



Linear local tangent space alignment with autoencoder

Ruisheng Ran¹ · Jinping Wang¹ · Bin Fang²

Received: 17 October 2022 / Accepted: 20 March 2023 / Published online: 28 April 2023
© The Author(s) 2023

Abstract

Linear local tangent space alignment (LLTSA) is a classical dimensionality reduction method based on manifold. However, LLTSA and all its variants only consider the one-way mapping from high-dimensional space to low-dimensional space. The projected low-dimensional data may not accurately and effectively “represent” the original samples. This paper proposes a novel LLTSA method based on the linear autoencoder called LLTSA-AE (LLTSA with Autoencoder). The proposed LLTSA-AE is divided into two stages. The conventional process of LLTSA is viewed as the encoding stage, and the additional and important decoding stage is used to reconstruct the original data. Thus, LLTSA-AE makes the low-dimensional embedding data “represent” the original data more accurately and effectively. LLTSA-AE gets the recognition rates of 85.10, 67.45, 75.40 and 86.67% on handwritten Alphadigits, FERET, Georgia Tech. and Yale datasets, which are 9.4, 14.03, 7.35 and 12.39% higher than that of the original LLTSA respectively. Compared with some improved methods of LLTSA, it also obtains better performance. For example, on Handwritten Alphadigits dataset, compared with ALLTSA, OLLTSA, PLLTSA and WLLTSA, the recognition rates of LLTSA-AE are improved by 4.77, 3.96, 7.8 and 8.6% respectively. It shows that LLTSA-AE is an effective dimensionality reduction method.

Keywords Linear local tangent space alignment · Autoencoder · Dimensionality reduction · Manifold learning

Introduction

The information era, a large amount of data has been generated every moment in many fields such as education, medical care, social media, business, etc. However, most of these data cannot be directly applied to the real situation as impurities and redundancy. Thus, data preprocessing such as data cleaning and data transformation is paid more and more attention. The raw data always contains many noises and unnecessary background and the extra information may affect the data usage such as classification, regression, etc. Dimensionality reduction (DR) [1, 2] is a part of data preprocessing

technique. Some important fields are also closely related to data dimension reduction. For example, in the field of object detection [3], the pictures taken by it often have the ultra-high resolution, so DR is needed to reduce the data dimension, which makes the following algorithm more smooth [4]. And in the field of iterative learning control domain which is related to robot field [5], the DR algorithms are also used in the preprocessing stage to reduce the computational complexity and computational time in the iterative learning algorithm [6]. The main goal of DR is to find out the optimal representative features or extract the low-dimensional features from the high-dimensional space to address the curse of dimensionality [7, 8]. Up to now, a variety of DR methods have been proposed to remove the redundant, insignificant, or noisy information from the raw data. For example, the supervised DRs represented by linear discriminant analysis (LDA) [9], the semi-supervised DRs represented by semi-supervised discriminant analysis (SDA) [10], and the unsupervised DRs represented by principal component analysis (PCA) [11].

The DR methods are generally divided into two categories: linear and nonlinear methods [12]. Principal component analysis (PCA) and linear discriminant analysis (LDA) are two of the most representative linear dimensionality reduction

✉ Bin Fang
fb@cqu.edu.cn
Ruisheng Ran
rshran@cqu.edu.cn
Jinping Wang
mw530@foxmail.com

¹ The College of Computer and Information Science, College of Intelligent Science, Chongqing Normal University, Chongqing 401331, China

² The College of Computer Science, Chongqing University, Chongqing 400044, China

techniques. PCA is proposed by Turk and Pentland in 1991, which is also called Eigenface [11] in face recognition. The mathematical foundations of PCA are the properties of the covariance matrix and the special meaning of eigenvectors. And PCA has many different implement algorithms, such as eigenvalues, latent variable analysis, factor analysis etc. [13–15]. LDA is proposed by Fisher in 1936, which is also called Fisherface in face recognition. And its main idea is to maximize the ratio of the between-class scatter to the within-class scatter, thereby maximizing the separability of between-class. However, both PCA and LDA are linear methods.

To address the non-linear problem, a large number of manifold learning algorithms have been proposed [16], such as Laplacian eigenmap (LE) [17], and locally linear embedding (LLE) [18], Isomap [19], local tangent space alignment (LTSA) [20]. All these methods can obtain a low-dimensional embedding which is regarded as the best representative subspace from the high-dimensional non-linear structure of the data. The idea of LE is to use the Laplacian of graphs to find the optimal low-dimensional representation that preserves local neighborhood information of the original manifold. LLE is based on the idea that each data point and its closest certain number of neighbors are viewed as a locally linear patch of the manifold and then reconstruct each data point from its neighbors in the corresponding subspace. The main objective of Isomap is to preserve the best similarity or dissimilarity on the manifold between any pairs of data points and it improves the computation method based on multi-dimensional scaling (MDS) [21, 22]. The idea of LTSA is to represent the local geometry of the high-dimensional manifold by using the tangent space in the neighborhood of a data point and then align those tangent spaces to construct the global coordinate system for the nonlinear manifold.

However, those DR methods mentioned above all face the “out-of-sample” problem [23] which means they are only defined on the training set and do not apply to the test set. To solve the problem, many linearization methods of manifold learning are proposed. For example, Isomap Projection (IsoP) [24] is the linearization of Isomap, locality preserving projection (LPP) [25] is the linearization of LE, neighborhood preserving embedding (NPE) [26] is the linearization of LE, and linear local tangent space alignment (LLTSA) [27] is the linearization of LTSA. Specially, LLTSA first projects the data set into PCA subspace to avoid the singularity of the matrix and throw away the redundancy information. Secondly, it denotes a set k nearest neighbors (KNN) by a matrix for every point. And thirdly, it finds out a low-dimensional embedding of high-dimensional data with a linear mapping. This mapping keeps the structure of the original manifold data points. LLTSA not only uses the tangent space to preserve manifold structure as LTSA does but also solves the

“out-of-sample” problem by providing the linear mapping available on both the training set and test set.

The objective of LLTSA is to seek out a low-dimensional embedding that keeps the structure of the local geometry from the original manifold data points. More specifically, LLTSA constructs a neighbor graph for each data point by k nearest neighbors (KNN), and then, by using tangent space, it computes a local linear approximation for the data set. Last, LLTSA obtains the optimal mapping by minimizing the error from high-dimensional manifold to the low-dimensional feature space. Besides, the eigenproblem of molecular alignment [28] is related to LLTSA. LLTSA aims to align the local tangent spaces in low-dimensional spaces, and eigenproblem translated for alignment of molecules aims to distinguish different molecular arrangements. Their goal is to align the items in the proper position in space. Furthermore, There are already several improved methods based on LLTSA. For example, adaptive linear local tangent space alignment (ALLTSA) [29] is aimed at solving the problem that it is always hard to choose the best k in the neighborhood selection. Orthogonal linear local tangent space alignment (OLLTSA) [30] eliminates the redundant information by taking the constraint of the basis vector on the orthogonal form. The warp linear local tangent space alignment [31] constructs a curved local tangent space measure to improve the performance. An improved linear local tangent space alignment algorithm based on principal component analysis (PLLTSA) [32] improves LLTSA by considering not only the local geometric structure of the data set but also the global structure of the samples. Weighted linear local tangent space alignment (WLLTSA) [33] is a recently proposed improved method of LLTSA, which uses the weighted version of PCA to approximate local tangent space in each neighborhood instead of conventional PCA.

As the conventional LLTSA method is unsupervised, some supervised improved methods are proposed to exert the label information to obtain more accurate low-dimensional embedding. By taking the label information into consideration and redefining the distance matrix, discriminant linear local tangent space alignment algorithm (DLLTSA) [34] and supervised-linear local tangent space alignment (S-LLTSA) [35] are proposed. And based on the above idea, many modified algorithms are proposed. For example, adaptive discriminant linear local tangent space alignment (ADLLTSA) [36] is proposed by Lv, whose main idea is to add the adaptive neighborhood selection to DLLTSA. And orthogonal discriminant linear local tangent space alignment (ODLLTSA) [37] orthogonalizes the subspace generated by DLLTSA. Marginal discriminant linear local tangent space alignment (MDLLTSA) [38] improves LLTSA by considering the margin of intraclass and interclass.

However, the LLTSA method and all its extended versions obtain the embedding only by considering the one-way map-

ping from high-dimensional space to low-dimensional space. This mapping enables the embedded low-dimensional data points to preserve the local neighborhood information of the original samples partly. At the same time, the information about the original high-dimensional space may lose. So it may not “represent” the original sample very accurately and effectively.

To address the above problems, in this paper, based on the encoder–decoder paradigm, we present a novel LLTSA method named LLTSA-AE (LLTSA with autoencoder). Specifically, under the condition of maintaining the neighborhood structure information of the samples, the data points in high-dimensional manifold space are encoded into data points in low-dimensional space by using the conventional LLTSA projection model. Furthermore, we also use the decoder to reconstruct the original high-dimensional data points from the low-dimensional data points and minimize the error between the original space and the reconstructed space. However, the original LLTSA algorithm only considers one-way mapping from high-dimensional space to low-dimensional space, and the modified methods of LLTSA all focus on enhancing the one-way mapping to improve performance. That is, compared with the conventional LLTSA and the modified methods of LLTSA, the new LLTSA method has an additional reconstruction stage. This stage enables the low-dimensional data to retain as much information as possible about the original high-dimensional data, so the embedded low-dimensional data “represent” the original samples more accurately and effectively.

The rest of this paper proceeds as follows: in “The related works”, we review the LLTSA method and autoencoder. In “Linear local tangent space alignment with autoencoder”, we propose the novel LLTSA method with the encoder–decoder paradigm. In “Experimental results”, we make experiments to evaluate the new method. The conclusion and future work are given in “Conclusion”.

The related works

Local tangent space alignment

LLTSA is the linearization version of LTSA, so the algorithm of LTSA is demonstrated as follows. Suppose there are l original samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l$ in \mathbb{R}^n space and denote $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l]$.

1. Construct neighborhoods: to construct a neighborhood for each point. In the original high-dimensional input space, the k nearest neighbors of each point are found based on Euclidean distance.
2. Extracting local coordinates: to preserve the local structure of the data points, the local coordinates are computed by solving the optimal linear problem Eq. (1).

$$\begin{aligned} & \sum_{j=1}^{k_i} \left\| \mathbf{x}_{ij} - (\bar{\mathbf{x}}_i + \mathbf{Q}_i \boldsymbol{\theta}_j^{(i)}) \right\|^2 \\ &= \min_{\mathbf{x}, \{\boldsymbol{\theta}_j\}, \mathbf{Q}^T \mathbf{Q} = \mathbf{I}} \left\| \mathbf{x}_{ij} - (\bar{\mathbf{x}}_i + \mathbf{Q}_i \boldsymbol{\theta}_j^{(i)}) \right\|^2 \end{aligned} \tag{1}$$

where \mathbf{Q} is an orthonormal basis matrix of tangent space, and the solution of Eq. (1) is $\bar{\mathbf{x}}_i = \frac{1}{k_i} \sum_{j=1}^{k_i} \mathbf{x}_{ij}$ and $\boldsymbol{\theta}_j^{(i)} = \mathbf{Q}_i^T (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)$, which are the local coordinates of \mathbf{x}_{ij} . The above procedure is actually a local principal component analysis.

3. Aligning local coordinates: to preserve more geometry in low-dimensional feature space as possible. The error of reconstruction from high-dimensional manifold space to low-dimensional feature space is minimized, i.e.

$$\min_{T, T^T = \mathbf{I}} \sum_{i=1}^l \min_{\substack{c_i \in \mathbb{R}^d \\ L \in \mathbb{R}^{d \times d}}} \frac{1}{k_i} \left\| \mathbf{t}_{ij} - (c_i + \mathbf{L}_i \boldsymbol{\theta}_j^{(i)}) \right\|^2 \tag{2}$$

where $T = [t_1, \dots, t_l] \in \mathbb{R}^{d \times l}$ is the objective global coordinates. Furthermore, Eq. (2) can be transformed into the following eigenvalue problem with algebra processing.

$$\min_{T^T T = \mathbf{I}} \text{tr}(\boldsymbol{\phi}^T) \tag{3}$$

where the alignment matrix $\boldsymbol{\phi} = \sum_{i=1}^l \frac{1}{k_i} \mathbf{S}_i \boldsymbol{\phi}_i \mathbf{S}_i^T$ is symmetric semidefinite. \mathbf{S}_i is the 0–1 selection matrix which computed by $T \mathbf{S}_i = [t_{i1}, \dots, t_{ik_i}] = \mathbf{T}_i$. $\boldsymbol{\phi}_i$ is an orthogonal projection.

Linear local tangent space alignment

Although LTSA can learn from the manifold space, it faces the problem that it is only available on the training set. Thus, the LLTSA method is proposed to solve the problem.

LLTSA is aimed at finding out a dimensionality reduction mapping available on both the training set and test set.

$$\mathbf{y}_i = \mathbf{A}^T \mathbf{x}_i \tag{4}$$

Considering the mapping (4), the objective function of LLTSA can be computed by the following problem:

$$\mathbf{A}_{\text{opt}} = \arg \min_{\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A} = \mathbf{I}} \text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{B} \mathbf{H} \mathbf{X}^T \mathbf{A}) \tag{5}$$

where $\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A} = \mathbf{I}$ is a constraint, $\mathbf{B} = \mathbf{S} \mathbf{W} \mathbf{W}^T \mathbf{S}^T$, $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_l]$, \mathbf{S}_i is the 0–1 selection matrix which computed by $\mathbf{Y} \mathbf{S} = \mathbf{Y}_i$, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_l]$ and \mathbf{y}_i is the global coordinates. And $\mathbf{W} = \text{diag}(\mathbf{W}_1, \dots, \mathbf{W}_l)$ with $\mathbf{W}_i =$

$\mathbf{H}_k (\mathbf{I} - \mathbf{V}_i \mathbf{V}_i^T)$. $\mathbf{H}_k = \mathbf{I} - e e^T / k$ represents the centering matrix, and \mathbf{V}_i is the matrix of d right singular vector of $\mathbf{X}_i \mathbf{H}_k$ corresponding to its d largest singular values.

The object function (5) is easy to resolve by applying Lagrange multiplier methods, and the function is transformed into a generalized eigenvalue problem:

$$\mathbf{X} \mathbf{H} \mathbf{B} \mathbf{H} \mathbf{X}^T \mathbf{a} = \lambda \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{a} \quad (6)$$

Then, the solution of (6) $\mathbf{a}_1, \dots, \mathbf{a}_d$ ordered corresponding to the size of the eigenvalues, $\lambda_1, \dots, \lambda_d$, so the mapping computed by the above process can be expressed as $\mathbf{A}_{\text{LLTSA}} = (\alpha_1, \dots, \alpha_d)$. But because the PCA is used to avoid singularity, the ultimate transformation matrix is:

$$\mathbf{A} = \mathbf{A}_{\text{PCA}} \mathbf{A}_{\text{LLTSA}} \quad (7)$$

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 expression 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.

Thus, there are many improvements for autoencoder, which include Stacked Autoencoder [39, 40], Sparse Autoencoder [41], Convolutional Autoencoder [42], Variational Autoencoder [43], etc.

For autoencoder, if the number of hidden layer nodes is less than the number of input layer nodes, it is called the undercomplete model; otherwise, it is called the overcomplete model. If the activation function of the hidden layer is linear, the autoencoder is called linear autoencoder. In this paper, the proposed method is based on the linear, undercomplete autoencoder with only one hidden layer.

Linear local tangent space alignment with autoencoder

Framework

In his section, the framework of the new LLTSA is demonstrated as follows. The new LLTSA method can be divided into two stages based on the structure of the encoder–decoder paradigm.

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

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

$$\hat{\mathbf{x}}_i = \mathbf{A}^* \mathbf{y}_i$$

To simplify the model, the weights of encoder and decoder in autoencoder can be tied as introduced in Ref. [44], i.e., $\mathbf{A}^* = (\mathbf{A}^T)^T = \mathbf{A}$. Thus, the decoding process may be rewritten as:

$$\hat{\mathbf{x}}_i = \mathbf{A} \mathbf{y}_i$$

Above all, the reconstruction error between original manifold data \mathbf{x}_i and the reconstruction data $\hat{\mathbf{x}}_i$ built by autoencoder is supposed to be minimized, which makes the low-dimensional embedded point “represents” the original sample more accurately and effectively. Figure 1 shows the architecture of the proposed LLTSA-AE method.

As described above, the proposed LLTSA-AE method is divided into two stages. The conventional LLTSA is regarded as the first stage. The second stage is to decode the low-dimensional data to the high-dimensional data space. And the reconstruction error between the original data and the reconstruction data is supposed to be minimized.

Compared with the LLTSA and other modified methods of LLTSA such as ALLTSA, OLLTSA, and WLLTSA, the proposed LLTSA-AE not only consider the error from high-dimensional space to low-dimensional space but also the reconstruction error between reconstructed space and original space, which is regarded as the decoding stage of the autoencoder. That is, our method obtains low-dimensional data by considering two-way mapping, rather than one-way mapping like traditional LLTSA algorithm and other modified methods of LLTSA. Thus, the proposed LLTSA-AE can “represent” the original samples more accurately and effectively. However, the original LLTSA algorithm obtains the low-dimensional embedding by considering minimizing the error from high-dimensional space to low-dimensional space. Besides, other modified methods of LLTSA also only concentrate on optimizing the process of the mapping from

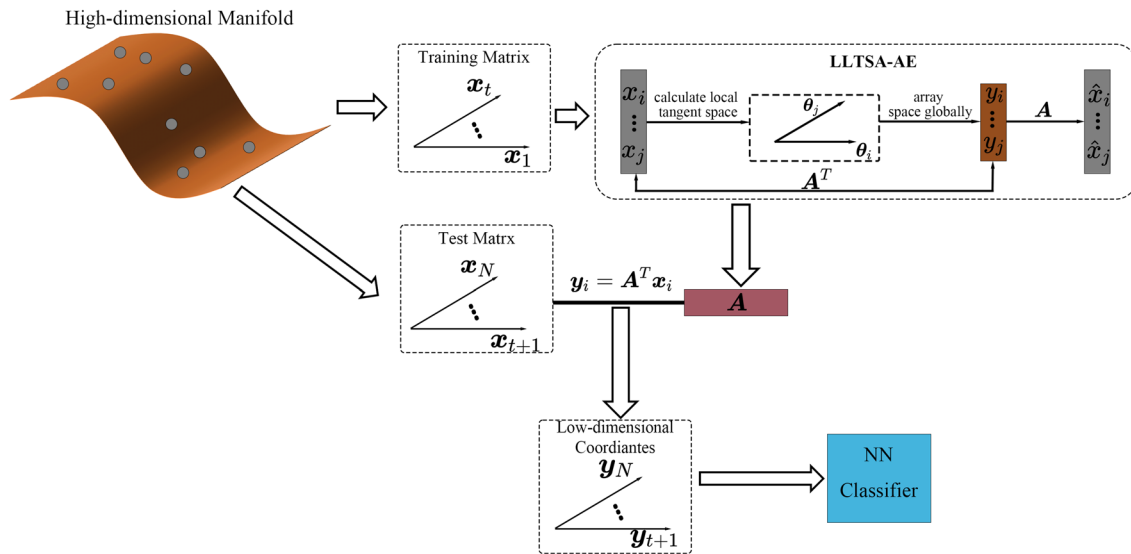


Fig. 1 The architecture of the proposed LLTSA-AE method

high-dimensional space to low-dimensional space. For example, ALLTSA tries to obtain better neighborhood number k to compute the better mapping by minimizing the error from high-dimensional space to low-dimensional space. OLLTSA is to eliminate the redundant information of the original manifold by taking the constraint of the basis vector on the orthogonal form. And WLLTSA enhances the mapping from high-dimensional space to low-dimensional space by replacing the original PCA algorithm with the weight PCA algorithm. Thus, the proposed LLTSA-AE is superior to the conventional LLTSA and other modified algorithms.

Based on such idea, the new method is named LLTSA-AE (Linear Local Tangent Space Alignment with Autoencoder).

The objective function

To implement the LLTSA-AE algorithm, the objective function of the LLTSA-AE is proposed in this section.

The first stage of LLTSA-AE is the conventional projection model of LLTSA and preserves the local neighborhood information of the samples. It can be formulated as minimizing Eq. (5). In this paper, Eq. (5) is used as the first item of the objective function of the new method, i.e.,

$$\mathcal{L}_{1\text{st}}(\mathbf{A}) = \text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{B} \mathbf{H} \mathbf{X}^T \mathbf{A}) \tag{8}$$

And the original constraint is also imposed on Eq. (8), i.e., $\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A} = \mathbf{I}$ can be relaxed as:

$$\mathcal{L}_{2\text{nd}}(\mathbf{A}) = \text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A}) - d \tag{9}$$

where, the parameter d is the dimensionality number of the target low-dimensional space, and the description is presented in “Local tangent space alignment”.

The second stage of LLTSA-AE is to reconstruct the original data and minimize the error between the original data and the reconstructed data. And the objective function of this stage denotes as follows:

$$\mathcal{L}_{3\text{rd}}(\mathbf{A}) = \text{tr} \left(\left(\mathbf{I} - \mathbf{A} \mathbf{A}^T \right) \mathbf{X} \mathbf{X}^T \left(\mathbf{I} - \mathbf{A} \mathbf{A}^T \right)^T \right) \tag{10}$$

Finally, combine Eqs. (8), (9), and (10), the objective function of LLTSA-AE algorithm is obtained. And the new method is to find the optimal projection matrix \mathbf{A} by minimizing the objective function:

$$\begin{aligned} \mathcal{L}(\mathbf{A}) &= \mathcal{L}_{1\text{st}}(\mathbf{A}) + \lambda \mathcal{L}_{2\text{nd}}(\mathbf{A}) + \gamma \mathcal{L}_{3\text{rd}}(\mathbf{A}) \\ &= \text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{B} \mathbf{H} \mathbf{X}^T \mathbf{A}) + \lambda (\text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A}) - d) \\ &\quad + \gamma \text{tr} \left(\left(\mathbf{I} - \mathbf{A} \mathbf{A}^T \right) \mathbf{X} \mathbf{X}^T \left(\mathbf{I} - \mathbf{A} \mathbf{A}^T \right)^T \right) \end{aligned} \tag{11}$$

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

Justification

As the first stage is the conventional LLTSA algorithm, the justification of Eqs. (8) and (9) is not discussed here and it can be viewed in detail in [27].

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

$$\sum_{i=1}^l \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \tag{12}$$

The reconstruction point \hat{x}_i can be further expressed as follows by considering Eq. (4), the linear mapping of LLTSA:

$$\hat{x}_i = \mathbf{A}y_i = \mathbf{A}\mathbf{A}^T x_i \quad (13)$$

Besides, Eq. (12) is the third part of the objective function of LLTSA-AE and it can be rewritten as:

$$\begin{aligned} \mathcal{L}_{3\text{rd}}(\mathbf{A}) &= \sum_{i=1}^l \left\| x_i - \mathbf{A}\mathbf{A}^T x_i \right\|_2^2 \\ &= \sum_{i=1}^l \left\| (\mathbf{I} - \mathbf{A}\mathbf{A}^T) x_i \right\|_2^2 \\ &= \text{tr} \left((\mathbf{I} - \mathbf{A}\mathbf{A}^T) \mathbf{X}\mathbf{X}^T (\mathbf{I} - \mathbf{A}\mathbf{A}^T)^T \right) \end{aligned} \quad (14)$$

It is worth mentioning that $\|\mathbf{A}\|_F^2$ regularization has not been considered in our model. It is unnecessary because the weights of the encoder and decoder are tied, i.e., $\mathbf{A}^* = \mathbf{A}$. If the norm $\|\mathbf{A}\|_F^2$ 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 \mathbf{A} , bad reconstruction will be produced. That is, the $\|\mathbf{A}\|_F^2$ regularization has been automatically handled by the reconstruction constraints [44].

Optimization

The formulation of LLTSA is linear and can be transformed into a generalized eigenvalue problem. However, the objective function of LLTSA-AE is non-linear and hard to solve directly. Thus, the stochastic gradient descent with momentum algorithm is employed to obtain the optimal matrix \mathbf{A} .

And the algorithm mainly includes three steps:

1. Calculate the gradient of the objective function (15)

$$\nabla \mathcal{L}(\mathbf{A}_t) = \nabla \mathcal{L}_{1\text{st}}(\mathbf{A}_t) + \lambda \nabla \mathcal{L}_{2\text{nd}}(\mathbf{A}_t) + \gamma \nabla \mathcal{L}_{3\text{rd}}(\mathbf{A}_t). \quad (15)$$

Where t is the number of iterations,

$$\begin{aligned} \nabla \mathcal{L}_{1\text{st}}(\mathbf{A}_t) &= 2\mathbf{X}_{i_t} \mathbf{H}_{i_t} \mathbf{B}_{i_t} \mathbf{H}_{i_t} \mathbf{X}_{i_t}^T \mathbf{A}_t, \\ \nabla \mathcal{L}_{2\text{nd}}(\mathbf{A}_t) &= 2\mathbf{X}_{i_t} \mathbf{H}_{i_t} \mathbf{X}_{i_t}^T \mathbf{A}_t, \\ \nabla \mathcal{L}_{3\text{rd}}(\mathbf{A}_t) &= -4 \left(\mathbf{I} - \mathbf{A}_t \mathbf{A}_t^T \right) \mathbf{X}_{i_t} \mathbf{X}_{i_t}^T \mathbf{A}_t. \end{aligned}$$

Where $i_t \in [1, n]$ represents the sample number randomly selected according to uniform distribution in the iteration.

2. Calculate the gradient of historical accumulation by the following formula:

$$\mathbf{v}_t = \rho \mathbf{v}_t + \alpha \nabla \mathcal{L}_{i_t}(\mathbf{A}_t) \quad (16)$$

Where $\rho \in (0, 1)$ is the coefficient of momentum, and its default value is 0.9 in the experiment. α is the learning rate, which is default set to $5 * 10^{-3}$ in the next experiment.

3. With the historical accumulation gradient, update the matrix \mathbf{A} using the following formula until the optimal matrix \mathbf{A} is found:

$$\mathbf{A}_{t+1} = \mathbf{A}_t - \mathbf{v}_t \quad (17)$$

1. The algorithm of LLTSA-AE is summarized in Algorithm 1.

Algorithm 1 the stochastic gradient descent with momentum method for training the optimal matrix \mathbf{A}

- 1: **Input:** The data matrix \mathbf{X} , the parameters λ, γ .
- 2: **Initialization:** Initialize \mathbf{A} randomly, \mathbf{v} as 0. Set the coefficient of momentum ρ to 0.9, set the threshold ϵ to 0.01 for the change of loss function, and set learning rate α to $5 * 10^{-3}$.
- 3: **Pre – calculation:** Calculate the weight matrix \mathbf{B} , the matrix \mathbf{H} .
- 4: **repeat**
- 5: Select the training sample \mathbf{X}_{i_t} and the matrix $\mathbf{B}_{i_t}, \mathbf{H}_{i_t}$ of this iteration.
- 6: Calculate the gradient $\nabla \mathcal{L}(\mathbf{A}_t)$ using Eq. (15)
- 7: Calculate the gradient of historical accumulation \mathbf{v}_t using Eq. (16)
- 8: Update the matrix \mathbf{A}_{t+1} using Eq. (17)
- 9: Calculate the objective function $\mathcal{L}(\mathbf{A}_{t+1})$ using Eq. (11)
- 10: **until** the change of objective function $< \epsilon$
- 11: **Output:** the optimal projection matrix \mathbf{A} .

After the above algorithm, the optimal projection matrix \mathbf{A} is obtained. And the solution matrix \mathbf{A} not only satisfies the formula of conventional LLTSA but also takes the reconstruction stage into consideration. Thus, the low-dimensional data point y_i computed by the linear mapping $y_i = \mathbf{A}^T x_i$ can “represent” the original high-dimensional data point x_i accurately.

The parameter analysis

Two free parameters λ and γ (see Eq. (11)) are contained in the proposed LLTSA-AE algorithm.

In Eq. (11), if $\gamma = 0$, Eq. (11) is reduced to

$$\mathcal{L} = \text{tr} \left(\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{B} \mathbf{H} \mathbf{X}^T \mathbf{A} \right) + \lambda \left(\text{tr} \left(\mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A} \right) - d \right) \quad (18)$$

Equation (18) is actually another way to solve the objective function of conventional LLTSA, because the reconstruction formula is no longer considered in it. And the aim of LLTSA is to find out the eigenvectors corresponding to the minimum eigenvalues. Therefore, the parameter λ is taken smaller values and it is validated by our next experiments.

In Eq. (11), the first item \mathcal{L}_{1st} can be regarded as 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.

Experimental results

In this Section, we make experiments on the Handwritten Alphadigits dataset, FERET (Face Recognition Technology) dataset, GT (Georgia Tech) face dataset, and Yale face dataset. The proposed LLTSA-AE is compared with LPP, NPE, LLTSA, ALLTSA, OLLTSA, PLLTSA, and WLLTSA respectively.

For the LPP, NPE, LLTSA, and the improved methods of LLTSA, PCA needs to be used to reduce the dimension of the original samples. For all datasets, 98% of the principal components are retained in LPP and NPE, and 70% in LLTSA and other improved methods of LLTSA in this step.

The accuracy in the experiment can be described as:

$$ACC = \frac{N^{CC}}{N}$$

Where N is the total number of classified subjects, N^{CC} is the total number of correctly classified subjects, and ACC is the accuracy [45]. Accuracy is widely used in scientific research. The advantage of the index is intuitive and clear, and the performance of the model can be viewed at a glance. Its disadvantage is that it may not perform well on unbalanced data sets. However, the data sets used in this paper are all balanced data sets, so the accuracy is the optimal index, which can not only well represent the performance of the model, but also make the performance more intuitive and clearer.

With Matlab's powerful matrix computing power and rich toolbox resources, we program on Intel i7-8750H CPU and 16GB RAM computer based on MatlabR2021a. Besides, the optimization techniques of our algorithm refer to an easy-to-use optimization library: SGDLibrary [46]. The SGDLibrary is a pure-MATLAB library or toolbox of a collection of stochastic optimization algorithms. The library contains many optimization algorithms, for example, the traditional SGD and SGD with classical momentum. Besides, other optimization algorithms such as Adam and RMSProp are also implemented in the library.

Experimental datasets

Handwritten Alphadigits dataset. This dataset consists of 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.

FERET face dataset [47] was constructed by the Army Research Laboratory. There are up to 200 human subjects and each person contains 7 images. Notably, the photos of the same person vary in expression, lighting, posture, and age. In the experiment, the size of the images is also adjusted to 32×32 pixels.

GT face dataset [48] contains 50 persons' face images taken between 06/01/99 and 11/15/99. And there are 15 images of each people. The images contain frontal and tilted faces with different facial expressions, lighting conditions, and scales. In the experiment, the images are resized to 32×32 pixels.

Yale face dataset [49] was made at the Yale Center for Computational Vision and Control. And it contains 165 grayscale images of 15 individuals, with 11 samples of each individual. The images contain variations in facial expressions, lighting, and with/without glasses. In this experiment, all images are aligned based on eye coordinates and are cropped and scaled to 24×24 .

Some sample images of the Handwritten Alphadigits, FERET, GT, and Yale are shown in the following Fig. 2.

Parameter settings

In this section, on four datasets, for the LLTSA-AE method, we make experiments to find the optimal configuration of the parameters λ and γ . The range of the parameters λ and γ depends on experience and the algorithm is an iterative process. At the beginning of the iteration, the values of each item in the objective function are larger, and after multiple iterations, the values of each item in the objective function are getting smaller. Therefore, after comprehensive consideration, we have taken a relative compromise range of [0, 50]. This range (described as the long range) is used to find the trend of the parameters, and then an optimal short range is found. I.e., λ and γ are separately found on the short-range [0, 4] and [0, 6] to obtain the optimal values. In the experiment, the training samples of 4 datasets, Handwritten Alphadigits, FERET, GT, and Yale, are 9, 5, 11, and 8, respectively. The neighborhood size parameter $k = 10$, and the subspace dimension is 40.

When the parameter γ is fixed, the recognition rate curves w.r.t the parameter λ in the four datasets are shown in Figs. 3a and 4a. When the parameter λ is fixed, the recognition rate curves w.r.t the parameter γ in the four datasets are shown in Figs. 3b and 4b. According to Figs. 3 and 4, their manifestations are as: (1) for parameter λ , the changing trend of its recognition accuracy is weak in the range [0, 3], and then it declined obviously after 3. (2) for parameter γ , the recognition accuracy increases with the increase of γ within



Fig. 2 Some samples in the experiments. The first line is the Handwritten Alphadigits dataset. The second line is the FERET dataset. The third line is the GT dataset. The fourth line is the Yale dataset

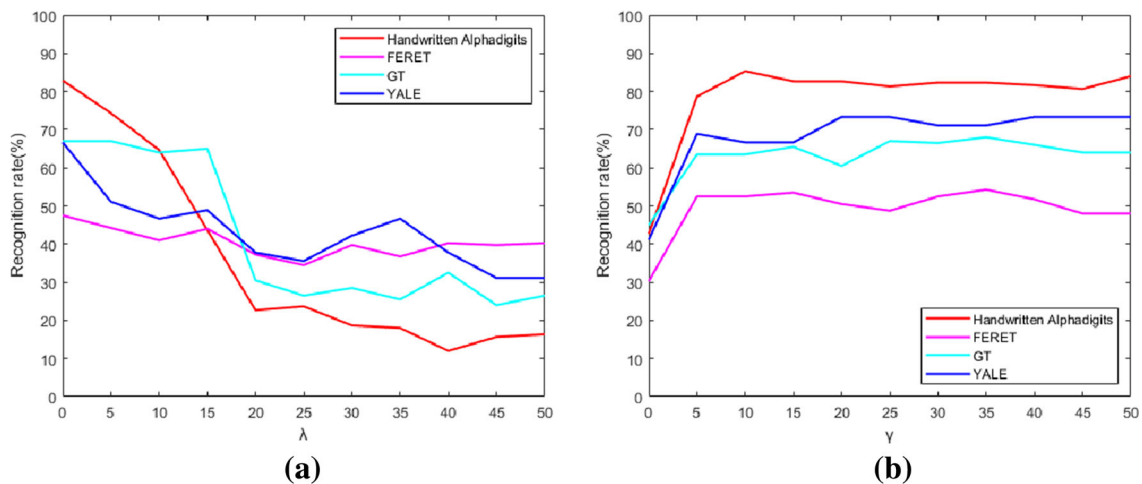


Fig. 3 The recognition accuracy in four datasets w.r.t the parameter λ (γ) while γ (λ) fixed within long ranges

the range of $[0, 2]$. And then the recognition accuracy keeps steady after 2. Thus, We can conclude that: (1) the optimal value of the parameter λ is usually within the smaller values and begins to decline steadily and substantially after 3. (2) the optimal value of parameter γ is usually greater than 2, and its recognition rate tends to be stable after it is greater than 2. Therefore, we choose to conduct a grid search for two parameters between $[0, 4]$ and $[0, 6]$, which can completely

cover the optimal parameter values obtained in the experiment and reduce meaningless search operations at the same time.

According to the recognition rate curve in Fig. 4, we can get the optimal configuration of the parameters λ and γ , which are listed in Table 1. The optimal parameter configuration is used for the following experiments.

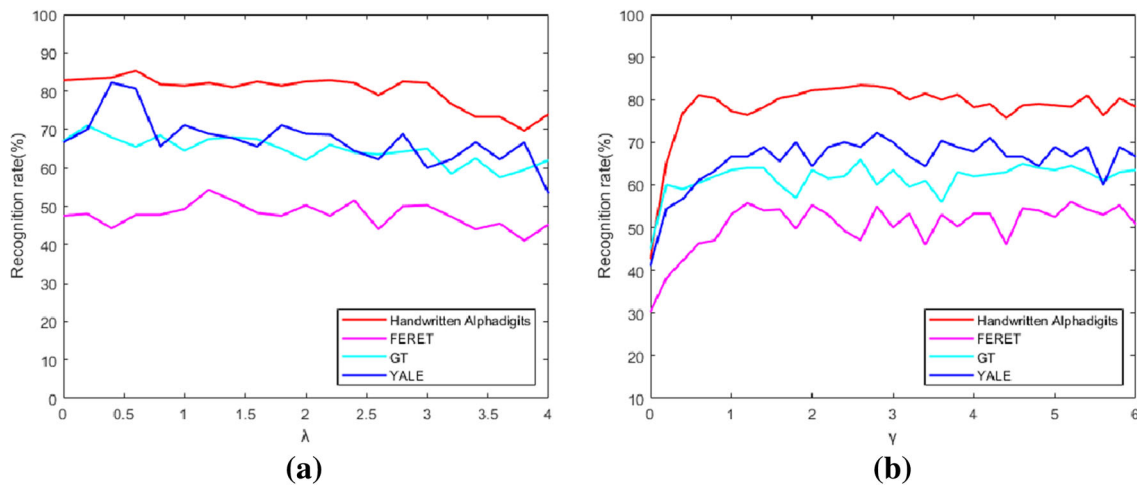


Fig. 4 The recognition accuracy in four datasets w.r.t the parameter λ (γ) while γ (λ) fixed within short ranges

Table 1 The parameter configuration of the LLTSA-AE method on four experimental datasets

Method	Handwritten Alphanumerals		FERET	
	λ	γ	λ	γ
LLTSA-AE	0.6	2.6	1.2	1.2
	GT		Yale	
	λ	γ	λ	γ
LLTSA-AE	0.2	2.6	0.4	2.8

The experiments are proceeded as follows: Firstly, p samples of each subject are selected randomly to form as the training set, and the rest of the samples are used as the test dataset. Secondly, fix p , the reduced dimensions are taken from [10, 100] at the step size 5. Thirdly, fix p and the reduced dimension value, the neighborhood size k is augmented from 5 to 25 with the interval of 5. Fourthly, the best recognition rate corresponding to the best k value is used as the recognition rate in the current p and reduced dimension. Finally,

for each of the reduced dimension and p , we can get the corresponding recognition rate.

We regard the above process as a cycle. For a given p , 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 in the current p and subspace dimensions. Finally, we take the best recognition rate from the chosen reduced dimensions as the final result of the training samples p .

Experimental results

In the four datasets, for the above eight methods, the best recognition rates, standard deviation, and the optimal dimension are reported as follows.

We first evaluate the performance of LLTSA-AE on the Handwritten Alphanumerals dataset which contains 26 capital letters “A” through “Z”. And the results is shown in Table 2 and Fig. 5.

As is shown in Table 2, LLTSA-AE achieves the highest accuracy compared with other methods. To be specific, the accuracy of LLTSA-AE exceeds that of the original LLTSA by 10.68, 9.65, and 9.4%. And it can be seen that with the

Table 2 Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on the Handwritten Alphanumerals dataset

Method	5 trains	7 trains	9 trains
LPP	66.29 ± 3.13(15)	68.81 ± 2.03(15)	72.13 ± 2.13(25)
NPE	70.41 ± 2.01(55)	75.31 ± 2.78(35)	78.46 ± 2.90(35)
LLTSA	67.26 ± 3.36(10)	72.00 ± 2.16(15)	75.70 ± 2.30(15)
ALLTSA	71.47 ± 3.44(10)	76.88 ± 3.5(10)	80.33 ± 2.13(20)
OLLTSA	72.25 ± 2.60(20)	78.32 ± 2.99(30)	81.14 ± 2.42(25)
PLLTSA	70.51 ± 2.43(20)	75.53 ± 3.10(25)	77.30 ± 2.56(20)
WLLTSA	72.47 ± 2.90(10)	75.03 ± 2.32(10)	76.50 ± 1.51(20)
LLTSA-AE	77.94 ± 2.52(80)	81.65 ± 1.35(95)	85.10 ± 1.19(100)

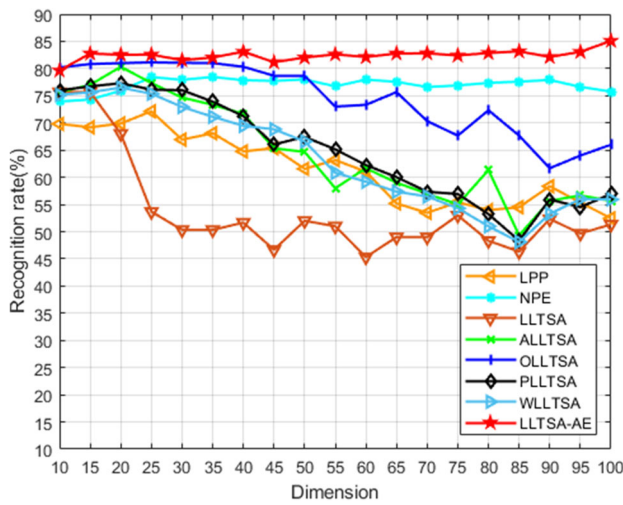


Fig. 5 The recognition rates versus the subpace dimension on the Handwritten Alphadigits dataset (9 training samples)

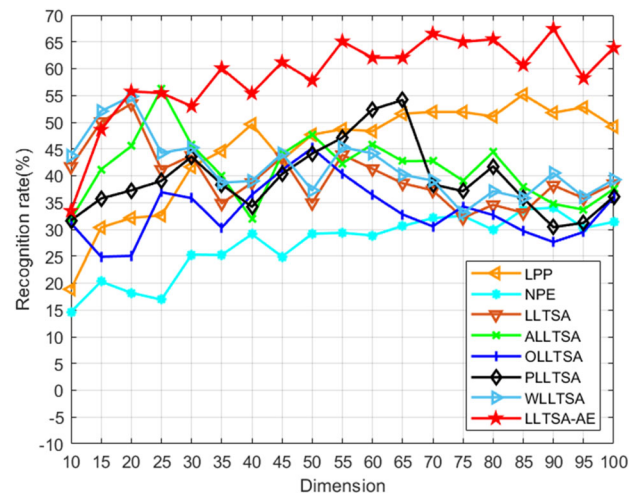


Fig. 6 The recognition rates versus the subpace dimension on the FERET dataset (5 training samples)

dimension increases, the accuracy of LLTSA-AE goes up steadily. While other methods such as LLTSA and LPP get lower accuracy at higher dimensions.

Then the FERET dataset with 3, 4 and 5 training samples are used to evaluate the algorithm.

According to the Table 3, LLTSA-AE also obtains the best performance. The recognition rates of LLTSA-AE are 53.50, 65.13 and 67.45%, which are higher than LLTSA 10.2, 20.63, and 14.03% separately. And Fig. 5 also shows that our proposed method can extract more representative information at

higher dimensions. It's worth explaining that our proposed method performs not so well at lower dimensions compared with some other methods, and this mainly because the bigger encoding rate by LLTSA algorithm may influence the reconstruction of the decoding stage.

And the experiment results on GT dataset is shown in following Table 4 and Fig. 7.

Based on the Table 4 and Fig. 7, the recognition rate of LLTSA-AE is much higher than the conventional LLTSA algorithm. As shown in the Fig. 6, the accuracy of LLTSA-AE

Table 3 Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on the FERET dataset

Method	3 trains	4 trains	5 trains
LPP	43.94 ± 8.07(90)	49.58 ± 9.40(100)	55.17 ± 5.34(85)
NPE	25.18 ± 7.30(90)	29.94 ± 7.98(100)	34.02 ± 11.24(90)
LLTSA	43.30 ± 12.20(20)	44.50 ± 11.02(20)	53.42 ± 7.08(20)
ALLTSA	44.37 ± 8.35(25)	46.89 ± 6.83(20)	56.2 ± 10.37(25)
OLLTSA	37.24 ± 6.20(70)	39.91 ± 4.34(55)	45.15 ± 3.30(50)
PLLTSA	44.07 ± 5.17(25)	45.63 ± 2.73(55)	54.15 ± 3.31(65)
WLLTSA	44.40 ± 6.92(20)	46.63 ± 4.24(25)	54.83 ± 5.74(20)
LLTSA-AE	53.50 ± 6.28(95)	65.13 ± 5.35(100)	67.45 ± 2.94(90)

Table 4 Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on the GT dataset

Method	7 trains	9 trains	11 trains
LPP	53.97 ± 0.93(60)	59.03 ± 1.66(70)	65.45 ± 2.89(80)
NPE	51.55 ± 2.35(95)	54.23 ± 2.91(100)	56.95 ± 3.70(70)
LLTSA	62.25 ± 0.67(35)	67.73 ± 2.46(25)	68.05 ± 3.38(25)
ALLTSA	65.62 ± 2.45(55)	68.56 ± 3.60(60)	70.21 ± 2.18(30)
OLLTSA	64.87 ± 1.98(50)	68.59 ± 2.26(55)	69.90 ± 4.79(35)
PLLTSA	63.26 ± 1.98(50)	69.36 ± 1.68(50)	69.70 ± 2.43(60)
WLLTSA	63.92 ± 1.60(15)	68.50 ± 1.23(20)	70.00 ± 4.64(20)
LLTSA-AE	69.02 ± 2.38(90)	71.5 ± 1.1(100)	75.40 ± 2.15(90)

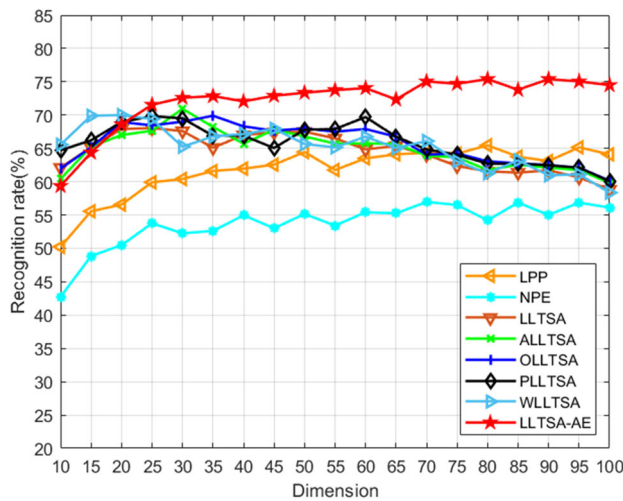


Fig. 7 The recognition rates versus the subspace dimension on the GT dataset (11 training samples)

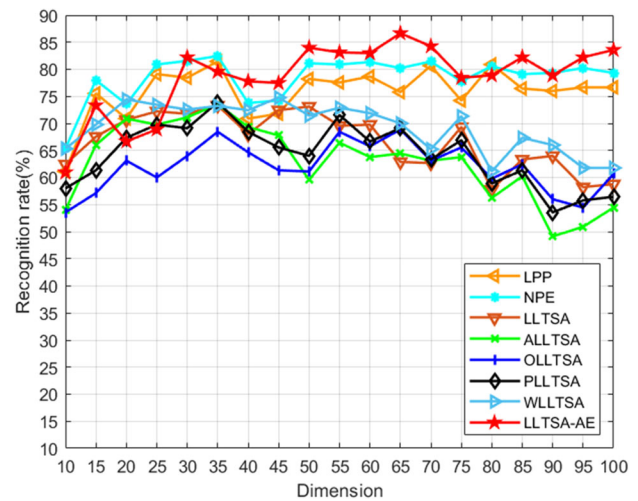


Fig. 8 The recognition rates versus the subspace dimension on the Yale dataset (8 training samples)

AE is still steadily improved in high dimensions, while other methods show a fluctuating trend.

Finally, the Yale face dataset is employed to test our proposed algorithm and the result is shown in the following Table 5 and Fig. 8.

On the Yale face dataset, LLTSA-AE still obtains the highest recognition rate compared with all other methods. In detail, the recognition rates of LLTSA-AE are 76.19, 83.02 and 86.67% which is higher than LLTSA 11.05, 10.89 and 12.39%.

The convergence of the proposed LLTSA-AE is also investigated. The convergence curves of LLTSA-AE on four datasets are presented in Fig. 9. Where, the training samples are 9, 5, 11 and 8 in the Handwritten Alphanumerals, FERET, GT, and YALE datasets respectively, the neighborhood size parameter $k = 10$, and the subspace dimension is 40. As can be seen from Fig. 9, LLTSA-AE can converge in only 30 steps on four datasets. Besides, we also simply investigate the computation time of our algorithm. We compute the average time 10 times under the same parameter settings as the above experiment on the four datasets. The time for each compu-

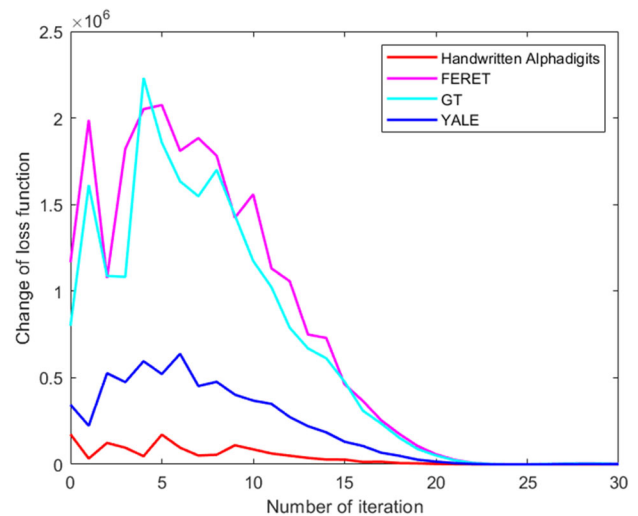


Fig. 9 The convergence curve of LLTSA-AE

tation and classification on the four datasets: Handwritten Alphanumerals, FERET, GT, and YALE datasets is 0.13, 2.13, 2.51, and 0.61 s separately. Since only one training process

Table 5 Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on the Yale dataset

Method	4 trains	6 trains	8 trains
LPP	71.23 ± 4.69(30)	75.86 ± 4.55(45)	81.44 ± 9.08(35)
NPE	73.04 ± 3.94(100)	78.13 ± 5.14(45)	82.44 ± 9.42(35)
LLTSA	65.14 ± 5.3(50)	72.13 ± 7.90(25)	74.28 ± 5.30(50)
ALLTSA	66.66 ± 8.21(20)	73.06 ± 8.01(15)	73.55 ± 6.20(35)
OLLTSA	65.23 ± 5.27(30)	66.93 ± 3.61(50)	68.88 ± 6.20(35)
PLLTSA	65.84 ± 7.96(30)	72.52 ± 5.99(20)	73.91 ± 5.10(35)
WLLTSA	66.47 ± 8.16(20)	73.60 ± 5.84(20)	74.44 ± 5.31(30)
LLTSA-AE	76.19 ± 3.75(90)	83.02 ± 4.51(75)	86.67 ± 5.30(75)

is required to get the mapping in real-world applications, the computation time is perfectly acceptable.

Analysis

From the above experiment, it can be concluded that: (1) as an improved method of LLTSA, the performance of LLTSA-AE has a great improvement on all datasets. Specifically, compared with the conventional LLTSA, LLTSA-AE has about 9, 14, 7, and 12% improvement in recognition rate on 4 datasets (Handwritten Alphadigits, FERET, GT, YALE), respectively. (2) moreover, LLTSA-AE is also compared with LPP, NPE, LLTSA, and the improved methods of LLTSA, for example, ALLTSA and OLLTSA. Compared with them, LLTSA-AE has different degrees of improvement, and the performance of LLTSA-AE is the best. (3) LLTSA-AE performs better at the high dimensions especially. And this is mainly because lower dimensions mean higher encoding rates, which may lose more information and it is hard to recover the original information accurately. Besides, this influences the reconstruction stage a lot which causes the lower recognition rate. (4) as can be seen from Figs. 4, 5, 6 and 7, the recognition rates of LLTSA-AE significantly outperform that of the other methods, which shows that LLTSA-AE has the best performance.

Conclusion

LLTSA and all its variants only consider one-way mapping from high-dimensional space to low-dimensional space, which may result in that projected low-dimensional data may not effectively “represent” the original data. In this paper, a novel LLTSA method called LLTSA-AE (linear local tangent space alignment with autoencoder) based on the encoder–decoder paradigm is proposed. The proposed LLTSA-AE is aiming at obtaining an optimal linear mapping from high-dimensional space to low-dimensional space by considering two-way mapping between high-dimensional space and low-dimensional space. And it makes the obtained low-dimensional data represent the original samples accurately and effectively. The main idea of LLTSA-AE is to take the conventional projection of LLTSA as the encoding stage and use the decoder to reconstruct the original data from the projected low-dimensional data. Compared with LLTSA, LLTSA-AE makes that the low-dimensional features more accurately and effectively “represent” the original samples with the reconstruction stage. The experiments on the Handwritten Alphadigits, FERET, GT, and Yale datasets show that LLTSA-AE obtains the optimal performance rate compared with other methods.

The idea of LLTSA-AE can be further extended to other manifold learning methods, such as neighborhood preserv-

ing embedding (NPE). Besides, there is no conflict with other promotion algorithms which means this idea can be employed in other improved methods of LLTSA.

And the classical autoencoder is simple and easy to implement, but in the face of a complex learning system, its learning ability is limited. Thus, some improved algorithms of autoencoder are proposed. For example, the stacked autoencoder is proposed to improve the learning ability by increasing the number of layers. And sparse autoencoder is proposed to address the problem that the traditional autoencoder may lose the automatic learning ability when the node number of the hidden layer is too many. Thus, the autoencoder used in our algorithm can further extend to these improved autoencoders. Furthermore, extending the autoencoder to other neural networks and combined with LLTSA algorithm is also a theoretically feasible direction.

Declarations

Novelty statements Based on the encoder–decoder paradigm, a new LLTSA method is proposed in this paper. The new LLTSA method is called LLTSA-AE (Linear Local Tangent Space Alignment with Autoencoder). In this method, the conventional LLTSA projection is regarded as the encoding stage, and the decoder is used to reconstruct the original high-dimensional data from the projected low-dimensional data. Since LLTSA and all its variants only consider the one-way mapping from high-dimensional space to low-dimensional space. The projected low-dimensional data of LLTSA may not “represent” the original data accurately and effectively. Based on the encoder–decoder paradigm, the proposed LLTSA-AE method not only make the low-dimensional embedding can “represent” the original data more accurately and effectively, but also preserves the original manifold structure. Finally, the experimental results on databases such as Handwritten Alphadigits, FERET, Georgia Tech (GT), etc show that LLTSA-AE outperforms LLTSA and some of its representative variants.

Acknowledgements This work was supported by National Natural Science Foundation of China under grant 61876026, Humanities and Social Sciences Project of the Ministry of Education of China under grant 20YJAZH084, Chongqing Technology Innovation and Application Development Project under Grant cstc2020jscx-msxmX0190 and cstc2019jscxmbdxX0061, the Key Project for Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZD-K202100505.

Data availability The Handwritten Alphadigits dataset that support the findings of this study are available at <https://www.cs.nyu.edu/~protect/unhbox/voidb@x\penalty\Mroweis/data.html>. The FERET dataset that support the findings of this study are available at <https://www.nist.gov/programs-projects/face-recognition-technology-feret>. The GT dataset that support the findings of this study are available at http://www.anefian.com/research/face_reco.htm. The yale dataset that

support the findings of this study are available at <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>.

Declarations

Conflict of 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.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Jia W, Sun M, Lian J et al (2022) Feature dimensionality reduction: a review. *Complex Intell Syst* 1–31
- Van Der Maaten L, Postma E, Van den Herik J et al (2009) Dimensionality reduction: a comparative. *J Mach Learn Res* 10(66–71):13
- Shen L, Tao H, Ni Y et al (2023) Improved yolov3 model with feature map cropping for multi-scale road object detection. *Meas Sci Technol*
- Loey M, Manogaran G, Taha MHN et al (2021) Fighting against Covid-19: a novel deep learning model based on yolo-v2 with ResNet-50 for medical face mask detection. *Sustain Cities Soc* 65(102):600
- Zhuang Z, Tao H, Chen Y et al (2022) An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints. *IEEE Trans Syst Man Cybern Syst*
- Lin CY, Sun L, Tomizuka M (2015) Robust principal component analysis for iterative learning control of precision motion systems with non-repetitive disturbances. In: 2015 American control conference (ACC). IEEE, pp 2819–2824
- Keogh EJ, Mueen A (2017) Curse of dimensionality. *Encycl Mach Learn Data Min* 2017:314–315
- Verleysen M, François D (2005) The curse of dimensionality in data mining and time series prediction. In: International work-conference on artificial neural networks. Springer, pp 758–770
- Fisher RA (1936) The use of multiple measurements in taxonomic problems. *Ann Eugen* 7(2):179–188
- Cai D, He X, Han J (2007) Semi-supervised discriminant analysis. In: IEEE international conference on computer vision. IEEE, pp 1–7
- Turk M, Pentland A (1991) Eigenfaces for recognition. *J Cognit Neurosci* 3(1):71–86
- Yan S, Xu D, Zhang B et al (2006) Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans Pattern Anal Mach Intell* 29(1):40–51
- Gupta A, Barbu A (2018) Parameterized principal component analysis. *Pattern Recogn* 78:215–227
- Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. *Philos T R Soc A* 374(2065):20150202
- Erichson NB, Zheng P, Manohar K et al (2020) Sparse principal component analysis via variable projection. *SIAM J Appl Math* 80(2):977–1002
- Gisbrecht A, Hammer B (2015) Data visualization by nonlinear dimensionality reduction. *Wires Data Min Knowl* 5(2):51–73
- Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering
- Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326
- Tenenbaum JB, Silva VD, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323
- Zhang Z, Zha H (2004) Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J Sci Comput* 26(1):313–338
- Lee JA, Lendasse A, Verleysen M (2004) Nonlinear projection with curvilinear distances: isomap versus curvilinear distance analysis. *Neurocomputing* 57:49–76
- Sani ZA, Ahmad Shalbfaf HB, Shalbfaf R (2015) Automatic computation of left ventricular volume changes over a cardiac cycle from echocardiography images by nonlinear dimensionality reduction. *J Digit Imaging* 28(1):91
- Bengio Y, Paiement JF, Vincent P et al (2003) Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. 177–184
- Cai D, He X, Han J et al (2007) Isometric projection. In: Association for the advancement of artificial intelligence, pp 528–533
- He X, Niyogi P (2004) Locality preserving projections. *Adv Neural Inf Process Syst* 16:153–160
- He X, Cai D, Yan S et al (2005) Neighborhood preserving embedding. 1208–1213
- Zhang T, Yang J, Zhao D et al (2007) Linear local tangent space alignment and application to face recognition. *Neurocomputing* 70(7–9):1547–1553
- Jäntschi L (2019) The eigenproblem translated for alignment of molecules. *Symmetry* 11(8):1027
- Zhang Z, Wang J, Zha H (2011) Adaptive manifold learning. *IEEE Trans Pattern Anal Mach Intell* 34(2):253–265
- Lei YK, Xu YM, Yang JA et al (2012) Feature extraction using orthogonal discriminant local tangent space alignment. *Pattern Anal Appl* 15(3):249–259
- Feng L, Liu S, Xiao Y et al (2015) A novel CBIR system with WLLTSA and ULRGA. *Neurocomputing* 147:509–522
- WenHua L (2011) Modified linear local tangent space alignment algorithm. *J Comput Appl* 31(01):247
- Hassan Shah MZ, Hu L, Ahmed Z (2022) Weighted linear local tangent space alignment via geometrically inspired weighted PCA for fault detection. *IEEE Trans Ind Inform* 1–1
- Fang L, Lv Y, Ma L et al (2017) Improved linear local tangent space alignment and its application to pattern recognition. *Int J Comput Appl T* 56(3):244–52
- Zuqiang Su eBaoping Tang (2017) Rotating machinery fault diagnosis with supervised-linear local tangent space alignment for dimension reduction. *Chin J Sci Instrum* 35(3):244–252
- Lv YX, Deng YN, Shi Y et al (2014) Adaptive discriminant linear local tangent space alignment algorithm on face recognition. In: Advanced materials research. Trans Tech Publ, pp 2381–2384
- Li Y, Luo D, Liu S (2009) Orthogonal discriminant linear local tangent space alignment for face recognition. *Neurocomputing* 72(4–6):1319–1323
- Wang Y, Wang Z, Zhang G et al (2012) Face recognition using marginal discriminant linear local tangent space alignment. In: International conference on intelligent system design and engineering application. IEEE, pp 1418–1421
- Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507

40. Bengio Y et al (2009) Learning deep architectures for AI. *Found Trends Mach Leg* 2(1):1–127
41. Bengio Y, Lamblin P, Popovici D et al (2006) Greedy layer-wise training of deep networks
42. Masci J, Meier U, Cireşan D et al (2011) Stacked convolutional auto-encoders for hierarchical feature extraction. In: *International conference on artificial neural networks*. Springer, pp 52–59
43. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114)
44. Ranzato M, Boureau YL, Cun Y et al (2007) Sparse feature learning for deep belief networks
45. Bolboacă SD, Jäntschi L (2014) Sensitivity, specificity, and accuracy of predictive models on phenols toxicity. *J Comput Sci-Neth* 5(3):345–350
46. Kasai H (2017) Sgdlibrary: a matlab library for stochastic optimization algorithms. *J Mach Learn Res* 18(1):7942–7946
47. Phillips PJ, Moon H, Rizvi SA et al (2000) The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans Pattern Anal Mach Intell* 22(10):1090–1104
48. Nefian AV (1999) Georgia tech face database. Georgia Institute of Technology. http://www.anefian.com/research/face_reco.htm
49. Belhumeur P, Kriegman D (1997) The Yale face database. Yale University. 1(2):4. <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.