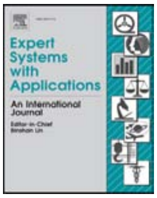
Expert Systems With Applications 151 (2020) 113352

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Locality adaptive preserving projections for linear dimensionality reduction

Aiguo Wang[a], Shenghui Zhao[b], Jinjun Liu[b], Jing Yang[c], Li Liu[d], Guilin Chen[b,*]

[a] School of Electronic Information Engineering, Foshan University, Foshan 528225, China
[b] School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China
[c] School of Computer and Information, Hefei University of Technology, Hefei 230009, China
[d] School of Big Data and Software Engineering, Chongqing University, Chongqing, 400044 China

ARTICLE INFO

Article history:
Received 23 June 2019
Revised 8 February 2020
Accepted 2 March 2020
Available online 4 March 2020

Keywords:
Dimensionality reduction
Feature extraction
Intrinsic dimensionality
Local structure

ABSTRACT

Dimensionality reduction techniques aim to transform the high-dimensional data into a meaningful reduced representation and have been consistently playing a fundamental role in the study of intrinsic dimensionality estimation and the design of an intelligent expert system towards real-world applications. From the perspective of manifold learning, locality preserving projections is a classical and commonly used dimensionality reduction method and it essentially learns the low-dimensional embedding under the constraint of preserving the local geometry of data. However, since it determines the neighborhood relationships in the original feature space that probably contains noisy and irrelevant features, the derived similarity between the neighbors are unreliable and the corresponding local data manifold tends to be error-prone, which inevitably leads to degraded performance for subsequent data analyses. Hence, how to accurately identify the true neighbor relationships for each sample remains crucial to the robustness improvement. In this work, we propose a novel approach, termed locality adaptive preserving projections (LAPP), to adaptively determine the neighbors and their relationships in the optimal subspace rather than in the original space. Specifically, due to the absence of prior knowledge of local properties of the underlying manifold, LAPP adopts a coarse-to-fine strategy to iteratively update the projected low-dimensional subspace and optimize the identification of the local structure of the data. Moreover, an iterative algorithm with fast convergence is utilized to solve the transformation matrix for explicit out-of-sample extension. Besides, LAPP is easy to implement and its key idea can be potentially extended to other methods for neighbor-finding and similarity measurement. To evaluate the performance of LAPP, we conduct comparative experiments on numerous synthetic and real-world datasets. Experimental results show that seeking the local structure in the original feature space misleads the selection of neighbors and the calculation of similarity and that the proposed method helps alleviate the negative effect of noisy and irrelevant features, which demonstrates its effectiveness. Besides, this study has the potential to enlighten relevant studies to consider the problem of optimizing the neighborhood relationships.

1. Introduction

For a variety of research fields and real-world applications that range from face recognition (He, Yan, Hu, Niyogi & Zhang, 2005) and smoke detection (Yuan, Xia, Shi, Li & Li, 2017) to activity recognition (Wang, Chen, Yang, Zhao & Chang, 2016) and finance management (Tayalı & Tolun, 2018; Zhong & Enke, 2017), we are often confronted with high-dimensional data and required to develop powerful analysis methods for the discovery of knowledge and the design of a decision support system, especially in the era of big data where we are faced with a massive amount of data that are characterized by complexity, variety, and high-dimensionality. Consequently, the prediction and evaluation models directly trained on such data not only suffer immensely from the curse of dimensionality, but also larger computational loads. Even worse, if the original feature space fails to reflect the intrinsic structure of the data, it leads to degraded performance and lowers the confidence of a decision system to a large extent (Qiao, Chen & Tan, 2010). For example, in terms of a face recognition system, we often organize a $w*h$ face image into a $w*h$ dimensional vector for

[*] Corresponding author at: Fengle Road No. 1528, Chuzhou University, Chuzhou 239000, China.

E-mail addresses: wangaiguo2546@163.com (A. Wang), zsh@chzu.edu.cn (S. Zhao), jinjunl@chzu.edu.cn (J. Liu), jsjyj0801@163.com (J. Yang), dcsliuli@cqu.edu.cn (L. Liu), glchen@chzu.edu.cn (G. Chen).

https://doi.org/10.1016/j.eswa.2020.113352
0957-4174/© 2020 Elsevier Ltd. All rights reserved.

appearance-based techniques, which is too large for robust face recognition (Bhowmik, Saha, Singha, Bhattacharjee & Dutta, 2019). In the task of building an intelligent expert system for daily stock market analyses, researchers usually collect a wide range of financial and economic features to maximize the stock market return. However, some of these features are irrelevant to the task and even redundant to each other (Zhong & Enke, 2017). Undoubtedly, this poses a serious challenge to the exploration of intrinsic dimensionality of the data, the efficiency of many machine learning models, and the generalization ability of a system for real-world scenarios. Accordingly, one common way to mitigate this problem is to utilize an effective and efficient dimensionality reduction method to reduce data dimensionality (Bhowmik et al., 2019; van der Maaten, Postma & Herik, 2009).

As an important preprocessing technique in data analysis, dimensionality reduction techniques basically work by transforming the data of high-dimensionality into a meaningful low-dimensional representation in a linear or non-linear way and they have been consistently playing a fundamental and important role in better revealing the intrinsic structure of the data and greatly facilitating the subsequent tasks (Zhao, Wang & Nie, 2018; Zhong & Enke, 2017). Particularly, dimensionality reduction contributes to the tasks of classification, regression, clustering, visualization, and data compression in a variety of applications such as face recognition, information retrieval, and disease diagnosis (Becht et al., 2019; van der Maaten & Hinton, 2008). For example, principle component analysis seeks a group of irrelevant variables by discarding redundant information and it helps reduce noise and improve the performance of a classifier. The key assumption behind dimensionality reduction is that the original feature space contains irrelevant features and some features are redundant to each other, we can then find a group of new features to represent the original ones (Tenenbaum, De Silva & Langford, 2000). Therefore, the task of dimensionality reduction is to find a reduced representation with the intrinsic dimensionality of the data by deriving an appropriately linear/non-linear transformation function under the carefully devised constraint conditions (van der Maaten et al., 2009; Zhao et al., 2018).

According to the requirement for the availability of data labels, existing dimensionality reduction techniques can be broadly categorized into three groups: supervised dimensionality reduction methods, unsupervised dimensionality reduction methods, and semi-supervised dimensionality reduction methods. Principle component analysis (PCA) is the most widely used unsupervised dimensionality reduction method and it attempts to seek a subspace by maximizing the variance of the projected data (Martínez & Kak, 2001). In contrast to PCA, linear discriminant analysis (LDA) utilizes the label information and it seeks the transformation matrix by simultaneously maximizing the rank of between-class scatter matrix and minimizing the rank of within-class scatter matrix in order to pull samples with the same label close and separate samples with different labels far from each other (Martínez & Kak, 2001). Though simple and intuitive, PCA and LDA are widely used in data reprocessing and perform well in a wealth of applications such as face recognition, seismic series analysis, visualization, and clustering (Belhumeur, Hespanha & Kriegman, 1997). However, both PCA and LDA only utilize the global structure of the data and assume there does not exist the local properties of the data, which limits their performance in handling complex cases when the above condition is not satisfied (Belhumeur et al., 1997; Martínez & Kak, 2001).

In contrast, another line of research is to explore the local properties of the data. From the perspective of manifold learning, dimensionality reduction essentially aims to find the low-dimensional manifold that is embedded into a high-dimensional space and this embedding keeps the data geometric characteristics as much as possible (Garcia-Vega & Castellanos-Dominguez, 2019; Tenenbaum et al., 2000). Accordingly, researchers have investigated the manifold learning and its application in dimensionality reduction. Isometric mapping (ISOMAP) (Tenenbaum et al., 2000), locally linear embedding (LLE) (Roweis & Saul, 2000), and Laplacian eigenmaps (LE) (He & Niyogi, 2004) are three representative local methods that find a lower-dimensional embedding of the data lying on or around a high-dimensional non-linear manifold. They have achieved satisfactory performance on multiple application domains (Krstanović et al., 2016; van der Maaten et al., 2009), however, they do not provide explicit mapping between the original data and the reduced representation. That is, researchers are generally required to recompute the projection vectors in coping with out-of-sample extension, which greatly limits their flexibility in use and leads to high time costs in processing streaming data. To allow for the efficient embedding of new datapoints, researchers have investigated the linearized version of several non-linear dimensionality reduction methods. For example, He, Cai, Yan and Zhang (2005) proposed the neighborhood preserving embedding (NPE) to linearly approximate LLE. Locality preserving projections (LPP) is a linear approximation to LE (He & Niyogi, 2004). Specifically, LPP is a commonly used and well-performing approach that attempts to obtain a linear transformation matrix by preserving the local neighborhood relationships of the data. LPP has a remarkable advantage in dimensionality reduction and returns an explicit mapping for serving the out-of-sample extension. Compared with most of existing manifold learning methods, LPP not only preserves the local properties of the data, but also returns an explicit transformation matrix. Particularly, the two components of LPP include the construction of neighbor graph and the measurement of similarity between neighbors, both of which largely determine its performance. In addition, several variants of LPP have been proposed and experimentally validated, such as the discriminant locality preserving projections (DLPP) that makes use of label information (Yu, Teng & Liu, 2006) and the null space discriminant locality preserving projections (NDLPP) that is targeted at the small sample size problem of DLPP (Yang, Gong, Gu, Li & Liang, 2008; Yu et al., 2006).

Although LPP and its variants have been successfully applied for real-world applications, it takes the risk of choosing false nearest neighbors and incorrectly calculating the similarity between neighbors and the derived local manifold tends to be error-prone, which inevitably leads to degraded performance for subsequent data analyses. This is mainly because LPP measures the similarity between neighbors in the original feature space where there exist noisy and irrelevant features (Wang et al., 2016; Zhao et al., 2018). Obviously, the obtained neighbor relationships in the optimal subspace are more reliable than the ones in the original feature space and can better reflect the truth. Therefore, how to maximally mitigate the effect of noisy factors and accurately identify the true neighbor relationships for each sample remains crucial. However, we have no prior knowledge of the optimal subspace, which poses a challenge to the determination of the true similarity between neighbors and the robustness improvement of manifold learning-based methods. Accordingly, in this study, we propose a novel approach, termed locality adaptive preserving projections (LAPP) to adaptively determine the neighborhood relationships in the optimal subspace rather than in the original feature space. Specifically, due to the absence of prior knowledge of local properties of the underlying manifold, LAPP adopts a coarse-to-fine strategy to handle the chicken and egg situation. Moreover, an iterative algorithm with fast convergence is utilized to solve the constrained optimization problem for explicit out-of-sample extension. This enables us to better reveal the underlying manifold and obtain corresponding robust embeddings. Particularly, the main contributions of this study are as follows. Frist, we analyze the manifold learning-based dimensionality reduction techniques, especially the commonly used LPP and point out that seeking the local structure in original feature space is error-prone in terms of neighbor-finding and similarity measurement. This potentially motivates researchers to pay special attention to such a problem for other dimensionality reduction methods. Second, we propose a locality adaptive preserving projections approach to optimizing the measurement of neighbor relationships. The proposed method iteratively updates the projected low-dimensional subspace and optimizes the identification of the local structure of the data. Besides, its key idea can be potentially extended to other similar methods. Third, we implement and evaluate the proposed approach on numerous synthetic and real-world datasets. Extensive experimental results show that constructing the neighbor graph in the original feature space suffers from lower performance, which demonstrates the effectiveness of the proposed method.

The reminder of this study is organized as follows. Section 2 briefly reviews related work on dimensionality reduction techniques by introducing four commonly used methods. We detail the proposed locality adaptive preserving projections method and its motivation in Section 3. Section 4 gives the experimental setup and results on both synthetic and real-world datasets and presents corresponding analyses. The last section concludes this study with a brief summary and discusses insightful future research directions.

2. Related work

Over the past few decades, a large number of dimensionality reduction methods have been proposed and used in diverse areas (e.g., decision support systems, face recognition, and data visualization), and we can categorize them from different perspectives. According to whether the mapping function between the high-dimensional space and the reduced feature space is linear, we can group dimensionality reduction techniques into linear methods (e.g., PCA and LDA) and non-linear methods (e.g., LLE, ISOMAP, and LE) (Martínez & Kak, 2001; Roweis & Saul, 2000; Tenenbaum et al., 2000). Compared with traditional linear methods, non-linear methods generally have an advantage in coping with the data that lie on or around a complex non-linear manifold (Weinberger, Sha & Saul, 2004). According to the availability of the supervised information of the data, existing dimensionality reduction techniques can be mainly divided into supervised, unsupervised, and semi-supervised methods (Passalis & Tefas, 2017). For example, LDA belongs to the supervised methods and PCA is a representative of unsupervised methods. Semi-supervised methods deal with the case that only some of the data have supervised information. From the perspective of optimization, we categorize them into convex (e.g., PCA, LLE, and LE) and non-convex dimensionality reduction methods (e.g., locally linear coordination, autoencoder) according to whether the corresponding objective function is convex (Hinton & Salakhutdinov, 2006; The & Roweis, 2002). Specifically, convex techniques ensure the global optimum, while the non-convex techniques get easily trapped into local optima. Besides, according to which information one dimensionality reduction technique aims to preserve, we divide existing dimensionality reduction techniques into global methods and local methods (van der Maaten et al., 2009). For example, PCA, LDA, and autoencoder belong to global methods that try to keep the global properties of the data, whereas LLE, LE, and LPP are local methods that preserve the underlying manifold structure (He & Niyogi, 2004; Tenenbaum et al., 2000). For dimensionality reduction methods, the key is how to measure the manifold and preserve the property in transforming the high-dimensional data into the reduced representation Cai, He, Han and Zhang (2006). For example, Passalis and Tefas (2017) analyzed the advantage of similarity metric over distance metric in preserving the manifold structure and used the target similarity matrix for projection learning. Garcia-Vega and Castellanos-Dominguez (2019) used the Mercer kernel to compute similarity and proposed a kernel-based cost function to minimize the discrepancy between the high-dimensional and reduced representations. A major disadvantage of such methods is that they compute similarities in the original feature space that have noisy and irrelevant features. Accordingly, Pang, Zhou and Nie (2019) modeled the similarity in the low-dimensional space and optimized a rational objective function to learn the projection matrix and class-wise neighborhood similarity. Yang, Wang and Zuo (2012) proposed to construct a neighbors-based local probability model in the subspace and then presented the fast neighborhood component analysis for metric learning. The enhanced performance inspires us to maximally mitigate the effect of noisy factors.

Besides the further endeavor on developing new algorithms, there are significant studies that explore the underlying connections among existing dimensionality reduction methods. For example, Yan et al. (2007) utilized the graph embedding to unify PCA, LDA, ISOMAP, LLE, LE, and LPP into a general framework by defining corresponding similarity matrix and constraint matrix, where the former encodes the statistical or geometric properties of the data and the latter represents scale normalization or a penalty graph. Furthermore, they proposed a novel algorithm called marginal fisher analysis within the framework. To better cope with partially labeled problem, Song, Nie, Zhang and Xiang (2008) investigated the semi-supervised dimensionality reduction techniques and proposed a framework that could unify PCA, LDA, LPP, and maximum margin criterion (MMC). To mitigate the problems and limitations of using an unbounded distance metric to express the objective function, Passalis and Tefas (2017) proposed a similarity-based dimensionality reduction framework. Within the framework, they discussed how to obtain the target similarity matrix and presented a method on how to clone an existing dimensionality reduction method. This enables the explicit out-of-sample extension. Herein, to have a general idea of different categories of dimensionality reduction techniques and better understand their relationships with the proposed approach, we introduce four representative algorithms that belong to linear/non-linear and global/local methods. Notably, to gain deeper insights into other techniques, there are excellent literature reviews on dimensionality reduction for a centralized outlook into a specific topic, such as metric learning (Weinberger, Blitzer & Saul, 2006), manifold learning, and non-linear dimensionality reduction techniques (van der Maaten et al., 2009). Next, we detail four dimensionality reduction methods, including PCA, LLE, LE, and LPP. By default, we use bold lower-case and upper-case front to denote vectors and matrices, respectively, and regular fonts for scalars.

2.1. Principle component analysis

As one of the commonly used preprocessing techniques, PCA is an unsupervised linear dimensionality reduction approach that transforms the high-dimensional data into a new subspace of lower dimensionality (Martínez & Kak, 2001). Specifically, given a training dataset $\boldsymbol{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x}_{\boldsymbol{N}}] \epsilon\mathbb{R}^{m\times N}$with N data points that are in a m-dimensional feature space, PCA aims to find a linear mapping $\mathbf{A}$ to derive a low-dimensional representation of the data while maximizing the amount of variance. In mathematical terms, the objective function of PCA is formalized as:

$$\begin{cases}\arg\max\limits_{\mathbf{A}}(\mathbf{A}^T\mathrm{cov}(\mathbf{X})\mathbf{A}) \\ s.t. \mathbf{A}^T\mathbf{A} = \mathbf{I}\end{cases} \tag{1}$$

where $\mathbf{A}^T$ is the transpose of $\mathbf{A}$, $\mathrm{cov}(\mathbf{X})$ is the covariance matrix of the data $\mathbf{X}$, and the constraint condition is to avoid trivial solutions.

After rigorous math transformations, the optimization problem of Eq. (1) reduces to an eigenproblem,

$$\mathrm{cov}(\mathbf{X})\mathbf{A} = \lambda\mathbf{A} \tag{2}$$

Obviously, the solutions **A** are the eigenvectors associated with the d principle eigenvalues of cov(**X**). Afterwards, we compute the low-dimensional representation **Y** of **X** by performing linear mapping with **A** $\epsilon\mathbb{R}^{m\times d}$, i.e., $\boldsymbol{Y} = \mathbf{A}^T\mathbf{X}$. PCA has advantages of simplicity, easy explanation, optimal reconstruction, and explicit out-of-sample extension, but it has limited power in handling data lying on or around a non-linear manifold such as Swiss roll data. Besides, PCA pays much attention to large pairwise distances rather than the small pairwise distances.

2.2. Locally linear embedding

The key idea of LLE is that the geometry of local neighbors should be maintained in the reduced feature space, that is, each data point can be linearly represented by its neighbors in the high dimensional space, and then exploit the learned weights to reconstruct each projected data point in the low-dimensional space (Roweis & Saul, 2000). Specifically, LLE first represents a datapoint $\mathbf{x}_i$ as a linear combination of its k nearest neighbors $\mathbf{x}_{ij}$ with corresponding weight vector $\mathbf{w}_i$ in the original feature space using Eq. (3),

$$\arg\min_{w}\sum_i\left\|\mathbf{x}_i - \sum_{j=1}^{k} w_{ij}\mathbf{x}_{ij}\right\|^2 \tag{3}$$

where w_{ij} represents the weight between $\boldsymbol{x}_i$ and its $\boldsymbol{x}_j$ and k denotes the number of neighbors of interest.

According to the local linearity assumption, $\mathbf{w}_i$ is invariant to rescaling, translation, and rotation, and then LLE fixes $\mathbf{w}_i$ in the embedded low-dimensional data representation. Thus, LLE uses $\mathbf{w}_i$ to reconstruct the reduced representation $\mathbf{y}_i$ of $\mathbf{x}_i$ with its neighbors $\mathbf{y}_{ij}$ and solves the following objective function to obtain the d-dimensional representation **Y** $\epsilon\mathbb{R}^{d\times N}$,

$$\begin{cases}\min\sum\limits_i\left\|\mathbf{y}_i - \sum\limits_{j=1}^{k} w_{ij}\mathbf{y}_{ij}\right\|^2\\ s.t.||\mathbf{y}^{(n)}||^2 = 1, \forall n\end{cases} \tag{4}$$

where $\boldsymbol{y}_{ij}$ is the jth neighbor of $\boldsymbol{y}_i$ and $\boldsymbol{y}^{(n)}$ is the nth column of $\boldsymbol{Y}$. The constraint condition is to avoid the trivial solution $\boldsymbol{Y} = \boldsymbol{0}$. Obviously, there is no explicit mapping between $\boldsymbol{X}$ and $\boldsymbol{Y}$ that we can use for out-of-sample extension.

2.3. Laplacian eigenmaps

Similar to LLE, LE also preserves the local properties of the data (He & Niyogi, 2004). The objective of LLE is to minimize the distances between a data point and its k nearest neighbors in the reduced feature space,

$$\min\varphi(\mathbf{Y}) = \min\sum_{i,j}\left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2 s_{ij} \tag{5}$$

where $\mathbf{y}_i$ is the low-dimensional representation of $\mathbf{x}_i$ and s_{ij} denotes the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. We can define s_{ij} in a supervised or unsupervised way.

For the supervised setting, given a dataset with c classes, define

$$s_{ij} = \begin{cases} d(\mathrm{x}_i,\mathrm{x}_j),\ \mathrm{if}(\mathrm{x}_i\mathrm{and}\mathrm{x}_j) \in \mathrm{thesameclass}\\ 0,\ \mathrm{otherwise}\end{cases} \tag{6}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ represents the weight of the edge between $\mathbf{x}_i$ and $\mathbf{x}_j$. We can calculate the weight with heat kernel or cosine function,

$$d_{ij} = e^{-\frac{-||\mathbf{x}_i-\mathbf{x}_j||^2}{2\delta^2}},\ \mathrm{or}d_{ij} = \cos(\mathbf{x}_i, \mathbf{x}_j) \tag{7}$$

For the unsupervised case, define

$$s_{ij} = \begin{cases} d(\mathrm{x}_i,\mathrm{x}_j),\ \mathrm{if}\mathrm{x}_i \in N_k(\mathrm{x}_j)\mathrm{or}\mathrm{x}_i \in N_k(\mathrm{x}_j)\\ 0,\ \mathrm{otherwise}\end{cases} \tag{8}$$

where $N_k(\mathbf{x}_j)$ denotes the set of k nearest neighbors of $\mathbf{x}_j$.

Afterwards, we can obtain a similarity matrix **S** $\epsilon\mathbb{R}^{N\times N}$whose ($i$, j) entry is s_{ij}. After rigorous math transformations, the optimization problem Eq. (5) reduces to

$$\begin{cases}\min\sum\limits_{i,j}\left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2 s_{ij} = \min\limits_{\mathbf{Y}} 2\mathbf{Y}^T\mathbf{L}\mathbf{Y}\\ s.t.\mathbf{Y}^T\mathbf{D}\mathbf{Y} = \mathbf{I},\ \mathbf{D} = \sum_j\mathbf{S}_{ij}\end{cases} \tag{9}$$

where **L** = **D - S** is the Laplacian matrix. We obtain **Y** by solving the following generalized eigenvalue problem (Eq. (10)),

$$\mathbf{L}\upsilon = \lambda\mathbf{D}\upsilon \tag{10}$$

The solutions are the eigenvectors that are associated with the d smallest non-zero eigenvalues. One drawback of LE is that there is no explicit mapping to directly transform new out-of-samples into the low-dimensional data representation.

2.4. Locality preserving projections

LPP is essentially a linearization procedure of LE and it attempts to obtain a transformation matrix $\boldsymbol{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_d]\ \epsilon\mathbb{R}^{m\times d}$ that best preserves the local properties of the reduced data (He & Niyogi, 2004). Each column $\mathbf{a}_i$ (i = 1, ..., d) of **A** is a basis vector of the low-dimensional space. The reduced data representation $\mathbf{y}_i$ $\epsilon\mathbb{R}^{d\times 1}$of $\mathbf{x}_i$ is $\mathbf{y}_i = \mathbf{A}^T\ \mathbf{x}_i$, and the objective function of LPP is to minimize the constrained optimization problem (Eq. (11)),

$$\begin{cases}\arg\min\limits_{\mathbf{a}}\sum\limits_{i,j}\left\|\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right\|^2 s_{ij}\\ s.t.\mathbf{a}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{a} = \mathbf{I}\end{cases} \tag{11}$$

which means if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other, $\mathbf{y}_i$ and $\mathbf{y}_j$ should be close. After rigorous math transformations, Eq. (11) reduces to a generalized eigenvalue problem,

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A} = \boldsymbol{\Lambda}\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A} \tag{12}$$

where **L** is the Laplacian matrix, **D** is the diagonal matrix, and **Λ** is a diagonal matrix with eigenvalues on the diagonal.

Compared with LE, LPP provides **A** to explicitly project new data points to its reduced representation of low-dimensionality. However, LPP measures the pairwise similarity between neighbors in the original feature space that contains noisy and irrelevant features, which misleads the selection of neighbors and the measurement of neighborhood similarity and further leads to degraded performance. In the next section, we present how to optimize the derivation of subspace and the selection of neighbors without having prior knowledge of the local properties.

3. Locality adaptive preserving projections

As we discussed above, the selection of nearest neighbors and the measurement of neighborhood relationships largely determines the performance of locality preserving projections. Particularly, how to largely reduce the effect of unimportant factors and accurately seek the neighbors in the optimal subspace remains critical. Ideally, if we have prior knowledge of the noisy and irrelevant features of the data, we get the true pairwise distances between neighbors and derive the optimal feature space. On the contrary, if we know the optimal subspace, we basically obtain the true neighborhood relationships of each data point and further mitigate the effect of noisy features. Unfortunately, we have no prior knowledge. Hence, how to adaptively determine the optimal subspace is the main focus of this study.

Naturally, the key idea of Expectation Maximization (EM) delivers a meaningful solution to the above dilemma. Accordingly, we are required to provide a good initial value. Since the locality preserving projections has the ability to extract meaningful representation of reduced dimensionality from the high-dimensional data and to preserve the manifold structure, we tentatively use the nearest neighbors that are measured in the original feature space as initial values and adopt an iterative process to optimize the transformation matrix **A**. Herein, we propose the locality adaptive preserving projections (LAPP) method towards better preserving the local properties of the data. Specifically, we first calculate the weight between any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ of **X** using (7). We then calculate the similarity matrix **S**, diagonal matrix **D** as well as Laplacian matrix **L**. Afterward, we solve a generalized eigenvalue problem and get a transformation matrix **A**. Subsequently, we conduct a step to refine the measurement of neighborhood relationships. With the returned **A**, we project **X** into a new subspace $\mathbf{X_0} = \mathbf{A}^T\mathbf{X}$, which essentially functions as a candidate of the optimal subspace. Then, we calculate the weight between any two points of $\mathbf{X_0}$ and select the nearest neighbors of each reduced data point. Afterwards, we get a new transformation matrix **A**. Continue with the above procedure until a specific condition is met, such as the difference of **A** between two consecutive iterations less than a threshold, the maximum number of iterations, and the time budget. Algorithm 1 presents the pseudo-code of the proposed LAPP, where two iteration stop conditions are used. Line 5 controls the maximum number of iterations and line 11 controls the precision.

With **A**, we can transform the original training set and new out-of-sample datapoints into their reduced representation for subsequent analysis. For example, in the application of face recognition, we project the original face images into a new reduced subspace, where we train a classifier on the mapped training set and further infer the label of the mapped test samples. Although, as expected, LAPP suffers from a high time complexity, it converges after less than twenty iterations on average, which is easily affordable for many real-world applications. Besides, LAPP is easy to implement for practical applications.

4. Experimental results and analysis

To evaluate the effectiveness of the proposed method, we conduct extensive experiments on two synthetic Swiss roll datasets, three face recognition benchmark datasets, including the Yale face database (YALE), Olivetti research laboratory database (ORL), and extended Yale Face Database B (E_YALE), as well as one handwritten digit recognition dataset MNIST. We compare LAPP with other four well-performing dimensionality reduction methods, including two global methods (PCA and LDA) and two local methods (NPE and LPP). In constructing the affinity matrix, if the class label is available, we connect two points of the same class and measure the similarity. For the Swiss roll datasets that lie in a three-dimensional space, we apply the above dimensionality reduction methods to transform the data into a two-dimensional space and plot corresponding data distributions. For the task on real-world datasets, we use dimensionality reduction methods to transform the data into reduced representations and then utilize the nearest neighbor classifier (NN) to associate the test sample with its labels. Accuracy is used as the evaluation metric. Besides, for the baseline method, classification is performed in the original feature space without conducting dimensionality reduction.

4.1. Results on synthesis datasets

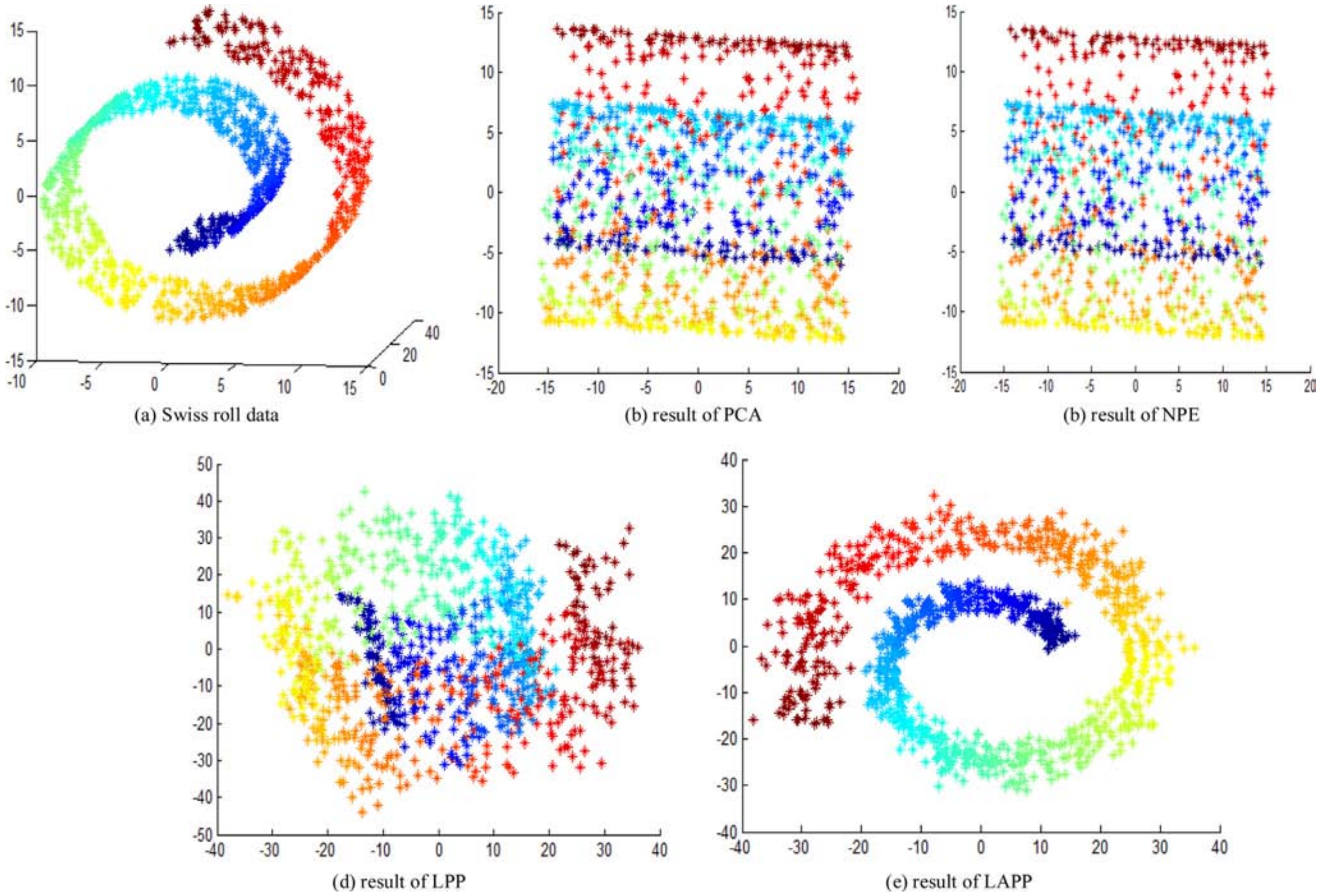
We first evaluate the performance of LAPP on two Swiss roll datasets. In this study, the first dataset consists of 1000 data points and the second contains 2000 data points. We generate the three coordinates (*x, y, z*) of Swiss roll according to the following functions,

$$\begin{cases} x = t{*}\cos(t), y = h, z = t * \sin(t) \\ s.t.\, t \in [3\pi/2, 9\pi/2], h \in [0, 41] \end{cases} \tag{13}$$

After obtaining the transformation matrix using the dimensionality reduction methods, we project the artificially generated three-dimensional data into a new two-dimensional space. Fig. 1 presents the results of PCA, LDA, NPE, LPP, and LAPP on 1000 data points, and Fig. 2 is associated with the results of 2000 data points. From Figs. 1 and 2, we observe that LAPP better preserves the local structure of the data. Compared to the three-dimensional Swiss roll, the results of LAPP also show a two-dimensional Swiss roll. That is, LAPP preserves both the local structure and the global distribution of the data. Besides, we observe that LPP performs better than NPE, and that NPE, LPP, and LAPP perform better than PCA. A possible explanation is that PCA aims to preserve the global property of the data and fails to discover the local structure that exists

Algorithm 1 Pseudo-code of locality adaptive preserving projections.

```
Input: $X = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x}_N] \in \mathbb{R}^{m\times N}$, the final dimension $d$, threshold $\delta$, maximum number of iterations $T$
Output: transformation matrix $\mathbf{A} \in \mathbb{R}^{m\times d}$
1. calculate the similarity $s(\mathbf{x}_i, \mathbf{x}_j)$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ of $\mathbf{X}$ using (6)
2. calculate S, D, and L according to $s(\mathbf{x}_i, \mathbf{x}_j)$
3. solve the generalized eigenvalue problem (12) and obtain A
4. $iteration = 0$ //counter
5. while $iteration < T$
6.         obtain the transformed data $\mathbf{X_0} = \mathbf{A}^T\mathbf{X}$
7.         $\mathbf{A_0} = A$ //save A
8.         measure the similarity $s(\mathbf{x}_i, \mathbf{x}_j)$ within $\mathbf{X_0}$
9.         calculate S, D, and L
10.          solve a generalized eigenvalue problem to get A:
               $\mathbf{X_0 L X_0^T A} = \mathbf{\Lambda X_0 D {X_0}^T A}$
11.         if $diff(\mathbf{A}, \mathbf{A_0}) < \delta$ //differences of two consecutive iterations
12.              $A = \mathbf{A_0}$, break;
13.         end if
14. $iteration = iteration + 1$
15. end while
16. return A
```



(a) Swiss roll data (b) result of PCA (b) result of NPE

(d) result of LPP (e) result of LAPP

Fig. 1. Results of PCA, NPE, LPP, and LAPP on a Swiss roll dataset that consists of 1000 points.

in complex non-liner data and NPE has a difficulty in coping with the manifolds that contain holes.

4.2. Results on YALE database

The YALE database consists of 165 images of 15 subjects (Belhumeur et al., 1997). For each individual, there are 11 images and one per different illuminations or facial expressions: happy, sleepy, surprised, sad, wink, normal, w/glasses, w/no glasses, center-light, left-light, and right-light. These images are manually cropped to 32×32 pixels and converted to grayscale images. Accordingly, Fig. 3 presents exemplar face images.

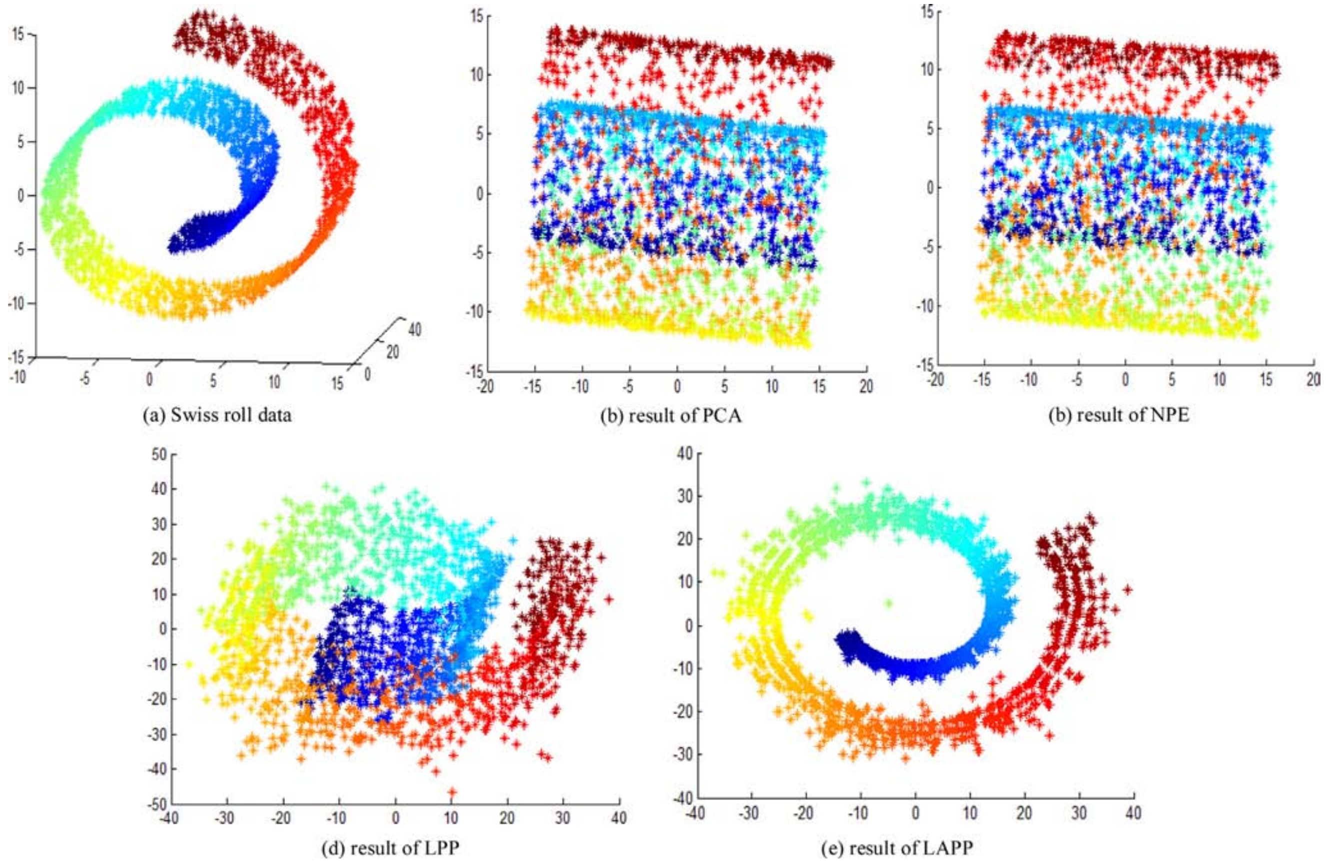
In our experiment, we randomly select$l$(= 2, 4, 6, 8) images per individual to form the training set and take the remaining images as the test set. For each l, we repeat the experiments over fifty random splits of the dataset and report the average results. To better explore the intrinsic dimensionality, we conduct experiments within a wide range of projected dimensions and set the maximal projected dimension to be 100. Note that, because LDA has at most c-1 nonzero eigenvalues (c is the number of individuals), an upper bound on the reduced dimensionality is $c-1$. The training samples are used to learn the projective functions and nearest neighbor (NN) is used as the classifier. For the baseline method, the recognition is performed in the original feature space without any dimensionality reduction. Table 1 presents the lowest error rates and corresponding dimension that are returned by PCA, LDA, NPE, LPP, and LAPP, respectively, for each training case with l (= 2, 4, 6, 8). The second row "Baseline" denotes the results without using dimensionality reduction. The best values in terms of error rates are highlighted in bold.

Table 1
Error rates (%) with l (= 2, 4, 6, 8) training images per individual on the YALE dataset. The dimension that results in the best error rate of each method is shown in the parentheses.

	2 Train	4 Train	6 Train	8 Train
Baseline	56.56	47.35	41.31	36.36
PCA	56.56 (29)	47.35 (60)	40.83 (32)	35.91 (30)
LDA	70.43 (12)	65.07 (14)	61.07 (14)	54.98 (14)
NPE	77.29 (14)	71.18 (23)	67.33 (26)	65.47 (14)
LPP	43.60 (14)	28.80 (14)	**22.83 (20)**	20.84 (21)
LAPP	**43.29 (14)**	**28.42 (15)**	**22.83 (23)**	**20.58 (23)**

We observe in Table 1 that there exists a general trend that the error rates decrease with the increase of the number of training samples. This indicates that a larger size of training set contributes to the improvement of recognition performance. Furthermore, we see that LAPP outperforms all its competitors and that LPP performs better than PCA, LDA, and NPE. For example, in the case of $l = 2$, PCA achieves an error rate of 56.56%, LDA has an error rate of 70.43%, NPE has an error rate of 77.29%, while the error rates of LPP and LAPP are 43.60% and 43.39%, respectively; for $l = 8$, LAPP achieves an error rate of 20.58%, which is lower than 25.91% of PCA, 54.98% of LDA, 65.47% of NPE, and 20.84% of LPP.

Besides, Fig. 4 plots the recognition error rates versus different reduced dimensions for PCA, LDA, NPE, LPP, and LAPP. The projected dimension varies from 5 to 100. The X-axis represents the dimension of reduced data representation and the Y-axis denotes the classification error rates of each method. From Fig. 4, we observe that the minimal error rate of LAPP is smaller than that of PCA, LDA, NPE, and LPP. We also observe that the error rates of



(a) Swiss roll data (b) result of PCA (b) result of NPE

(d) result of LPP (e) result of LAPP

Fig. 2. Results of PCA, NPE, LPP, and LAPP on a Swiss roll dataset that consists of 2000 points.

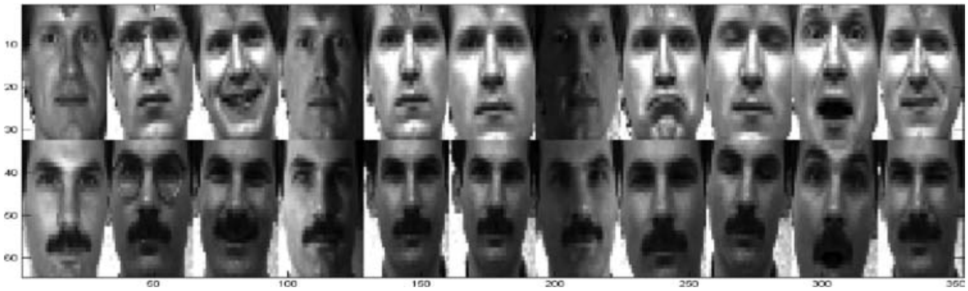


Fig. 3. Exemplar face images of YALE.

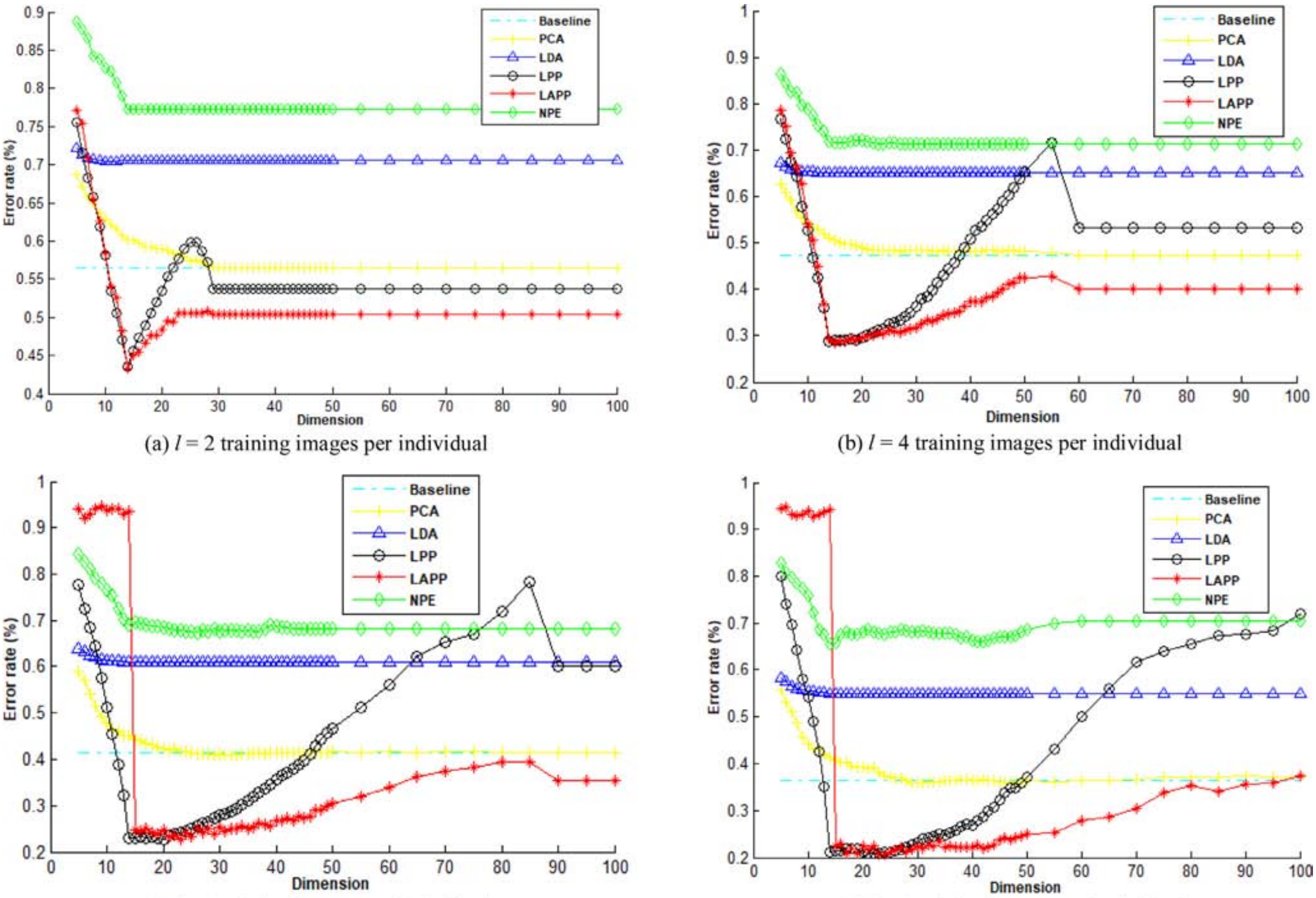
each method first decrease and then increase along with the increase of dimension. The phenomenon is probably because we first approach the intrinsic dimensionality and then stray from it. This indicates that the inflection point potentially corresponds to the intrinsic dimensionality of the data. Besides, from Fig. 4, we observe that in the case of a small number of training samples, LAPP obtains slightly better recognition rates than its competitors, while in the case of a large number of samples, LAPP performs quite differently from other methods and has a steep inflection point that helps better study the intrinsic dimensionality of the data.

4.3. Results on ORL database

The Olivetti Research Laboratory (ORL) database contains 400 grayscale face images of 40 individuals (10 samples per individual) (Samaria & Harter, 1994). The images were taken at different conditions: different times, lighting, facial details (with/out glass), and facial expressions (open/closed eyes, smiling/not smiling), and all images were taken against a dark homogeneous background. These images are manually cropped to 32 × 32 pixels and form a 1024-dimensional feature vector. To have a general idea of ORL, Fig. 5 presents exemplar face images.

In this experiment, we randomly select$l(= 2, 4, 6, 8)$ images per individual to form the training set and take the remaining images as the test set. For each l, we repeat the experiments over 50 random splits of the dataset and report the average accuracy. The training samples are used to learn the projective functions and NN is used as the classifier. We give the best error rates and corresponding dimension that are returned by PCA, LDA, NPE, LPP, and LAPP for each training case with l $(= 2, 4, 6, 8)$ in Table 2. The best values achieved by these methods in terms of error rates are highlighted in bold. The second row "Baseline" presents the results without using dimensionality reduction.

Similar to the case of YALE, we observe in Table 2 that there exists a general trend that the error rates decrease with the increase of the number of training samples. Furthermore, we see



(a) $l = 2$ training images per individual

(b) $l = 4$ training images per individual

(c) $l = 6$ training images per individual

(d) $l = 8$ training images per individual

Fig. 4. Error rates versus dimensions of PCA, LDA, NPE, LPP, LAPP, and Baseline method on YALE.

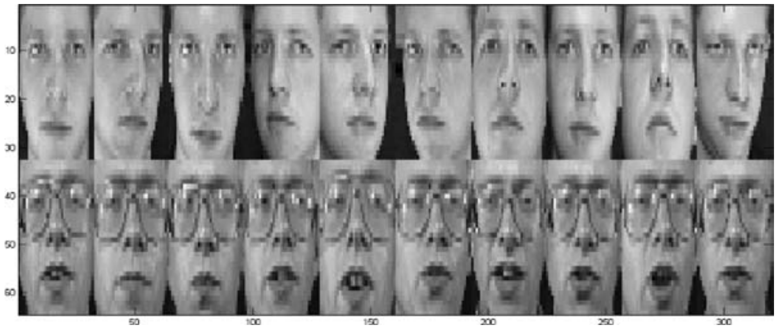


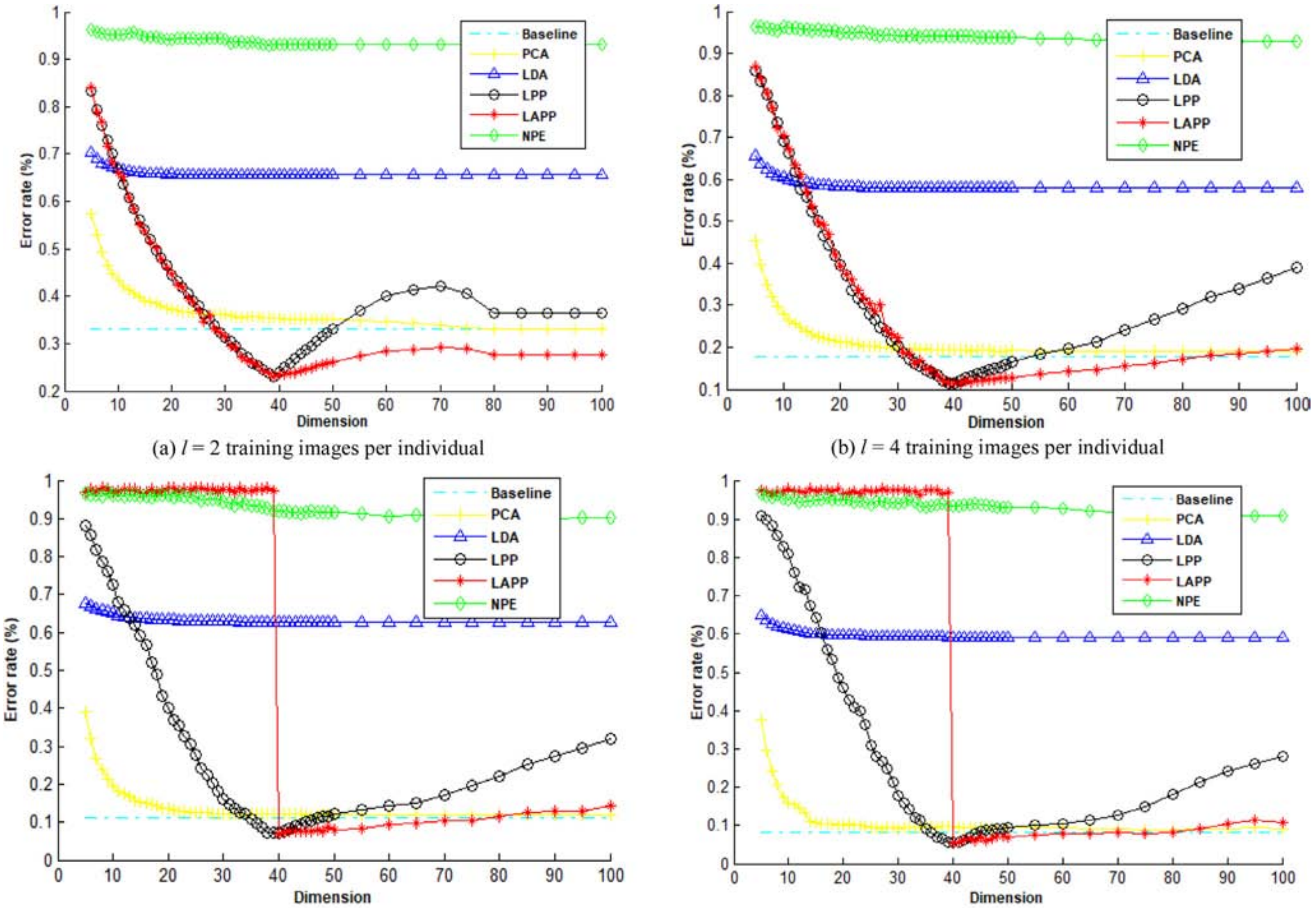
Fig. 5. Exemplar face images of ORL.

Table 2
Error rates (%) with l (= 2, 4, 6, 8) training images per individual on the ORL dataset. The dimension that results in the best error rate for each method is shown in the parentheses.

	2 Train	4 Train	6 Train	8 Train
Baseline	33.08	17.86	11.06	8.25
PCA	33.08 (80)	18.98 (100)	11.69 (70)	8.75 (75)
LDA	65.59 (38)	57.90 (37)	62.81 (36)	59.25 (39)
NPE	92.96 (38)	93.00 (75)	90.03 (90)	90.69 (85)
LPP	23.08 (39)	11.38 (39)	7.16 (39)	5.56 (39)
LAPP	**22.91 (39)**	**11.38 (39)**	**7.00 (40)**	**5.19 (40)**

that LAPP outperforms its competitors. For example, for the case of $l = 2$, LAPP achieves an error rate of 22.91%, which is lower than 33.08% of Baseline, 33.08% of PCA, 65.59% of LDA, 92.96% of NPE, and 23.08% of LPP; for the case of $l = 8$, LAPP achieves an error rate of 5.19%, compared to the 8.25% of Baseline, 8.75% of PCA, 59.25% of LDA, 90.69% of NPE, and 5.56% of LPP. We also observe that the inappropriate choice of dimensionality reduction methods can degrade the classification performance. For example, for $l = 4$, Baseline returns an error rate of 17.86%, which is smaller than the 18.98% of PCA, 57.90% of LDA, 93.00% of NPE.

Besides, Fig. 6 plots the recognition error rates versus different projected dimensions for PCA, LDA, NPE, LPP, and LAPP. The projected dimension varies from 5 to 100. The X-axis represents the dimension of reduced data representation and the Y-axis denotes the classification error rates of each method. From Fig. 6, we observe that LAPP obtains the minimal error rate and that the error rates of each method first decrease and then increase along with the increase of dimension. This indicates that we first approach the intrinsic dimensionality and then stray from it.



(a) $l = 2$ training images per individual

(b) $l = 4$ training images per individual

(c) $l = 6$ training images per individual

(d) $l = 8$ training images per individual

Fig. 6. Error rates versus dimensions of PCA, LDA, NPE, LPP, LAPP, and Baseline method on ORL.

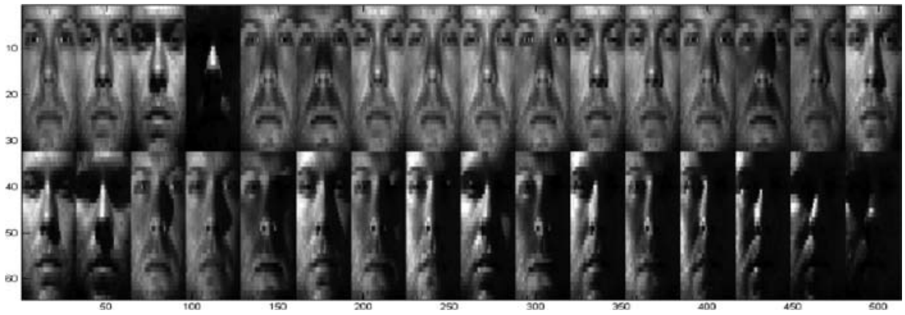


Fig. 7. Exemplar face images of E_YALE.

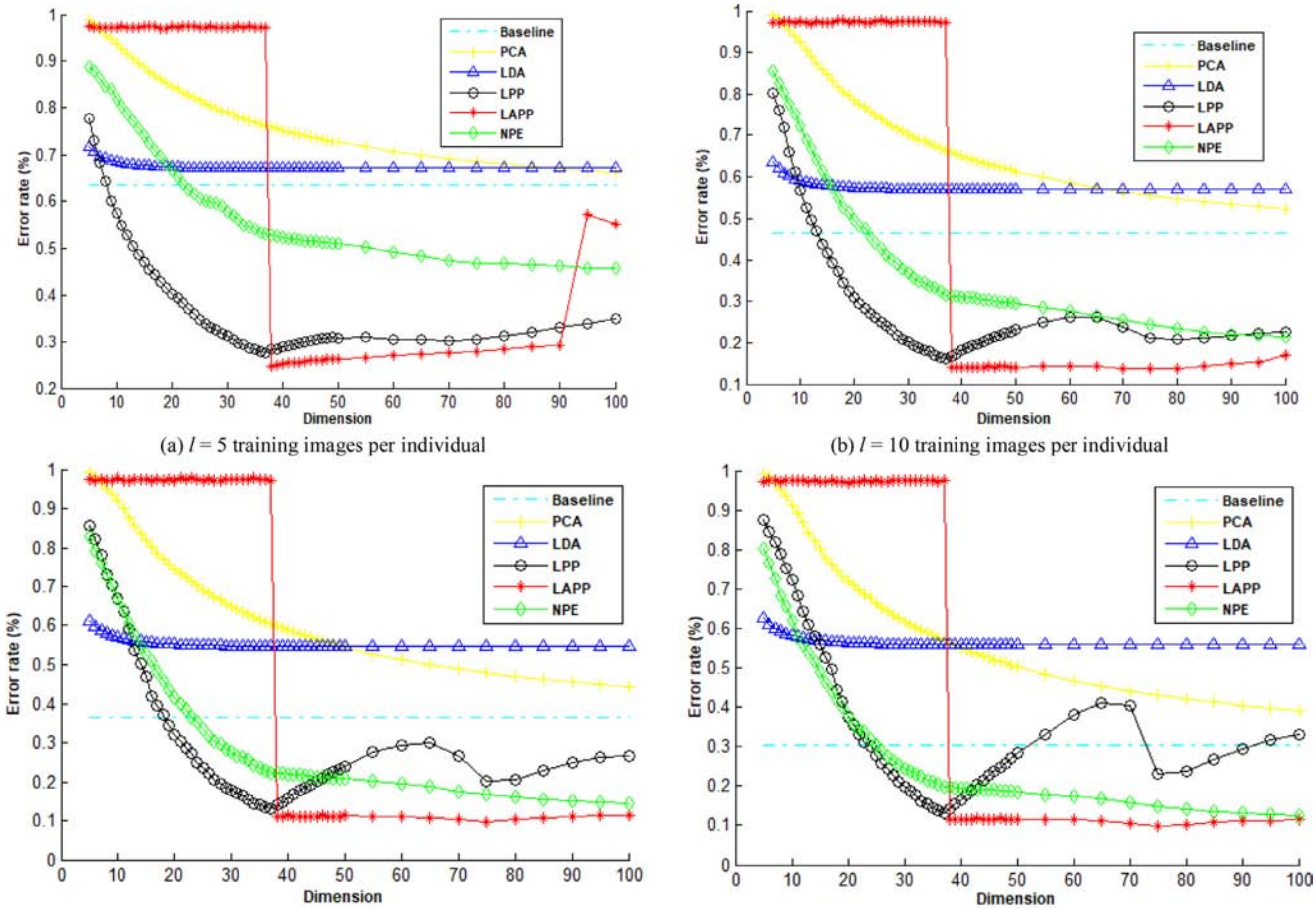
4.4. Results on E_YALE database

The extended Yale Face Database B (E_YALE) consists of 16,128 images of 28 individuals that are collected under 9 poses and 64 illumination conditions (Lee, Ho & Kriegman, 2005). Compared with YALE, this poses a great challenge to face recognition. These images are manually cropped to 32×32 pixels and form a 1024-dimensional feature vector. To have a general idea of E_YALE, Fig. 7 presents exemplar face images.

In this experiment, we randomly select l (= 5, 10, 15, 20) images per individual to form the training set and take the remaining images as the test set. For each l, we repeat the experiments over 50 random splits of the dataset and report the average accuracy. The training samples are used to learn the projective functions and NN is used as the classifier. We give the best error rate and corresponding dimension returned by PCA, LDA, NPE, LPP, and LAPP for each training case with l (= 5, 10, 15, 20) in Table 3. The best values achieved by these methods in terms of error rates are highlighted in bold. Also, the second row "Baseline" presents the results without using dimensionality reduction.

Table 3
Error rates (%) with l (= 5, 10, 15, 20) training images per individual on the E_YALE dataset. The dimension that results in the best error rate for each algorithm is shown in the parentheses.

	5 Train	10 Train	15 Train	20 Train
Baseline	63.60	46.40	36.56	30.40
PCA	66.10 (100)	52.24 (100)	44.07 (100)	39.04 (100)
LDA	67.16 (37)	57.06 (37)	54.78 (37)	55.87 (37)
NPE	45.68 (95)	21.35 (100)	14.33 (100)	12.50 (100)
LPP	27.48 (37)	16.09 (37)	12.98 (37)	12.96 (37)
LAPP	**24.61 (38)**	**13.67 (75)**	**9.68 (75)**	**9.65 (75)**



(a) $l = 5$ training images per individual

(b) $l = 10$ training images per individual

(c) $l = 15$ training images per individual

(d) $l = 20$ training images per individual

Fig. 8. Error rates versus dimensions of PCA, LDA, NPE, LPP, LAPP, and Baseline method on E_YALE.

According to Table 3, we also observe that there exists a general trend that the error rates decrease with the increase of the number of training samples. We also observe that LAPP outperforms its competitors. For example, for the case of $l = 5$, LAPP achieves an error rate of 24.61%, compared to the 63.60% of Baseline, 66.10% of PCA, 67.16% of LDA, 45.68% of NPE, and 27.48% of LPP; for the case of $l = 20$, compared to the 30.40% of Baseline, 39.04% of PCA, 55.87% of LDA, 12.50% of NPE, and 12.96% of LPP, LAPP achieves an error rate of 9.65%.

In addition, Fig. 8 plots the recognition error rates versus different projected dimensions for PCA, LDA, NPE, LPP, and LAPP. The projected dimension varies from 5 to 100. The X-axis represents the dimension of reduced data representation and the Y-axis denotes the classification error rates of each method. From Fig. 8, we observe that LAPP obtains the minimal error rate. Besides, we observe LAPP has a steep inflection point that probably corresponds to the intrinsic dimensionality.

4.5. Results on MNIST

We further conduct comparative experiments on the handwritten digit recognition dataset, i.e., MNIST. These images are manually cropped to 28 × 28 pixels and each is represented by a 784-dimensional feature vector. There are ten different digits (0 to 9), and Fig. 9 presents the associated exemplar images. MNIST has separate train and test samples, and we select the first $l = 3000$ and 4000 samples from the train set, respectively, and use the whole test samples for two groups of evaluation. Fig. 10 presents the experimental results. We observe that the performance of manifold-based methods is inferior to PCA that does not use neighborhood graphs and even the baseline method. This is consistent with previous studies that report similar results (van der Maaten et al., 2009), and also indicates the nature of data influences a dimensionality reduction approach (Passalis & Tefas, 2017). The possible explanation lies in the lack of manifold in the data and the global covariance matrix is more feasible for capturing the latent information. Despite this, we observe that the proposed method outperforms LPP, indicating the superiority of our proposed strategy in optimizing the local structure.

4.6. Evaluation of the number of neighbors

Herein, we evaluate the effect of varying the number of neighbors on the performance. Fig. 11 presents corresponding results. The X-axis represents the number of neighbors and its maximal value is less than the number of samples per class (minus one), and the Y-axis corresponds to the error rates. From Fig. 11, we observe that the number of neighbors indeed influences the performance of LAPP and that taking two samples within the same class as neighbors generally obtains better performance. A possible explanation lies in the adaptive measurement of neighborhood similarity.

Overall, according to the above results and comparative analyses, we conclude that the proposed method helps reduce the negative effect of noisy and irrelevant features and contributes to obtaining accurate neighborhood relatedness and preserving the data manifold. Particularly, as for the above face recognition experiments, we find that LAPP obtains slightly lower recognition error

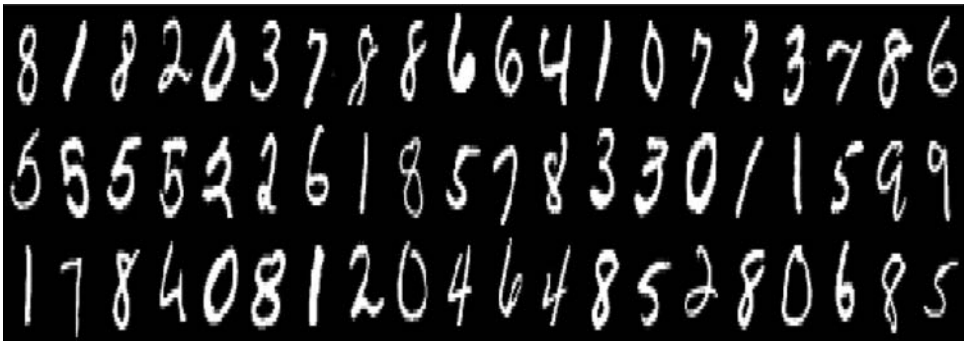
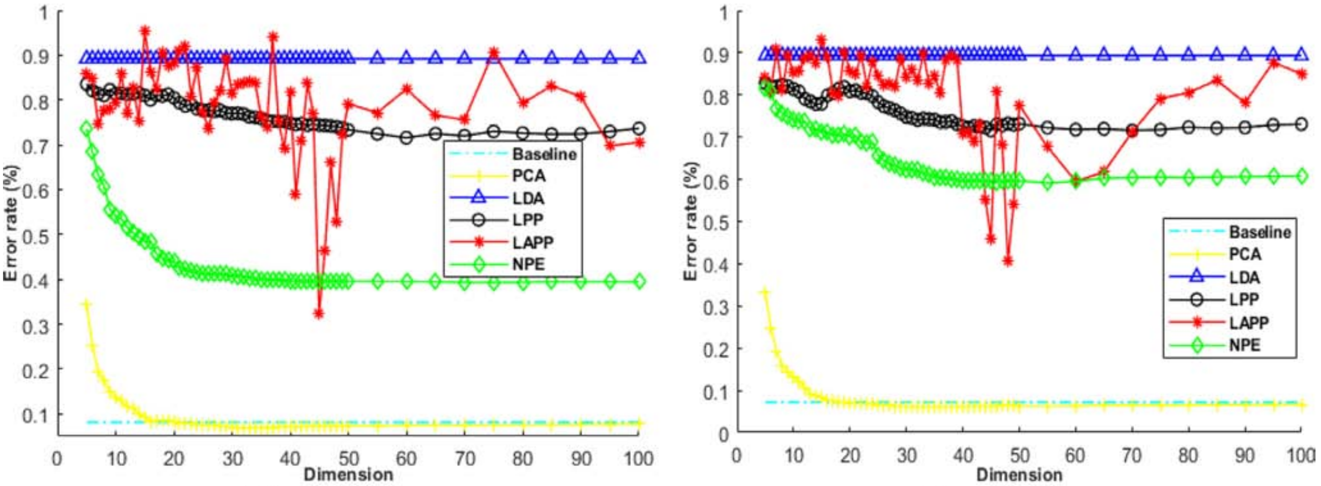
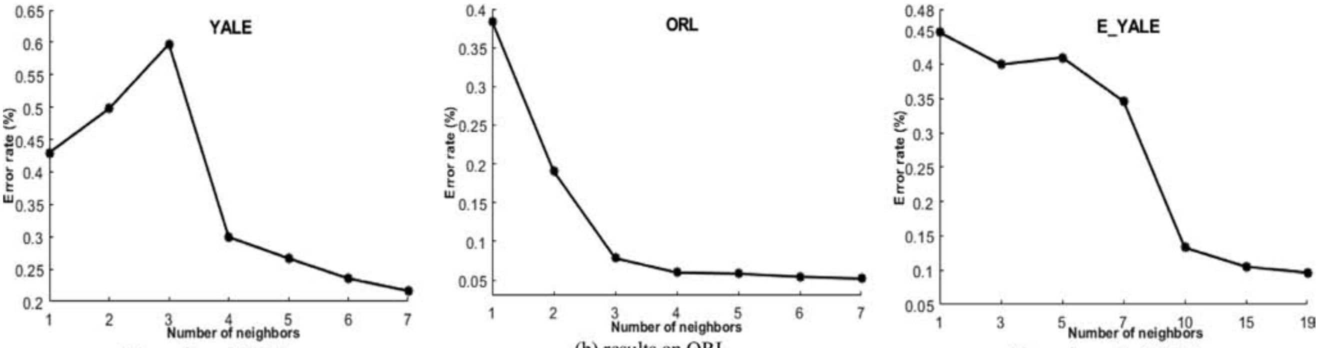


Fig. 9. Exemplar digit images of MNIST.



(a) l = 3000 samples (b) l = 4000 images

Fig. 10. Error rates versus dimensions of PCA, LDA, NPE, LPP, LAPP, and Baseline method on MNIST.



(a) results on YALE (b) results on ORL (c) results on E_YALE

Fig. 11. Error rates versus the number of neighbors on the projection.

rates than its competitors under a small-size training set, while in case of a larger number of training samples, the use of LAPP obtains lower error rates and helps explore the intrinsic dimensionality of the data. This indicates the superiority of LAPP in handling a relatively large-size dataset to a certain extent.

5. Conclusions

Dimensionality reduction techniques have been consistently playing an important role in the procedure of data analysis and the design of an intelligent expert system such as face recognition and disease diagnosis, and they significantly facilitate, among other tasks, the classification, clustering, visualization, and compression in handling the data of high-dimensionality. Due to the complex non-linear relations inherent in data, however, dimensionality reduction methods that preserve the local properties of the data generally obtain better performance, among which, locality preserving projections explores the manifold structure and has been successfully applied to multiple applications. Particularly, for local methods, how to obtain the true neighbors and accurately measure the neighborhood relatedness for each data point remains crucial. Because of the noisy and irrelevant features, it is error-prone to determine neighbors in the original feature space. Herein, we propose the locality adaptive preserving projections (LAPP) method to seek reliable neighbors in the optimal subspace, where LAPP iteratively updates the projected optimal subspace and optimizes the similarity measurement of the data. Moreover, LAPP is easy to implement. Extensive comparative experiments are conducted on both synthetic and real-world datasets. Experimental results show that seeking the local structure in the original space misleads the selection of neighbors and the calculation of similarity, which demonstrates the superiority of the proposed method.

Along with this study, we plan to work in the following research lines for the future work. First, the proposed method may fail to effectively deal with the linearly non-separable data. In view of the fact that kernel-based techniques help alleviate the problem (Schölkopf, Smola & Müller, 1998), we plan to explore its corresponding kernel version and experimentally validate it. Second, although the proposed method facilitates the search and selection of candidate intrinsic dimensions for a specific application, the underlying theoretical explanation and other theoretical ways to determining the intrinsic dimensionality are in need. Third, the experimental results demonstrate the significant role of neighborhood relationships in designing a dimensionality reduction algorithm. This inspires us to explore and improve relevant methods for enhanced performance. Fourth, previous researches show that working on 1-D vector fails to exploit the spatial structure information of the multiple dimensional objects such as medical images and remote sensing images (Zhang, Nie, Zhang & Li, 2018). The use of matrix-based 2-D and even tensor-based dimensionality reduction methods is a priority. Similarly, preserving the true neighborhood relatedness for such types of data is crucial, which deserves further study to explore how to enhance their robustness. Finally, the choice of dimensionality reduction methods for real-world applications mainly depends on the characteristics of the used data and the type of the learning task, while the large number of existing methods may confuse users. Therefore, an initial effort towards developing a practical guideline on how to objectively evaluate existing methods and appropriately choose the most suitable one remains another topic for future research.

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.

Credit authorship contribution statement

Aiguo Wang: Conceptualization, Methodology, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Shenghui Zhao:** Validation, Formal analysis, Writing - review & editing. **Jinjun Liu:** Writing - review & editing. **Jing Yang:** Formal analysis, Writing - review & editing. **Li Liu:** Writing - original draft, Writing - review & editing. **Guilin Chen:** Conceptualization, Methodology, Supervision, Project administration.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (No. 61902068), the Key Research and Development Project of Anhui Province (No. KJ2019ZD44), the Major Special Projects of Anhui Province(No. 201903A06020026), and the Anhui Provincial Natural Science Foundation (No. 1908085MF211).

References

Becht, E., McInnes, L., Healy, J., Dutertre, C. A., Kwok, I., & Ng, L. G. (2019). Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology, 37*(1), 38–44.

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

Bhowmik, M. K., Saha, P., Singha, A., Bhattacharjee, D., & Dutta, P. (2019). Enhancement of robustness of face recognition system through reduced gaussianity in Log-ICA. *Expert Systems with Applications, 116*, 96–107.

Cai, D., He, X., Han, J., & Zhang, H. J. (2006). Orthogonal laplacianfaces for face recognition. *IEEE Transactions on Image Processing, 15*(11), 3608–3614.

Garcia-Vega, S., & Castellanos-Dominguez, G. (2019). Similarity preservation in dimensionality reduction using a kernel-based cost function. *Pattern Recognition Letters, 125*, 318–324.

He, X., Cai, D., Yan, S., & Zhang, H. J. (2005). Neighborhood preserving embedding. In *Proceedings of IEEE international conference on computer vision (ICCV)* (pp. 1208–1213).

He, X., & Niyogi, P. (2004). Locality preserving projections. In *Proceedings of advances in neural information processing systems (NeurIPS)* (pp. 153–160).

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H. J. (2005). Face recognition using Laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(3), 328–340.

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

Krstanović, L., Ralević, N., Zlokolica, V., Obradović, R., Mišković, D., Janev, M., et al. (2016). GMMs similarity measure based on LPP-like projection of the parameter space. *Expert Systems with Applications, 66*, 136–148.

Lee, K. C., Ho, J., & Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(5), 684–698.

Martínez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(2), 228–233.

Pang, Y., Zhou, B., & Nie, F. (2019). Simultaneously learning neighborship and projection matrix for supervised dimensionality reduction. *IEEE Transactions on Neural Networks and Learning Systems, 30*(9), 2779–2793.

Passalis, N., & Tefas, A. (2017). Dimensionality reduction using similarity-induced embeddings. *IEEE Transactions on Neural Networks and Learning Systems, 29*(8), 3429–3441.

Qiao, L., Chen, S., & Tan, X. (2010). Sparsity preserving projections with applications to face recognition. *Pattern Recognition, 43*(1), 331–341.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science (New York, N.Y.), 290*(5500), 2323–2326.

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

Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*(5), 1299–1319.

Song, Y., Nie, F., Zhang, C., & Xiang, S. (2008). A unified framework for semi-supervised dimensionality reduction. *Pattern Recognition, 41*(9), 2789–2799.

Tayalı, H., & Tolun, S. (2018). Dimension reduction in mean-variance portfolio optimization. *Expert Systems with Applications, 92*, 161–169.

Tenenbaum, J., De Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.), 290*(5500), 2319–2323.

The, W. Y., & Roweis, S. T. (2002). Automatic alignment of hidden representations. In *Proceedings of advances neural information processing systems (NeurIPS)* (pp. 841–848).

van der Maaten, L., & Hinton, G. E. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*, 2579–2605.

van der Maaten, L., Postma, E., & Herik, H. (2009). Dimensionality reduction: A comparative review. URL http://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf

Wang, A., Chen, G., Yang, J., Zhao, S., & Chang, C. Y. (2016). A comparative study on human activity recognition using inertial sensors in a smartphone. *IEEE Sensors Journal, 16*(11), 4566–4578.

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2006). Distance metric learning for large margin nearest neighbor classification. In *Proceedings of advances in neural information processing systems (NeurIPS)* (pp. 1473–1480).

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on machine learning (ICML-04)* (p. 106).

Yan, S., Xu, D., Zhang, B., Zhang, H. J., Yang, Q., & Lin, S. (2007). Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(1), 40–51.

Yang, L., Gong, W., Gu, X., Li, W., & Liang, Y. (2008). Null space discriminant locality preserving projections for face recognition. *Neurocomputing, 71*(16–18), 3644–3649.

Yang, W., Wang, K., & Zuo, W. (2012). Fast neighborhood component analysis. *Neurocomputing, 83*, 31–37.

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

Yuan, F., Xia, X., Shi, J., Li, H., & Li, G. (2017). Non-linear dimensionality reduction and Gaussian process-based classification method for smoke detection. *IEEE access : practical innovations, open solutions, 5*, 6833–6841.

Zhang, H., Nie, F., Zhang, R., & Li, X. (2018). Auto-weighted 2-dimensional maximum margin criterion. *Pattern Recognition, 83*, 220–229.

Zhao, H., Wang, Z., & Nie, F. (2018). Adaptive neighborhood MinMax projections. *Neurocomputing, 313*, 155–166.

Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications, 67*, 126–139.