the parameters $\boldsymbol{W}$, λ, γ randomly, set a learning rate α of gradient descent, and set a threshold ε for the change of loss function.
Pre-calculation: calculate the diagonal matrix $\boldsymbol{D}$, the Laplace matrix $\boldsymbol{L}$, and the objective function $\mathcal{L}$ using Eq. (8).
Repeat:
1. Calculate the gradient $\partial\mathcal{L}/\partial\boldsymbol{W}$, $\partial\mathcal{L}/\partial\lambda$ and $\partial\mathcal{L}/\partial\gamma$ using Eq. (10)
2. Update the parameters $\boldsymbol{W}$, λ, γ using Eq. (11)
3. Calculate the objective function $\mathcal{L}$ using Eq. (8).
Until the change of objective function $< \varepsilon$
Output: the optimal parameters $\boldsymbol{W}$, λ, γ

3.4. LPPAE-based application

By Algorithm 1, the LPPAE model can be trained and then used in many applications. The workflow of the application of LPPAE is shown in Fig. 3.

For the input high-dimensional data $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_l$, an adjacency graph is firstly constructed, and then the input data are sent to train the LPPAE model, then the optimal projection matrix $\boldsymbol{W}$ is outputted. Once the projection matrix $\boldsymbol{W}$ is got, any data point $\boldsymbol{x'}_i$ in high-dimensional manifold space can be embedded as the data point $\boldsymbol{y'}_i$ in low-dimensional space with the linear mapping: $\boldsymbol{y'}_i = \boldsymbol{W}^T\boldsymbol{x'}_i$. The

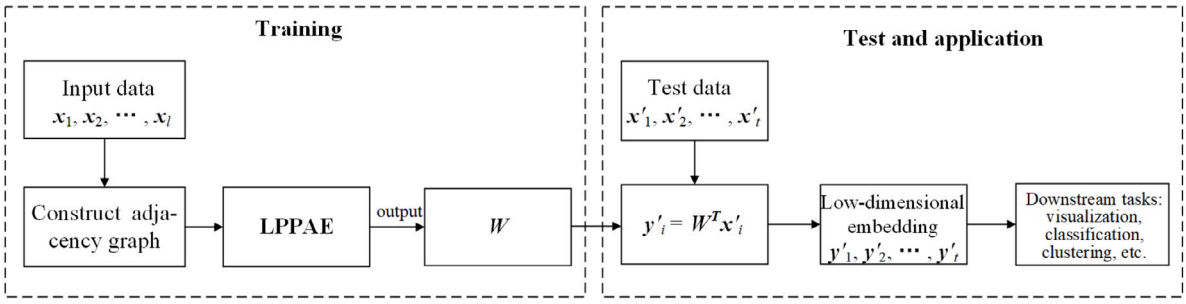


Fig. 3. The workflow of the application of LPPAE.

low-dimensional embedded point $\boldsymbol{y}'_i$ represents the high-dimensional data point $\boldsymbol{x}'_i$, and it can be used to many downstream tasks such as visualization, classification, clustering, etc.

4. Extension on the variants of LPP

The idea of LPPAE can serve as a general framework that can be extended to variants of LPP, thereby obtaining a series of new methods of LPP. As examples, this paper selects two representative variants of LPP, Regularization LPP (RLPP) (Cai, He, Han, 2007) and Exponential LPP (ELPP) (Wang et al., 2011, 2014), and expands them to RLPPAE (RLPP with Autoencoder) and ELPPAE (ELPP with Autoencoder) under this framework.

4.1. RLPPAE

Recall that the objective function of the RLPP method (Cai, He, Han, 2007) is

$$\min_{\boldsymbol{W}^T(r\boldsymbol{I}+\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T)\boldsymbol{W}=\boldsymbol{I}} tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}), \tag{12}$$

where r is a parameter. Like LPPAE, the objective function of RLPPAE is also composed of three parts. Based on Eq. (12), the first item of the objective function of RLPPAE is

$$\mathcal{L}_{1st} = tr\left(\boldsymbol{W}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}\right). \tag{13}$$

The second item of the objective function of RLPPAE may be written as:

$$\mathcal{L}_{2nd} = tr(\boldsymbol{W}^T(r\boldsymbol{I} + \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T)\boldsymbol{W}) - m. \tag{14}$$

Considering the encoder–decoder paradigm, the reconstruction item, Eq. (7), may be used as the third item of the objective function of RLPPAE. Thus, the objective function of RLPPAE is:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{1st} + \lambda\mathcal{L}_{2nd} + \gamma\mathcal{L}_{3rd}\\ &= tr\left(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}\right) +\\ &\lambda\left(tr(\boldsymbol{W}^T(r\boldsymbol{I} + \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T)\boldsymbol{W}) - m\right) +\\ &\gamma tr\left(\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)^T\right).\end{aligned} \tag{15}$$

4.2. ELPPAE

Recall that the objective function of the ELPP method (Wang et al., 2011, 2014) is

$$\min_{\boldsymbol{W}^T exp(\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T)\boldsymbol{W}=\boldsymbol{I}} tr(\boldsymbol{W}^T exp(\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T)\boldsymbol{W}). \tag{16}$$

Similar to LPPAE, with Eq. (16) and the encoder–decoder paradigm, the ELPP method may be improved to the ELPPAE method. The objective function of the ELPPAE method may be written as:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{1st} + \lambda\mathcal{L}_{2nd} + \gamma\mathcal{L}_{3rd}\\ &= tr\left(\boldsymbol{W}^T exp(\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T)\boldsymbol{W}\right) +\\ &\lambda\left(tr(\boldsymbol{W}^T exp(\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T)\boldsymbol{W}) - m\right) +\\ &\gamma tr\left(\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^T\right)^T\right).\end{aligned} \tag{17}$$

The RLPPAE and ELPPAE method have also two free parameters: λ and γ. Like Section 3.2, we can also analyze them. Empirically, the parameter λ also takes in the interval [0, 2], and the parameter γ takes in the interval [0, 3].

5. Experiments

In the experiment section, we evaluate the proposed LPPAE, RLPPAE, and ELPPAE through several experiments and compare them with LPP, RLPP, and ELPP, respectively. PCA and LPP are also used for experiments and comparison. For the LPP method, PCA is first used to reduce the dimension of the original sample. For all datasets, 98% of the principal components are retained.

We first perform a visualization experiment on a synthesized dataset, then make classification experiments on a Handwritten Alphadigits dataset, an object dataset, and two face datasets. Finally, we also compare the LPPAE method with the latest improved version of LPP on three datasets.

5.1. Experiments on synthesized data

The experiment is conducted on the Swiss roll dataset generated by 5000 random sample points. The generating function of the Swiss roll data points is $x = tcost$, $y = h$, $z = tsint$, where $t \in [3\pi/2, 9\pi/2]$, $h \in [0, 41]$.

For the LPP, RLPP, ELPP, LPPAE, RLPPAE, and ELPPAE methods, Fig. 4 shows the two-dimensional (2D) embedding of six methods on the Swiss roll dataset. In Fig. 4, the first row is the three-dimensional (3D) Swiss roll, the second row is the 2D projection of LPP, RLPP, and ELPP on the 3D Swiss roll respectively, and the third row is the 2D projection of LPPAE, RLPPAE and ELPPAE on the 3D Swiss roll respectively.

As seen from Fig. 4, the LPPAE, RLPPAE and ELPPAE methods in the third row better preserve the intrinsic structure of the data points, and the "close" points in the 3D figure are also "close" in the 2D figure, for example, the red and yellow dots. However, the 2D projections of the LPP, RLPP, and ELPP methods in the second row are relatively "scattered" and do not "unfold" as well as the figures in the third row. Additionally, compared with the second row, the separability of each color point of LPPAE, RLPPAE, and ELPPAE in the third row is better. Although the 2D projection of ELPP is comparable to that of LPPAE, RLPPAE, and ELPPAE, the yellow dots in the 2D projection of ELPP are not "close", while the yellow dots in the 3D original figure are "close".

5.2. Experiments on real data

In this section, we conduct experiments on the Handwritten Alphadigits dataset (Lucas, 2001), COIL-20 object dataset (Nene, Nayar, Murase, et al., 1996), Yale (Belhumeur, Hespanha, & Kriegman, 1997) and AR (Martinez & Benavente, 1998) face datasets. We first introduce four datasets, then present the parameter settings of experiments, and finally compare the classification results of these methods on the four datasets.

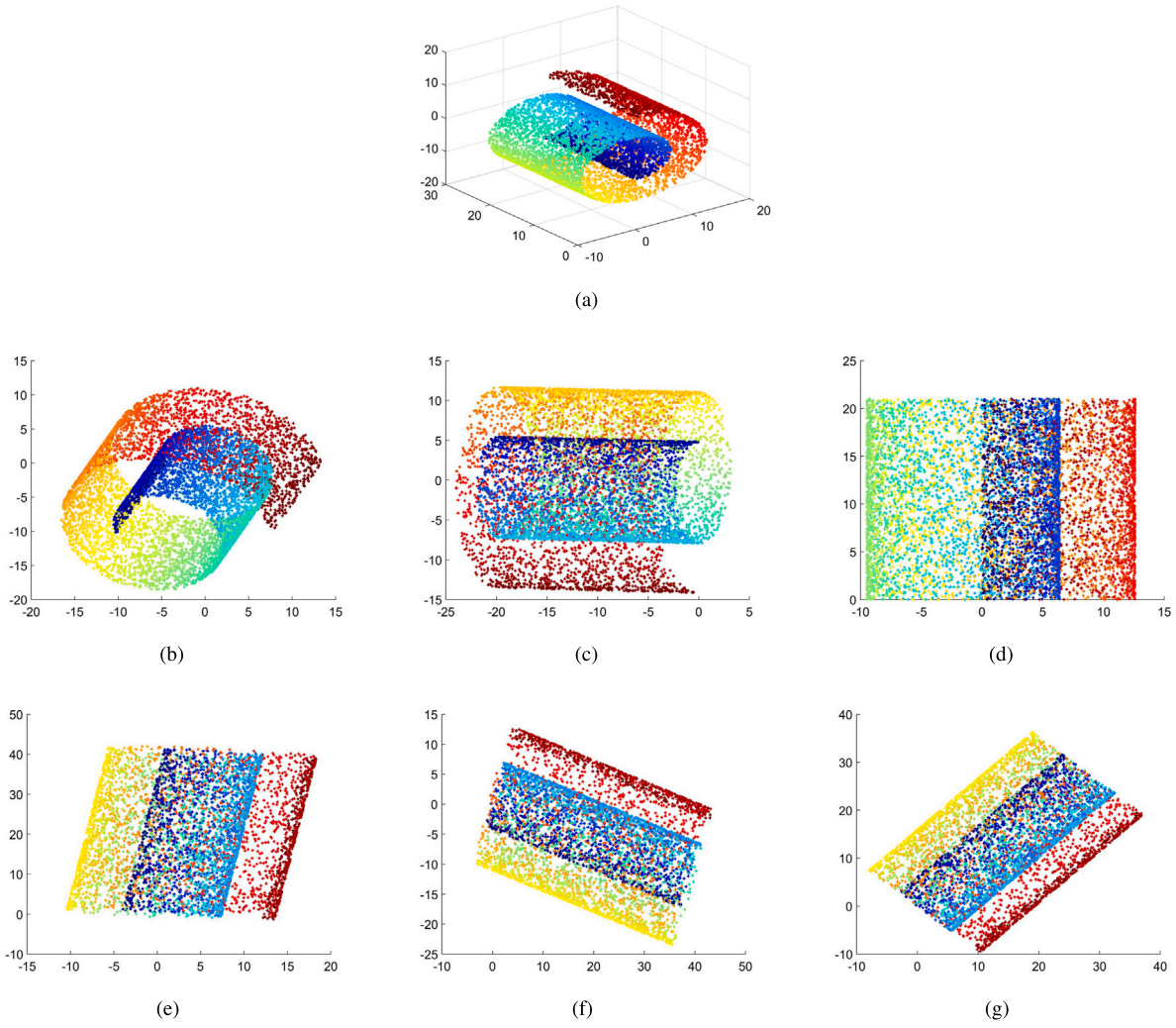


Fig. 4. Two-dimensional embedding on Swiss roll dataset. (a) Swiss roll data set (b) LPP (c) RLPP (d) ELPP (e) LPPAE (f) RLPPAE (g) ELPPAE.

5.2.1. Experimental datasets

Handwritten Alphadigits dataset. This dataset comprises 10 digits of "0" through "9" and 26 capital letters "A" through "Z", with 39 examples of each class. Each sample is a handwritten image. In the experiment, the image size is 20 × 16 pixels. The database is publicly available on S. Roweis' web page (Lucas, 2001).

COIL-20 object dataset. The Columbia University Image Library (COIL-20) contains 20 objects. For each object, a photo is taken every 5° and rotated 360° horizontally in total, so there are 72 images for each object. In the experiment, the image size is adjusted to 32 × 32 pixels (Nene et al., 1996).

The Yale face dataset. This dataset contains 165 grayscale images of 15 individuals, with 11 samples of everyone. The images contain variations in facial expressions, lighting, with/without glasses. In this experiment, all images are aligned based on eye coordinates and are cropped and scaled to 24 × 24 (Belhumeur et al., 1997).

The AR face dataset. It collects face images of 126 people (including 70 men and 56 women). The image has changed such as expression, lighting, and occlusion (whether wearing sunglasses or scarves). The image set has more than 4000 color images. The size of each image is 40 × 50 pixels in the experiments (Martinez & Benavente, 1998).

Some sample images of these datasets are shown in the following Fig. 5.

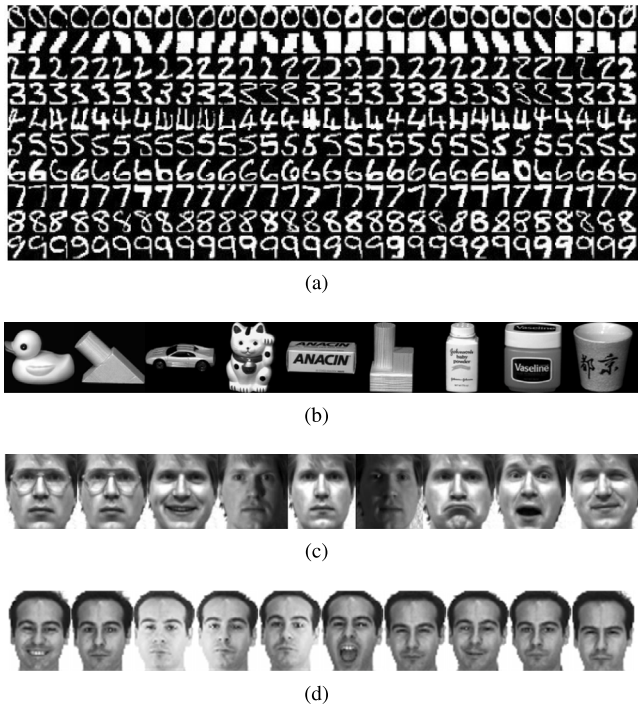
5.2.2. Parameter settings

In each data set, p images for each class are randomly selected to form the training set, and the remaining images are used as the test set. Where the p-value indicates the number of images per class that are randomly selected as training samples in each dataset. For the given p, the reduced dimensionality is incremented from 10 to 100 at step size 5. For the given p and subspace dimension, the neighborhood size parameter k is taken from 5, 10, 15, 20, 25. The best recognition rate from the best k value is used as the recognition rate in the current p and subspace dimensions. And then, for each of the subspace dimensions, we may get the corresponding recognition rate.

We regard the above process as a cycle. For the p training samples, we calculate 10 cycles. In this way, there are 10 recognition rates for each subspace dimension, and then we take their average value as the recognition rate for the current p and subspace dimensions. Finally, we take the best recognition rate from the best subspace dimensions as the result of the training sample = p.

5.2.3. Experimental results

Tables 2 to 5 present the best recognition rates, standard deviations, and optimal subspace dimensions for the seven methods mentioned above on the four datasets. For each class, the p value is 5, 7, and 9 for the Handwritten Alphadigits dataset; 6, 8, and 10 for the COIL-20 dataset; 2, 3, and 4 for the Yale dataset; and 3, 4, and 5 for the AR dataset.



(a) (b) (c) (d)

Fig. 5. Some sample images in the experiments. (a) Handwritten Alphadigits dataset (b) COIL-20 object dataset (c) Yale face dataset (d) AR face dataset.

Table 2
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on Handwritten Alphadigits data set.

Method	5 trains	7 trains	9 trains
PCA	39.57 ± 3.56(30)	39.88 ± 3.12(25)	40.25 ± 1.93(35)
LPP	39.13 ± 2.59(20)	40.37 ± 2.30(15)	38.27 ± 0.71(25)
RLPP	47.85 ± 2.05(30)	52.00 ± 0.38(25)	56.45 ± 1.71(25)
ELPP	53.62 ± 1.39(30)	58.16 ± 0.79(20)	62.07 ± 0.59(95)
LPPAE	**57.45 ± 2.11(25)**	**61.09 ± 1.50(20)**	**64.33 ± 1.05(20)**
RLPPAE	**57.56 ± 1.26(40)**	**61.56 ± 0.84(30)**	**64.23 ± 1.06(40)**
ELPPAE	**56.06 ± 0.62(95)**	**61.07 ± 1.04(30)**	**64.09 ± 1.00(70)**

Table 3
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on COIL-20 data set.

Method	6 trains	8 trains	10 trains
PCA	78.53 ± 2.85(60)	80.59 ± 1.06(50)	83.17 ± 1.77 (40)
LPP	80.28 ± 2.85(30)	81.22 ± 1.20(40)	84.49 ± 1.19 (40)
RLPP	86.57 ± 2.29(15)	89.53 ± 0.90(40)	92.10 ± 1.95(50)
ELPP	86.97 ± 1.02(10)	90.55 ± 2.10(40)	93.52 ± 1.20(50)
LPPAE	**89.42 ± 1.24(60)**	**92.75 ± 1.98(60)**	**94.58 ± 1.16(60)**
RLPPAE	**90.50 ± 0.94(80)**	**92.12 ± 2.60(75)**	**94.67 ± 1.20(15)**
ELPPAE	**90.27 ± 1.41(60)**	**92.90 ± 1.03(65)**	**94.51 ± 1.08(50)**

Table 4
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on Yale data set.

Method	2 trains	3 trains	4 trains
PCA	58.60 ± 4.20(45)	62.60 ± 3.90(60)	69.20 ± 4.16(75)
LPP	59.26 ± 3.02(60)	64.72 ± 1.71(85)	69.84 ± 1.96(60)
RLPP	61.23 ± 2.72(60)	67.78 ± 1.04(100)	74.29 ± 2.81(60)
ELPP	48.15 ± 4.57(15)	56.94 ± 6.18(65)	73.02 ± 1.96(65)
LPPAE	**67.70 ± 3.67(80)**	**73.00 ± 3.080(70)**	**75.24 ± 2.82(55)**
RLPPAE	**68.74 ± 3.69(80)**	**72.33 ± 2.81(100)**	**75.81 ± 5.40(70)**
ELPPAE	**66.22 ± 3.85(80)**	**72.33 ± 2.75(90)**	**75.42 ± 4.42(95)**

Table 5
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on AR data set.

Method	3 trains	4 trains	5 trains
PCA	76.64 ± 3.82(80)	80.67 ± 2.50(100)	82.53 ± 3.31(90)
LPP	77.19 ± 4.06(100)	79.81 ± 2.06(95)	80.82 ± 4.43(85)
RLPP	79.05 ± 3.59(100)	83.11 ± 2.83(95)	88.69 ± 2.38(85)
ELPP	75.28 ± 4.59(100)	81.36 ± 1.16(95)	86.76 ± 4.39(95)
LPPAE	**83.95 ± 4.06(75)**	**86.46 ± 2.26(85)**	**89.68 ± 3.37(80)**
RLPPAE	**81.53 ± 3.10(80)**	**85.92 ± 2.40(65)**	**90.56 ± 2.13(85)**
ELPPAE	**83.31 ± 3.98(100)**	**86.10 ± 4.10(65)**	**93.06 ± 3.37(80)**

Table 6
Consumed time of LPPAE (in second).

Datasets	Training	Optimization	Testing
Handwritten Alphadigit	0.202	0.167	0.001
Yale	0.460	0.437	0.042
Coil-20	2.022	1.921	0.020
ORL	1.944	1.862	0.014

Generally, the performances of these methods are also related to the reduced dimension. Based on the above experimental results, the recognition results of each method in different reduced dimensions are analyzed, as shown in Fig. 6(a)–(d). In these Figures, the dotted lines represent the LPP, RLPP, and ELPP methods, and solid lines represent the LPPAE, RLPPAE, and ELPPAE methods respectively.

As indicated in Tables 2–5, the newly introduced methods, namely LPPAE, RLPPAE, and ELPPAE, demonstrate superior classification capabilities compared to their counterparts LPP, RLPP, and ELPP. Specifically, the recognition rates of LPPAE on Handwritten Alphadigits, COIL-20, Yale, and AR datasets are 18.32, 9.13, 8.44, 6.76% higher than those of the original LPP respectively. Similarly, RLPPAE always outperforms RLPP and ELPPAE outperforms ELPP.

As depicted in Fig. 6(a)–(d), the recognition rates of LPPAE, RLPPAE, and ELPPAE are significantly better than those of LPP, RLPP, and ELPP across different subspace dimensions. This improved performance could be attributed to the additional decoder present in LPPAE, RLPPAE, and ELPPAE that can back-correct the low-dimensional embeddings, thereby enhancing their accuracy. Consequently, when these corrected embeddings are utilized as image features for classification tasks, superior results can be achieved.

We have also evaluated the performance of the proposed LPPAE method in terms of training time, parameter optimization time, and testing time. These metrics are reported in Table 6. As indicated in Table 6, the new method demonstrates reduced training time, parameter optimization time, and testing time, which underscores its usability. Where, the LPPAE method is taken as an example to report the training time, parameter optimization time, and testing time, the three times of the RLPPAE and ELPPAE methods can be similarly reported.

5.3. The comparisons with the latest methods

Recently, the DNLPP method was introduced (Long et al., 2019), which enhances LPP through the use of neural networks. The primary distinction between DNLPP and the newly proposed LPPAE method lies in their architectural differences. While DNLPP also employs a neural network to map the input layer to a low-dimensional output layer, replacing linear projection, it does not incorporate an auto-encoder architecture and thus lacks a decoder. On the other hand, the LPPAE method is founded on the encoder–decoder paradigm. It not only utilizes encoders but also employs decoders to reconstruct the original high-dimensional data from the projected low-dimensional data. Given the similarities between DNLPP and LPPAE, we have conducted experiments to compare their performances, which will be discussed in this section.

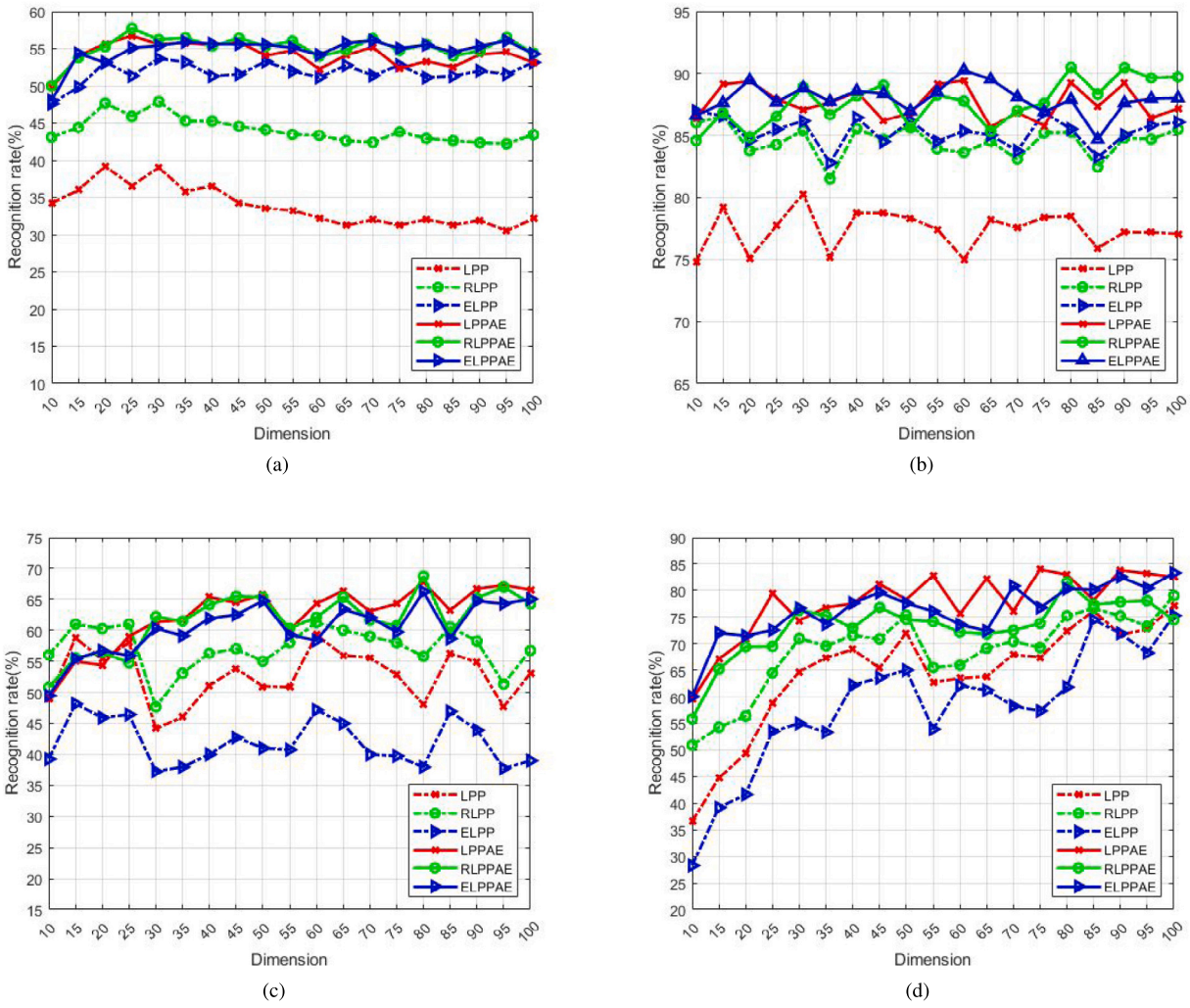


Fig. 6. The recognition rates versus the subspace dimension. (a) Handwritten Alphadigits dataset (5 training samples) (b) COIL-20 object dataset (6 training samples) (c) Yale face dataset (2 training samples) (d) AR face dataset (3 training samples).

According to the report of DNLPP in Long et al. (2019), we chose to conduct experiments on MNIST, COIL-20, and ORL datasets.

The MNIST dataset consists of 10 images of numbers from "0" to "9", and each number has 7000 images. The original black and white images were size normalized to fit a 20 × 20 pixel box and they are centered in a 28 ×28 image by computing the center of mass of the pixels (LeCun, 1998).

The ORL face dataset has 400 images composed of 40 persons and there are 10 images of each person. The images were taken at different times, varying the lighting, facial expressions, and facial details. All images are resized into 32 × 32 pixels in the experiment (Samaria & Harter, 1994). The COIL-20 dataset has been introduced in the previous Section. Some sample images of the three datasets are shown in the following Fig. 7.

In order to compare under the same experimental parameters, the experimental parameters of DNLPP are used (Long et al., 2019). The training image number p for each class on the MNIST, COLI-20, and ORL datasets are 6000, 60, and 8 respectively, the reduced dimensionalities are 10, 40, 70, and 100.

The proposed LPPAE method is also compared with the latest LPP variant methods, including the LPP+TSVD (Wei et al., 2020), LPP+TR (Wei et al., 2020), LAPP (Wang et al., 2020), LPPMDC (Ran et al., 2022). Besides, the LPP, RLPP (Cai, He, Han, 2007), ELPP (Wang et al., 2011, 2014) methods are also compared with the LPPAE method. These methods use the above experimental parameters.

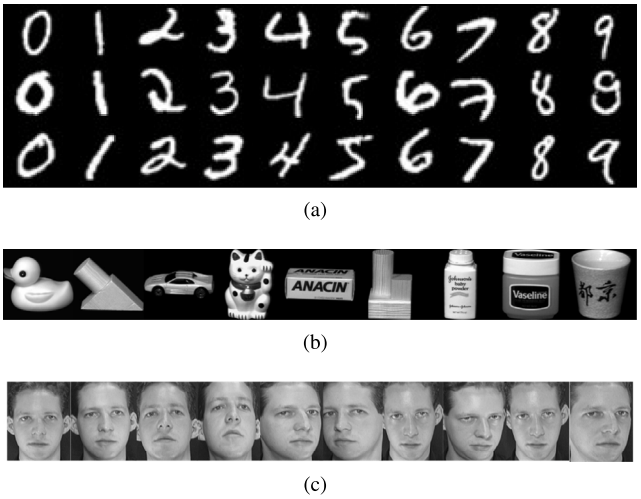


Fig. 7. Some sample images in the experiments. (a) MNIST dataset (b) COIL-20 object dataset (c) ORL face dataset.

Table 7
Average recognition accuracy (in percent) of corresponding dimensions on the three datasets over 50 random splits.

Dim	Dataset	LPP	RLPP	ELPP	LPPMDC	LAPP	LPP+TSVD(0.98)	LPP+TR(0.05)	DNLPP	LPPAE
10	MNIST	37.60	90.17	10.00	77.50	50.00	71.00	79.00	**95.25**	92.60
	COLI20	37.92	87.93	84.55	79.65	–	88.00	95.00	65.92	**99.00**
	ORL	80.83	78.05	52.52	88.00	84.00	79.00	77.00	**90.83**	89.72
40	MNIST	33.20	96.93	10.00	82.50	51.00	87.50	87.00	95.49	**97.62**
	COLI20	72.50	91.97	98.07	83.45	–	95.00	97.00	95.83	**99.94**
	ORL	83.33	88.82	84.30	93.50	94.81	93.00	92.00	94.17	**94.45**
70	MNIST	49.20	96.67	43.37	85.60	52.00	88.00	89.00	94.92	**97.44**
	COLI20	70.41	95.05	99.17	89.56	–	94.50	97.00	94.59	**99.94**
	ORL	85.00	90.82	89.25	94.50	91.00	92.00	93.00	93.30	**95.95**
100	MNIST	48.40	96.68	63.68	85.60	60.00	88.00	89.00	95.49	**98.10**
	COLI20	68.75	97.20	99.34	90.50	–	94.00	97.00	95.83	**99.95**
	ORL	85.83	91.22	90.75	93.50	90.00	92.00	93.00	95.00	**96.57**

For the LPP, RLPP, ELPP, LPPMDC, and LPPAE method, for the p training samples and subspace dimension, we calculate 50 cycles. Thus, there are 50 recognition rates on a given dataset and each subspace dimensionality. The final recognition rate on a dataset and subspace dimensionality is the average of these 50 results. Since there are no source codes for the LPP+TSVD, LPP+TR, LAPP, and DNLPP methods, their results are derived from the corresponding literature. On the three datasets, for these methods mentioned above, the experimental results are shown in Table 7.

When compared to the latest enhanced version of LPP, LPPAE generally exhibits superior recognition rates, with the exception of instances where the subspace dimension equals 10 on the MNIST and ORL datasets. Both DNLPP and LPPAE are built upon a neural network structure. While DNLPP employs a deep neural network, LPPAE utilizes a simpler two-layer network structure. However, as indicated in Table 7, LPPAE consistently outperforms DNLPP. This could be attributed to their differing network structures. Although both methods leverage network structures, DNLPP employs a unidirectional network mapping from high-dimensional space to low-dimensional space. In contrast, LPPAE incorporates an autoencoder that includes a decoder for back-correcting the low-dimensional embeddings, thereby enhancing their accuracy.

5.4. Convergence analysis

Taking LPPAE as an example, the convergence curves of LPPAE on four datasets are presented in Fig. 8. Where, the training samples are 5, 6, 2, 3, and 2 in Handwritten Alphadigits, COIL-20, Yale, AR, and ORL datasets respectively, the neighborhood size parameter $k = 10$, and the subspace dimension is 40. Because the initial value of the objective function on the MNIST dataset is very large, the convergence curve on this dataset is difficult to display in the same figure as that of the other datasets. Therefore, the convergence curve of the MNIST dataset is not shown in Fig. 8. But it still converges within 30 steps.

As illustrated in Fig. 8, LPPAE is capable of converging within 30 steps across all datasets. The objective function of LPPAE is a multivariate function with respect to $\boldsymbol{W}$, λ, and γ. To find the optimal solutions, we employ the Multivariate Stochastic Gradient Descent with Nesterov momentum method (Multivariate NAG). This method accelerates the optimization process and enables the rapid discovery of optimal solutions. The robust convergence of this method underscores the usability of the model.

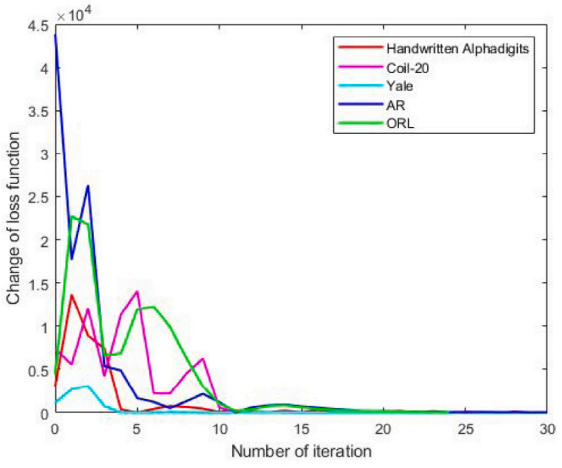


Fig. 8. The convergence curve of LPPAE.

6. Discussion

The proposed LPPAE method creatively applied the encoding-decoding mechanism of Autoencoder to the conventional LPP, making the obtained low-dimensional embeddings more accurate and representative, thereby improving the performance of LPP. In theory, this is an important improvement to LPP. The idea of LPPAE may also be seen as a general framework and applied to other dimensionality reduction methods, such as Isomap projection (IsoP) (Cai, He, Han, et al., 2007), neighborhood preserving embedding (NPE) (He et al., 2005), linear LTSA (LLTSA) (Zhang et al., 2007), etc., thus a series of new methods can be gotten.

As an effective dimensionality reduction and feature extraction method, the proposed LPPAE method can be applied to visual analysis, clustering, and classification problems in fields such as facial recognition, financial management, and fault diagnosis, and can improve and enhance the application effect. However, in real life, images may not be as standard as those in an image set. For example, facial images in face recognition may be affected by lighting, occlusion, and other noises, all of which can affect the accuracy of classification. The current method in this paper does not consider these factors. Inspired by Ref. Xu, Lv, Li, Sun, and Sheng (2022), in our future research, we will make further improvements to LPPAE to avoid disturbing factors such as noise, e.g., the improvement of LPPAE based on L1-norm or Lp-norm metrics, which is insensitive to outliers and has more stable performance.

While LPPAE has demonstrated superior performance compared to LPP and its most recent enhancement, it still possesses certain limitations that could be addressed in future iterations. One such limitation is the relatively weak learning ability of LPPAE. This could be attributed to two factors: firstly, both LPP and Autoencoder are linear structures, which inherently limit their learning capabilities; secondly, LPPAE is based on an encoding–decoding paradigm, which, due to its simplistic structure, struggles with handling large-scale complex scenarios. In future developments, the learning ability of LPP could be enhanced by incorporating network structures with stronger learning capabilities. For instance, a multilayer structure based on a stacked autoencoder could be used to develop a deeper version of LPPAE, potentially boosting its learning ability. Another limitation of

LPPAE is that the construction of LPP primarily focuses on preserving the geometric relationship between neighboring points, without considering the overall probability distribution of the samples. This could lead to a discrepancy between the probability distribution of the low-dimensional embeddings and that of the original samples. Future research could explore improvements from this perspective.

7. Conclusions

In this paper, we proposed a novel LPP method, termed as LPPAE (Locality Preserving Projections with Autoencoder), which incorporates the encoder–decoder paradigm of Autoencoders. The key innovation of LPPAE lies in its construction of a bidirectional mapping: one from the high-dimensional space to the low-dimensional space via the encoder, and the other from the low-dimensional space back to the original high-dimensional space using the decoder. This approach ensures that the low-dimensional features more accurately and effectively "represent" the original high-dimensional data. LPPAE solves the problem that conventional LPP has only a one-way mapping and the resulting low-dimensional features are not accurate enough. The experimental results on Handwritten Alphadigits, COIL-20, Yale, AR datasets show that the recognition rate of LPPAE is 26.06, 10.09, 5.40, 8.86% higher than that of the original LPP, respectively. The experimental results on the MNIST, COIL-20, and ORL datasets indicate that LPPAE always outperforms the latest improved methods of LPP. Moreover, LPPAE exhibits robust convergence within 30 steps, thereby ensuring the model's usability. The concept of LPPAE can serve as a universal framework and be extended to variants of LPP and other similar methods. In this regard, this paper extends two variants of LPP, RLPP and ELPP, to RLPPAE and ELPPAE. Experimental results indicate that both RLPPAE and ELPPAE outperform RLPP and ELPP, respectively.

CRediT authorship contribution statement

Ruisheng Ran: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Ji Feng:** Software, Validation, Software. **Zheng Li:** Data curation, Visualization. **Jinping Wang:** Data curation, Visualization. **Bin Fang:** Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by Chongqing Municipal Education Commission under grant KJZD-K202100505, the Ministry of Education of China under grant 20YJAZH084, Chongqing Science and Technology Bureau under grant cstc2020jscx-msxmX0190 and cstc2019jscx-mbdxX0061.

References

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720.

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems, Vol. 14*.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems, Vol. 19*.

Cai, D., He, X., & Han, J. (2007). Spectral regression for efficient regularized subspace learning. In *2007 IEEE 11th international conference on computer vision* (pp. 1–8). IEEE.

Cai, D., He, X., Han, J., et al. (2007). Isometric projection. In *AAAI* (pp. 528–533).

Chen, W. J., Li, C. N., Shao, Y.-H., Zhang, J., & Deng, N.-Y. (2019). 2DRLPP: Robust two-dimensional locality preserving projection with regularization. *Knowledge-Based Systems*, *169*, 53–66.

Du, H., Li, G., Wang, S., & Zhang, F. (2019). Discriminant locality preserving projections based on L2, p-norm for image feature extraction and recognition. *Journal of Visual Communication and Image Representation*, *58*, 166–177.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

He, X., Cai, D., Yan, S., & Zhang, H.-J. (2005). Neighborhood preserving embedding. In *Tenth IEEE international conference on computer vision (ICCV'05) Vols. 1, 2* (pp. 1208–1213).

He, X., & Niyogi, P. (2004). Locality preserving projections. In *Advances in neural information processing systems, Vol. 16* (pp. 153–160).

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Hu, X., Sun, Y., Gao, J., Hu, Y., & Yin, B. (2018). Locality preserving projection based on F-norm. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 32*.

Kingma, D. P., & Welling, M. (2022). Auto-encoding variational bayes. arXiv:1312.6114.

LeCun, Y. (1998). The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/. 1.

Liu, Y., Gao, Q., Gao, X., & Shao, L. (2018). {L_{2, 1}}-norm discriminant manifold learning. *IEEE Access*, *6*, 40723–40734.

Long, T., Gao, J., Yang, M., Hu, Y., & Yin, B. (2019). Locality preserving projection via deep neural network. In *2019 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Lu, C., Liu, X., & Liu, W. (2012). Face recognition based on two dimensional locality preserving projections in frequency domain. *Neurocomputing*, *98*, 135–142.

Lu, J., & Tan, Y.-P. (2009). Regularized locality preserving projections and its extensions for face recognition. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *40*(3), 958–963.

Lu, G.-F., Wang, Y., Zou, J., & Wang, Z. (2018). Matrix exponential based discriminant locality preserving projections for feature extraction. *Neural Networks*, *97*, 127–136.

Lu, Y., Yuan, C., Lai, Z., Li, X., Wong, W. K., & Zhang, D. (2017). Nuclear norm-based 2DLPP for image classification. *IEEE Transactions on Multimedia*, *19*(11), 2391–2403.

Lucas, S. (2001). Handwritten alphadigits dataset. https://cs.nyu.edu/~roweis/data.html. 1.

Martinez, A., & Benavente, R. (1998). The ar face database: Cvc technical report. (p. 4).

Nene, S. A., Nayar, S. K., Murase, H., et al. (1996). *Columbia object image library (COIL-20): Technical report*, (p. 6). Columbia university.

Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence O (1/k2). In *Dokl. akad. nauk. SSSR, Vol. 269* (p. 543).

Pang, Y., & Yuan, Y. (2010). Outlier-resisting graph embedding. *Neurocomputing*, *73*(4–6), 968–974.

Ran, R., Feng, J., Zhang, S., & Fang, B. (2020). A general matrix function dimensionality reduction framework and extension for manifold learning. *IEEE Transactions on Cybernetics*, *52*(4), 2137–2148.

Ran, R., Qin, H., Zhang, S., & Fang, B. (2022). Simple and robust locality preserving projections based on maximum difference criterion. *Neural Processing Letters*, *54*(3), 1783–1804.

Ran, R., Ren, Y., Zhang, S., & Fang, B. (2021). A novel discriminant locality preserving projections method. *Journal of Mathematical Imaging and Vision*, *63*, 541–554.

Ranzato, M., Boureau, Y. L., Cun, Y., et al. (2007). Sparse feature learning for deep belief networks. In *Advances in neural information processing systems, Vol. 20*.

Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *10*(2), 159–203.

Samaria, F. S., & Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE workshop on applications of computer vision* (pp. 138–142). IEEE.

Shikkenawis, G., & Mitra, S. K. (2015). 2D orthogonal locality preserving projection for image denoising. *IEEE Transactions on Image Processing*, *25*(1), 262–273.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71–86.

Wang, S. J., Chen, H. L., Peng, X. J., & Zhou, C. G. (2011). Exponential locality preserving projections for small sample size problem. *Neurocomputing*, *74*(17), 3654–3662.

Wang, Q., Gao, Q., Xie, D., Gao, X., & Wang, Y. (2016). Robust DLPP with nongreedy L1-norm minimization and maximization. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(3), 738–743.

Wang, S. J., Yan, S., Yang, J., Zhou, C. G., & Fu, X. (2014). A general exponential framework for dimensionality reduction. *IEEE Transactions on Image Processing*, *23*(2), 920–930.

Wang, A., Zhao, S., Liu, J., Yang, J., Liu, L., & Chen, G. (2020). Locality adaptive preserving projections for linear dimensionality reduction. *Expert Systems with Applications*, *151*, Article 113352.

Wei, W., Dai, H., & Liang, W. (2020). Regularized least squares locality preserving projections with applications to image recognition. *Neural Networks*, *128*, 322–330.

Xu, Z., Lv, Z., Li, J., Sun, H., & Sheng, Z. (2022). A novel perspective on travel demand prediction considering natural environmental and socioeconomic factors. *IEEE Intelligent Transportation Systems Magazine*, *15*(1), 136–159.

Yin, S., Sun, Y., Gao, J., Hu, Y., Wang, B., & Yin, B. (2021). Robust image representation via low rank locality preserving projection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *15*(4), 1–22.

Yu, W., Teng, X., & Liu, C. (2006). Face recognition using discriminant locality preserving projections. *Image and Vision Computing*, *24*(3), 239–248.

Yu, W., Wang, R., Nie, F., Wang, F., Yu, Q., & Yang, X. (2018). An improved locality preserving projection with L1-norm minimization for dimensionality reduction. *Neurocomputing*, *316*, 322–331.

Zhang, J., Wang, J., & Cai, X. (2017). Sparse locality preserving discriminative projections for face recognition. *Neurocomputing*, *260*, 321–330.

Zhang, T., Yang, J., Zhao, D., & Ge, X. (2007). Linear local tangent space alignment and application to face recognition. *Neurocomputing*, *70*(7–9), 1547–1553.