Journal of Intelligent & Fuzzy Systems 41 (2021) 2195–2205
DOI:10.3233/JIFS-210891
IOS Press

Local quasi-linear embedding based on kronecker product expansion of vectors

Guo Niu[a] and Zhengming Ma[b,*]
School of Electronics and Information Engineering, Foshan University, Foshan, China
School of Electronics and Information Technology, SunYat-sen University, Guangzhou, China

Abstract. Locally Linear Embedding (LLE) is honored as the first algorithm of manifold learning. Generally speaking, the relation between a data and its nearest neighbors is nonlinear and LLE only extracts its linear part. Therefore, local nonlinear embedding is an important direction of improvement to LLE. However, any attempt in this direction may lead to a significant increase in computational complexity. In this paper, a novel algorithm called local quasi-linear embedding (LQLE) is proposed. In our LQLE, each high-dimensional data vector is first expanded by using Kronecker product. The expanded vector contains not only the components of the original vector, but also the polynomials of its components. Then, each expanded vector of high dimensional data is linearly approximated with the expanded vectors of its nearest neighbors. In this way, the proposed LQLE achieves a certain degree of local nonlinearity and learns the data dimensionality reduction results under the principle of keeping local nonlinearity unchanged. More importantly, LQLE does not increase computation complexity by only replacing the data vectors with their Kronecker product expansions in the original LLE program. Experimental results between our proposed methods and four comparison algorithms on various datasets demonstrate the well performance of the proposed methods.

Keywords: Dimensionality reduction, locally linear embedding, local quasi-linear

1. Introduction

Machine learning is a hot topic in artificial intelligence research, and also has been increasingly and successfully developed in various engineering problems, such as materials engineering [1], satellite control system [36], supply chain networks [3, 7, 8], Bioengineering [5], social engineering [6], energy vehicle [4], cloud model [2] and so on. Researches on machine learning [9, 37] usually face with a huge amount of high-dimensional data to be processed. Although the high-dimensional data could be contain plenty data information, it also includes a large amount of invalid or disturbing information, even causing the curse of dimensionality. Therefore, data dimensionality reduction is getting more and more attention. The task of data dimensionality reduction is to effectively transform the given high-dimensional data set into a meaningful low-dimensional data set. Roughly speaking, the algorithms of dimensionality reduction can be divided into two kinds: linear and nonlinear. Most of traditional algorithms of dimensionality reduction are linear, such as Principal Component Analysis (PCA) [10], Linear Discriminant Analysis (LDA) [11–13] and so on, while the emerging manifold learning algorithms are nonlinear, such as LLE [14], LE [15], HLLE [16], LTSA [17], ISOMAP [18] and so on.

LLE (acronym for Local Linear Embedding), published in Science in 2000, is honored as the first algorithm for manifold learning. LLE claims to be a locally linear but globally nonlinear algorithm, in which each high-dimensional data is linearly approxi-

*Corresponding author. Zhengming Ma, School of Electronics and Information Technology, SunYat-sen University, Guangzhou, China. E-mail: issmzm@mail.sysu.edu.cn.

ISSN 1064-1246/$35.00 © 2021 – IOS Press. All rights reserved.

mated with its nearest neighbors. The approximation coefficients are called local learning patterns. LLE keeps the local linear patterns unchanged during dimensionality reduction. Since the core of local feature preservation is intuitive and effective, LLE is developed in many applications and methods [1–35]. However, with the development of manifold learning, the defects of LLE have been exposed. For example, the original LLE is strict for local selection (neighborhood selection). If the neighborhood is too small, the local geometry structure of manifold cannot be well preserved. If the neighborhood is too large, the local reconstruction error will be too large, causing the algorithm fail. In order to solve this problem, a series of algorithms are proposed, [20–23]. The learning of local linear patterns of LLE is actually a local reconstruction process. The local reconstruction process may cause a relatively large error when the local curvature of the original data is too large. And when the center data of neighborhood is far away from the plane composed of neighbors, the local geometric information also will be lost. Therefore, the improvements of the local reconstruction process focus on finding more reasonable methods to reconstruction weights (approximation coefficients) [26]. LLE linearly reconstructs a high-dimensional data by utilizing its neighborhood data and corresponding reconstruction weights for learning the local features of the high-dimensional data set. It is the first and critical step in LLE. But actually, most of the linear reconstruction is difficult to completely. In other words, the original high-dimensional cannot be linearly reconstructed by its neighbors in some cases. To solve this problem, we propose a local quasi-linear embedding method.

In theory, local linear patterns are just an approximation to local nonlinear patterns and therefore local nonlinear embedding is an intuitive improvement. Inspired by it, we propose a novel nonlinear dimensionality reduction algorithm, named local quasi-linear embedding (LQLE). LQLE utilizes Kronnecker product to nonlinearly expand each high-dimensional data vector, and learns the local geometry structure of data by using the merits of LLE. The novelty of our method can be summarized as follows:1) the proposed method can learn an effective low-dimensional embedded representation for high-dimensional data; 2) the proposed method achieves some form of local nonlinear patterns data and keeps the local nonlinear patterns unchanged during the process of dimensionality reduction; 3) the proposed method does not increase computation complexity by only replacing the data vectors with their Kronecker product expansions in the original LLE program; 4) the proposed method can be easily transplanted to the application fields of LLE.

The remaining part of the paper is organized as follows. In the Section 2, some related works are introduced. Our mainly work are proposed in Section 3, including the extended vectors of data vectors, the local quasi-linearity for high-dimensional data, and the novel LQLE. Section 4 demonstrates the experimental results and analysis. Finally, Section 5 gives the summary of our work.

2. Related work

LLE has been concerned by researchers since it was proposed. It has a high value in research and application. Since it has been proposed, the improvement algorithms on it have emerged in an endless stream. Besides, researches have made detailed overviews of it [19].

2.1. Local linear embedding

The main idea of LLE is to approximate the data point with linear weight of its neighbors and preserve the linear weight in low-dimensional space. Let $X = \{x_1, \cdots, x_N\} \subseteq R^D$ be the N points high-dimensional data set of D dimensions. The purpose of LLE is to find the low-dimensional embedding of X in the embedding space R^d, $(d \ll D)$. We denote the N points low-dimensional embedding by $Y = \{y_1, \cdots, y_N\} \subseteq R^d$.

LLE firstly find the k-nearest neighbors for each data point x_i, $i = 1, \cdots, N$. The neighbor set is defined as $N_i = \{x_{i1}, \cdots, x_{ik}\}$. For each sample x_i, the linear weight $\{w_{ij}, j = 1, \cdots, k\}$ can be computed by solving the following problem:

$$\underset{W}{min} \sum_{i=1}^{N} \left\| x_i - \sum_{j=1}^{k} w_{ij} x_j \right\|_F^2 \qquad s.t. \sum_{j=1}^{k} w_{ij} = 1 \tag{1}$$

After the linear weight $W = \{w_{ij}, j = 1, \cdots, k, i = 1, \cdots, N\}$ is learned, the low-dimensional embedding can be found, which best preserves the geometric properties of the original space. That is to minimize:

$$\min_{W}\sum_{i=1}^{N}\left\|y_i-\sum_{j=1}^{k}w_{ij}y_j\right\|_F^2$$
$$s.t.\sum_{i=1}^{N}y_i=0,\ \frac{1}{N}\sum_{i=1}^{N}y_iy_i^T=I \tag{2}$$

2.2. Improvements of LLE

LLE reduces the dimension of data by preserving the relationship between data point and neighborhood locally, so the selection of neighborhood plays an important role in the performance of the algorithm. The Euclidean distance is used directly in the original LLE to find the neighbors. Some later works attempt to introduce other method to measure the distance for selecting neighbors.

Kernel Local Linear Embedding (KLLE) [20] proposed by Dennis Deste finds the k-nearest neighbor point with kernel distance (linear kernel) instead of Euclidean distance in original LLE algorithm. The KLLE proved the feasibility of substituting Euclidean distance with kernel distance. Varini believed that selecting neighbor points with Euclidean distance would make two points with short Euclidean distance but long geodesic distance select as neighborhood points. To avoid this situation, LLE with geodesic distance (ISO-LLE) [21] was proposed, which searches for the neighbor points with geodesic distance. Experiments show that ISO-LLE can solve the above situation effectively. Weighted Local Linear Embedding (WLLE) [22] adjusts the LLE algorithm based on weighted distance [23], which gives the original data with high density a smaller weight scaling while those with low density a larger weight scaling. WLLE works well in preventing the selected neighbor points from being over congregating in uneven distributed locality. Rank-Order Distance [24] was proposed to solve the problem that different sparsity in different clusters leads to inaccurate image classification in face recognition. Rank-Order Distance was applied in the selection of neighbors in LLE. Similar to the before algorithm, LLE based on Rank-Order Distance [25] was proposed, and the application of Rank-Order Distance also solve the problem of uneven density to some extend.

Some work focus on use different reconstruction weights to improve LLE. As the weight matrix is set, one direction of improvement is to find more reasonable methods to reconstruction weights. Nonlinear Embedding Preserving Multiple Local-Linearities (NEML) [26] proposed a nonlinear pattern with multiple weights and proved that multiple sets of local construction weights that are approximately optimal and can be used to improve the stability of LLE which can improve the performance of it.

2.3. Out-of-sample extension of LLE

The Original algorithm of LLE didn't provide an effective solution to out-of-sample problem. To solve this problem, NPE [27], NPP [28] and ONPP [29] are proposed. These algorithms tried to match the result of LLE with linear projection from high-dimensional space to low-dimensional space. Given the high-dimensional data set $X \subseteq R^D$, these algorithm attempt to find out its low-dimensional embedding $Y \subseteq R^d$ and the explicit projection from R^D to R^d at the same time. That is to find a projection matrix $V \in R^{D\times d}$ that satisfy $Y = V^TX$. To solve this problem, the optimization problem 2 should be transformed:

$$\min_{W}\sum_{i=1}^{N}\left\|V^Tx_i-\sum_{j=1}^{k}w_{ij}V^Tx_j\right\|_F^2$$
$$\begin{cases} s.t.\sum_{i=1}^{N}V^Tx_i=0,\ \frac{1}{N}\sum_{i=1}^{N}V^Tx_ix_i^TV=I \\ \quad\text{(NPP,NPE)} \\ s.t.\sum_{i=1}^{N}V^Tx_i=0,\ \frac{1}{N}\sum_{i=1}^{N}V^TV=I \\ \quad\text{(ONPP)} \end{cases} \tag{3}$$

After training, the out-of-sample points x_o can be directly projected into low-dimensional space by $y_o = V^Tx_o$.

Based on these work, Hong Qiao proposed a non-linear projection method NPPE and its simplified algorithm SNPPE [23], which expanded the projection coefficients to vector polynomials. The coefficients of y_i are not simply linear combination of coefficients of x_i but a polynomial instead. The form of projection $y_i = V^Tx_i$ can be rewritten as $y_i = V_p^Tx_{ip}$, x_{ip} is defined as:

$$x_{ip}=\begin{cases} x_i \\ x_i\otimes x_i \\ \vdots \\ \underbrace{x_i\otimes\cdots\otimes x_i}_{p} \end{cases}$$

where $\otimes$ stands for the Kronecker product defined on matrices. And V_p means V being extended into corresponding row numbers. The idea of NPPE is similar to NPE or NPP expect $x_i p$ should be computed from x_i before embedding or projection.

In recent years, besides the improvements to LLE algorithm, the development of LLE mostly lies in the application of the algorithm [24, 31]. Proposed a sparse linear embedding based on LLE and ONPP. Besides manifold learning itself, another important branch of LLE application is image processing, such as feature extraction [25, 32]. Moreover, the value of LLE algorithm is not limited in algorithm research, but also showed in various engineering problems, which had proved its practicability [1, 33–35].

3. Local quasi-linear embedding (LQLE)

Given the high-dimensional dataset $X = [x_1 \cdots x_N] \in R^{D\times N}$, the task of dimensionality reduction is to get a low-dimensional dataset $Y = [y_1 \cdots y_N] \in R^{d\times N}$, where y_n is the low-dimensional representation of x_N and $n = 1, \cdots, N$, $(d \ll D)$. In this paper, we proposed a local quasi-linear embedding (preserving) algorithm for data dimensionality reduction. Note: In the section, the dataset is represented by a matrix, and each column of it represents a data vector.

3.1. The extended vectors of data vectors

Let $x = [x^{(1)} \cdots x^{(D)}]^T \in R^D$ denotes a data vector, in which $x^{(i)}$ represents the ith component of x, $i = 1, \cdots, D$, then $\underbrace{x_i \odot \cdots \odot x_i}_{p} \in R^{D_p}$ represent the p-order simplified Kronecker product of x, where $D_p = D + \frac{D!}{p!(D-p)!}$, $p \geq 2$, By 'simplified', it means to remove the same component. For example,

$$x \otimes x = \begin{bmatrix} x^{(1)} \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} \\ x^{(2)} \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} x^{(1)}x^{(1)} \\ x^{(1)}x^{(2)} \\ x^{(2)}x^{(1)} \\ x^{(2)}x^{(2)} \end{bmatrix}$$

$$\Longrightarrow_{simplified}$$

$$x \odot x = \begin{bmatrix} x^{(1)}x^{(1)} \\ x^{(2)}x^{(2)} \\ x^{(1)}x^{(1)} \end{bmatrix} \tag{4}$$

The p-order extended vector x^p of x is then defined as

$$x = \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(D)} \end{bmatrix} \in R^D \Longrightarrow x^p = \begin{bmatrix} x \\ x \odot x \\ \vdots \\ \underbrace{x_i \odot \cdots \odot x_i}_{p} \end{bmatrix} \in R^{D_p} \tag{5}$$

where $D_p = D + \sum_{k=2}^{p} \frac{D!}{p!(D-k)!}$, $p \geq 2$. Furthermore, we set $x^{(1)} = x$, then $D_p = D$.

3.2. The local quasi-linearity for high-dimensional data

For all x_n, let $x_{n1}, \cdots, x_{nK}$ be its K nearest neighbors in the given data set, and then x_n^p and $x_{n1}^p, \cdots, x_{nK}^p$ represent their extended vectors respectively. In LQLE, x_n^p is linearly approximated by

$$\min_{w_n} \left\| x_n^p - \sum_{k=1}^{K} w_{nk} x_{nk}^p \right\|_F^2$$

$$s.t. \sum_{k=1}^{K} w_{nk} = 1 \tag{6}$$

where $w_n = [w_{n1} \cdots w_{nK}]^T$. It is clear that w_n reflects some nonlinear relation between x_n and its nearest neighbors $\{x_{n1}, \cdots, x_{nK}\}$.

The derivation of w_n is as follows:

$$\left\| x_n^p - \sum_{k=1}^{K} w_{nk} x_{nk}^p \right\|_F^2 = \left\| -x_n^p + \sum_{k=1}^{K} w_{nk} x_{nk}^p \right\|_F^2$$

$$= \left\| \left[x_{n1}^p - x_n^p \cdots x_{nK}^p - x_n^p \right] \begin{bmatrix} w_{n1} \\ \vdots \\ w_{nK} \end{bmatrix} \right\|_F^2$$

$$= \left\| \bar{X}_n^p w_n \right\|_F^2 = w_n^T (\bar{X}_n^p)^T \bar{X}_n^p w_n \tag{7}$$

where $\bar{X}_n^p = \left[x_{n1}^p - x_n^p \cdots x_{nK}^p - x_n^p \right] \in R^{D_p \times K}$. Then Eq.(6) can be solved by minimizing

$$\varphi(w_n) = w_n^T (\bar{X}_n^p)^T \bar{X}_n^p w_n + \lambda (w_n^T \Gamma_K - 1) \tag{8}$$

By using the Lagrange method, i.e., taking the partial derivative of the above function (8) with respect to

w_n and λ respectively, we get

$$w_n = \frac{\left((\bar{X}_n^p)^T \bar{X}_n^p\right)^{-1} \Gamma_K}{\Gamma_K \left((\bar{X}_n^p)^T \bar{X}_n^p\right)^{-1} \Gamma_K} \tag{9}$$

where $\Gamma_K = [1 \cdots 1]^T \in R^K$. It should be noted that $\bar{X}_n^p$ seems to be a huge matrix, while $(\bar{X}_n^p)^T \bar{X}_n^p \in R^{K \times K}$ is a smaller matrix. And $\bar{X}_n^p$ is generated by known operations with x_n and $\{x_{n1}, \cdots, x_{nK}\}$ (see Eq.(7) and (5)). Therefore, it is unnecessary to generate $\bar{X}_n^p$ in practical operations. We can computed $(\bar{X}_n^p)^T \bar{X}_n^p$ directly and take inverse of it.

LQLE takes into consideration the fitting relationship between the high order component, so the computed W in LQLE is different from the weights matrix in LLE.

3.3. Data dimensionality reduction based on local quasi-linear representation

Corresponding to the local data x_n and $\{x_{n1}, \cdots, x_{nK}\}$ in high-dimensional space, the dimension-reduced data in low-dimensional space are y_n and $\{y_{n1}, \cdots, y_{nK}\}$. LQLE proposed in this paper preserves the same locally linear pattern w_n between y_n and $\{y_{n1}, \cdots, y_{nK}\}$, i.e.,

$$\begin{aligned} &\min_{y_n, y_{n1}, \cdots, y_{nK}} \left\| y_n^p - \sum_{k=1}^{K} w_{nk} y_{nk}^p \right\|_F^2 \\ &s.t. \sum_{k=1}^{K} w_{nk} = 1 \end{aligned} \tag{10}$$

Making further derivation,

$$\begin{aligned} &\left\| y_n^p - \sum_{k=1}^{K} w_{nk} y_{nk}^p \right\|_F^2 \\ &= \left\| \left[y_n^p \; y_{n1}^p \cdots y_{nK}^p \right] \begin{bmatrix} 1 \\ -w_{n1} \\ \vdots \\ -w_{nK} \end{bmatrix} \right\|_F^2 \\ &= \left\| Y_n^p \begin{bmatrix} 1 \\ -w_n \end{bmatrix} \right\|_F^2 = \left\| Y^p S_n \begin{bmatrix} 1 \\ -w_n \end{bmatrix} \right\|_F^2 \\ &= \left\| Y^p L_n \right\|_F^2 \\ &= tr\left(Y^p L_n L_n^T (Y^p)^T \right) \end{aligned} \tag{11}$$

where $Y_n^p = \left[y_n^p \; y_{n1}^p \cdots y_{nK}^p \right]$, $Y^p = \left[y_1^p \cdots y_N^p \right]$, and $L_n = S_n \begin{bmatrix} 1 \\ -w_n \end{bmatrix} \in R^N$. Here S_n is selection matrix, which makes $Y_n^p = Y^p S_n$. Eq. (11) only considers the local nonlinear preservation of one sample data. Then we take into account the local nonlinear preservation of all data, and get

$$\begin{aligned} &min \sum_{n=1}^{N} \left\| y_n^p - \sum_{k=1}^{K} w_{nk} y_{nk}^p \right\|_F^2 \\ &= min \sum_{n=1}^{N} tr\left(Y^p L_n L_n^T (Y^p)^T \right) \\ &= min\, tr \left(Y^p \left(\sum_{n=1}^{N} L_n L_n^T \right) (Y^p)^T \right) \\ &= min\, tr\left(Y^p L L^T (Y^p)^T \right) \\ &s.t., Y^p (Y^p)^T = I_d \end{aligned} \tag{12}$$

Notes that although Y^p is a huge matrix, we only need a low dimension data matrix Y consist of the first few d rows of Y^p. Therefore, we just need to find out the matrix constituted by the d smallest eigenvectors corresponding to the d smallest eigenvalue of LL^T. The learning problem of (12) is a typical generalized Rayleigh quotient problem. Therefore, Eq. (12) can be learned in a closed form, i.e, Y can be easily obtained by the eigenvalue decomposition of LL^T.

4. Experimental results

In this section, we verify the effectiveness of the proposed algorithm by conducting the dimensionality reduction experiments and classification experiments with LLE [14], WLLE [22], and ONPP [29] algorithms on both toy and real-world data sets. Among these algorithms, LLE [14] is the classical local linear embedding algorithm, WLLE [22] is the representative improved LLE algorithm based on neighborhood selection, and ONPP [29] is the typical out-of-sample extension of LLE.

4.1. Datasets description and experimental setting

Toy datasets: two data sets are generated by randomly selecting 2000 data samples from the Swiss Roll and Swiss Hole, respectively.

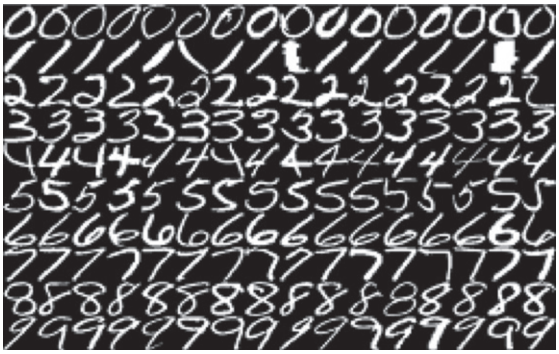

Fig. 1. Partial data of USPS data set.

Fig. 2. Partial data of Binary Alphadigits data set.

USPS [38]: a famous handwritten digit dataset (Fig. 1), contains handwritten images of 10 classes corresponding to the digits 0 to 9, with 1100 sample images per class. The size of each sample image is 16×16. We randomly select 200 images from each class to construct the experimental dataset with a total of 2000 images.

Binary Alphadigits [39]: another handwritten dataset(Fig. 2), contains binary 20×16 digits of 0 through 9 and capital A through Z, 39 examples of each class. We only use the capital A through Z, and randomly select 30 images from each class to construct the experimental data set with a total of 780 images.

COIL20 [40]: an object data set(Fig. 3), contains 20 objects taken from different angles, and 72 images per object. Each image is uniformly processed as the size of 32×32.

Table 1 lists the parameter settings of the three real-world datasets in the experiments.

4.2. Dimensionality reduction on toy datasets

We first compare the LQLE algorithm with the LLE algorithms on two toy data sets. The number of neighbor is set to be $K = 10$ and the polynomial order is set to be $p = 2$. The experiment results are showed in Fig. 4. It can be seen that the LLE algorithm has a poor performance when $K = 10$. But the proposed LQLE would achieve a better result. It means that our LQLE introduces more local information to data dimensionality reduction.

Fig. 3. Partial data of COIL20 data set.

Table 1
Parameters settings of data sets

Datasets Parameter	USPS	Binary Alphadigits	COIL20
Number	2000	780	1400
Original Dimensionality	256	320	256
Number of Categories	10	26	20
Number of Train Set	1000	390	700
Number of Testing Set	1000	390	700

4.3. Classification experimental results

In the classification experiments, we adopt the KNN classifier with $K = 8$ on reduced-dimensional data sets, and use the cross-validation method for computing the classification accuracy. We randomly select 50% of the dimensionality reduced data from each category of the data set as the training set, and the remaining 50% as the test set. Each process repeats 5 times and the average of the classification accuracy of the 5 tests is used as the final. Generally, the number of neighborhood points for each compared algorithm is selected according to the principle that the number of neighborhood points is about 1% of the number of data in the corresponding data set. $Dim.$ denotes the dimension of the dimensionality reduction results. The value of $Dim.$ is setting from 2 to 100 on USPS and Binary Alphadigits data sets, and is setting from 2 to 200 on COIL20 data set. The polynomial order in LQLE is set to be $p = 2$.

Table 2 lists the average classification results of four comparison algorithms on the USPS data set. In order to show more direct observation, the average classification results of all experiments on USPS data set are also shown in Fig. 5 for visualization. Seen from them, when Dim. is less than 60, ONPP

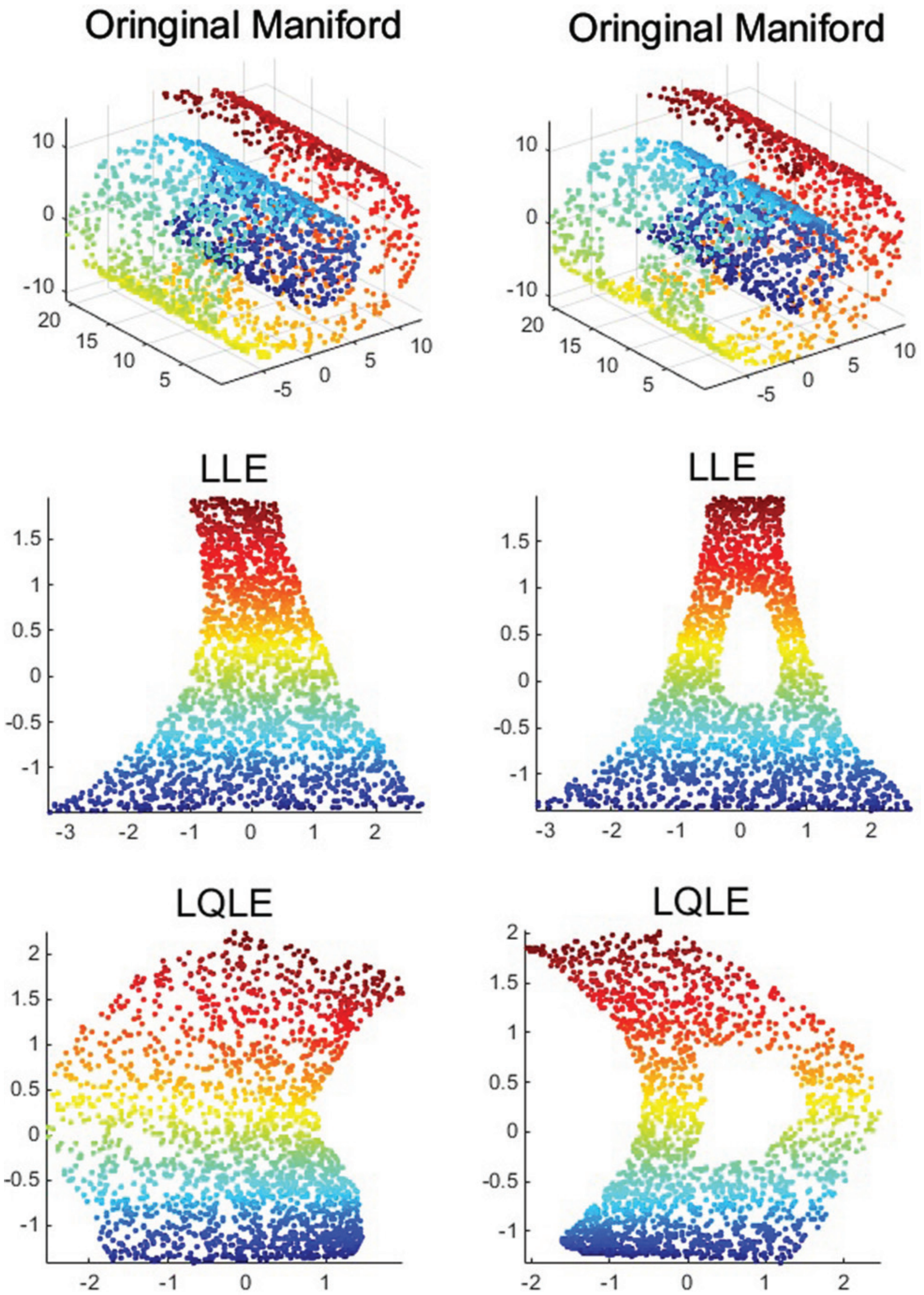

Fig. 4. The dimensionality reduction results of LLE and LQLE on toy dataset.

has the worst performances on USPS, which maybe caused by strict global linear constraints. In most cases, WLLE has the best results, our LQLE is second, and the last is LLE. But when Dim. is less than 15, our method works better than WLLE. When the dimension of dimensionality reduction is less than 50, the performance of all the algorithms almost become better as the Dim. increasing. Overall, the performances of our LQLE are better than LLE. It proves that the proposed method is an improvement to LLE. And with increasing of the dimension, the performances of LQLE maintain a certain degree of stability. It means that our algorithm is robust.

Table 2
The average classification results of four compared algorithms on USPS data set. The classification performances of four compared algorithms versus different dimension (*Dim*.) on USPS data set are shown here. LQLE is the proposed algorithm.

Dim.	LLE	WLLE	ONPP	LQLE
2	0.601	0.398	0.159	0.579
5	0.743	0.647	0.251	0.705
10	0.73	0.747	0.323	0.77
20	0.697	0.828	0.382	0.816
30	0.729	0.821	0.456	0.797
40	0.734	0.842	0.577	0.807
50	0.745	0.825	0.691	0.786
60	0.756	0.827	0.781	0.781
80	0.701	0.819	0.866	0.778
100	0.626	0.801	0.898	0.779
Average	0.7062	0.7555	0.5384	**0.7598**

Table 3
The average classification results of four compared algorithms on Binary Alphadigits data set. The classification performances of four compared algorithms versus different dimension (*Dim*.) on Binary Alphadigits data set are shown here. LQLE is the proposed algorithm.

Dim.	LLE	WLLE	ONPP	LQLE
2	0.78	0.0986	0.3357	0.78
5	0.8086	0.2643	0.73	0.79
10	0.8757	0.5	0.8814	0.8743
20	0.9429	0.6571	0.95	0.93
30	0.9486	0.7386	0.9557	0.9471
40	0.9586	0.7629	0.9557	0.9629
50	0.9586	0.7971	0.9629	0.95
60	0.9543	0.85	0.9657	0.9586
80	0.9471	0.95	0.9686	0.9629
100	0.91	0.9529	0.9714	0.95
Average	0.9084	0.6572	0.8677	**0.9106**

Table 3 exhibits the average classification results of the four comparison algorithms on the Binary Alphadigits data set when the dimension of the dimensionality reduction is changing. Fig. 6 shows the classification performances of all compared methods on Binary Alphadigits data set. Obviously, when the dimension of dimensionality reduction is less than 60, WLLE has the worst performances on Binary Alphadigits. It means that on this data set, using weighted distance in WLLE cannot improve the performance of LLE. When Dim. is less than 10, ONPP works worse than LLE and our LQLE. And when Dim. is larger than 40, LQLE works better than LLE. And overall, with increasing of the dimension, LQLE still maintains better performances. And all the average results of the four compared methods reported in the last column of Table 3 further illustrate the robustness of our algorithm.

Table 4 lists the average classification results of the four comparison algorithms on the COIL20 data set when the dimension of the dimensionality reduction is changing. And Fig. 7 shows the classification performances of all compared algorithms on COIL20. From them, we can observed that different from the results on the Binary Alphadigits, when Dim. is less than 70, WLLE almost has the best performances. Follow it, our algorithm also works well on COIL20 data set. But the performances of WLLE are not stable as the Dim. increasing. Overall, the performances of the proposed LQLE are better than LLE in most cases. And LQLE still maintains better performances on COIL20 as the Dim. increasing. Furthermore, the total average results of the four compared methods reported in the last column of Table 4 show that our algorithm has the best result. To sum up, our LQLE

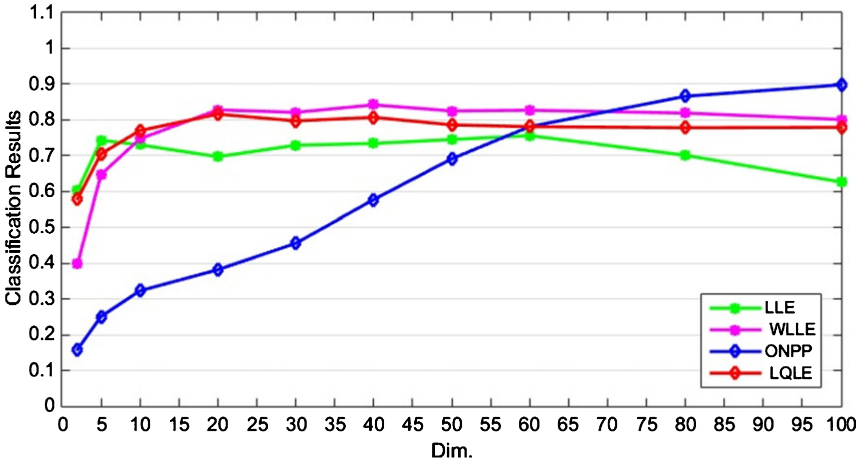

Fig. 5. The average classification results of four compared algorithms on USPS data set. LQLE is the proposed algorithm.

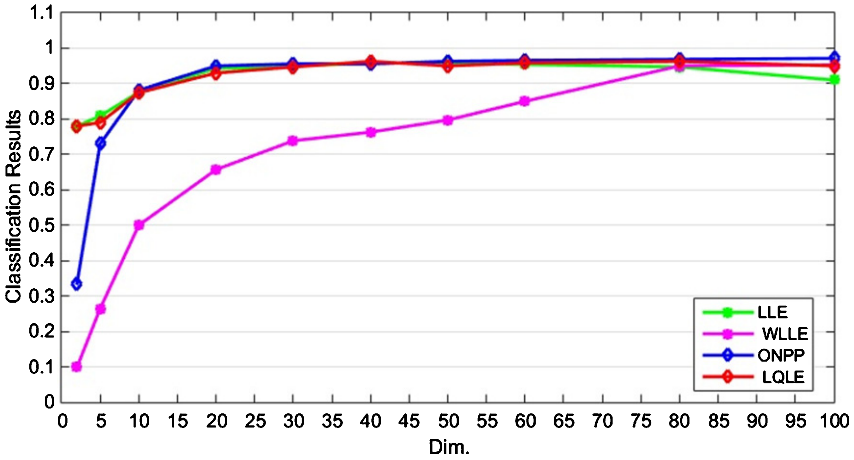

Fig. 6. The average classification results of four compared algorithms on Binary Alphadigits data set. LQLE is the proposed algorithm.

Table 4
The average classification results of four compared algorithms on COIL20 data set. The classification performances of four compared algorithms versus different dimension (*Dim.*) on COIL20 data set are shown here. LQLE is the proposed algorithm

Dim.	LLE	WLLE	ONPP	LQLE
2	0.3513	0.3487	0.0667	0.3308
5	0.5128	0.5205	0.1077	0.5231
10	0.6179	0.6359	0.1128	0.6
20	0.5897	0.6462	0.2026	0.6462
30	0.5897	0.6641	0.2949	0.6205
40	0.6	0.6615	0.341	0.6231
50	0.5974	0.641	0.3974	0.6231
60	0.6103	0.6385	0.4667	0.6308
80	0.5846	0.6179	0.5564	0.6026
100	0.5846	0.5641	0.6256	0.6128
150	0.6179	0.4487	0.7026	0.5769
200	0.559	0.3872	0.7026	0.5744
Average	0.5679	0.5645	0.3814	**0.5804**

algorithm can maintain better performance on three different data sets. It further proves that the proposed algorithm has good robustness.

5. Conclusions

In the paper, we propose a novel nonlinear dimensionality reduction, named local quasi-linear embedding (LQLE). The proposed method nonlinearly expands each data vector by using Kronecker product and then performs local linear embedding for these nonlinearly-expanded data vectors. LQLE explores the local nonlinear structure of data and learns an effective low-dimensional embedded representation for the high-dimensional data. In general, the nonlinear improvement to the local linear pat-

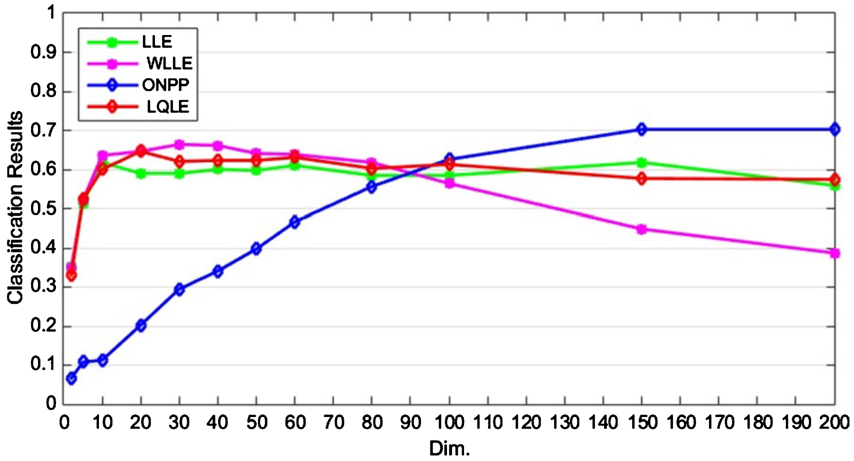

Fig. 7. The average classification results of four compared algorithms on COIL20 data set. LQLE is the proposed algorithm.

terns of LLE may lead to a significant complexity. Compared with LLE, LQLE hardly increases any computational complexity and inherits the merits of LLE. Therefore, it can be easily transplanted to the application fields of LLE. We also conduct the dimensionality reduction and classification experiments on both toy and real-world data sets to demonstrate the effectiveness of the proposed method.

Acknowledgment

This work is supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515111143 and by the Natural Science Foundation of China under Grant 62006042.

References

[1] E. Lopez, D. Gonzalez, J.V. Aguado, E. Abisset-Chavanne, E. Cueto, C. Binetruy and F. Chinesta, A Manifold Learning Approach for Integrated Computational Materials Engineering, *Archives of Computational Methods in Engineering* **25**(1) (2018), 59–68.

[2] C. Zhang, G.D. Tian, A.M. Fathollahi-Fard, W.J. Wang, P. Wu and Z.W. Li, Interval-Valued Intuitionistic Uncertain Linguistic Cloud Petri Net and Its Application to Risk Assessment for Subway Fire Accident, *IEEE Transactions on Automation Science and Engineering* (2020). DOI: 10.1109/TASE. 2020.3014907

[3] A.M. Fathollahi-Fardab, A. Ahmadia and S.M.J. Mirzapour Ale-Hashemac, Sustainable Closed-loop Supply Chain Network for an Integrated Water Supply and Wastewater Collection System under Uncertainty, *Journal of Environmental Management* **275**(111277) (2020).

[4] H.J. Yua, H.L. Dai, G.D. Tian, B.B. Wu, T.Z. Zhang, A.M. Fathollahi-Fard and Q. He and H. Tang, Key technology and application analysis of quick coding for recovery of retired energy vehicle battery, *Renewable and Sustainable Energy Reviews* **135**(110129) (2021).

[5] A.M. Fathollahi-Fard, M. Hajiaghaei-Keshteli and R. Tavakkoli-Moghaddam, Red deer algorithm (RDA): a new nature-inspired meta-heuristic, *Soft Computing* **24** (2020), 14637–14665.

[6] A.M. Fathollahi-Fard, M. Hajiaghaei-Keshteli and R. Tavakkoli-Moghaddam, The social engineering optimizer (SEO), *Engineering Applications of Artificial Intelligence* **72** (2018), 267–293.

[7] J. Moosavi, L.M. Naeni and A.M. Fathollahi-Fard, Blockchain in supply chain management: a review, bibliometric, and network analysis, *Environmental Science and Pollution Research* (2021), 1–15.

[8] M.R. Islam, S.M. Ali, A.M. Fathollahi-Fard and G. Kabir, A novel particle swarm optimization-based grey model for the prediction of warehouse performance, *Journal of Computational Design and Engineering* **8**(2) (2021), 705–727.

[9] I. Izonin, N. Kryvinska, R. Tkachenko, K. Zub and P. Vitynskyi, An Extended-Input GRNN and its Application, *Procedia Computer Science* **160** (2019), 578–583.

[10] M. Turk and A. Pentland, Eigenfaces for recognition, *J Cognit Neurosci* **3**(1) (1991), 71–86.

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

[12] C. Fraley and A.E. Raftery, Model-based clustering, discriminant analysis, and density estimation, *Journal of the American Statistical Association* **97**(458) (2002), 611–631.

[13] Q. Liu, H. Lu and S. Ma, Improving kernel Fisher discriminant analysis for face recognition, *IEEE Transactions on Circuits and Systems for Video Technology* **14**(1) (2004), 42–49.

[14] S.T. Roweis and L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* **290**(5500) (2000), 23230–2326.

[15] M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computing* **15**(6) (2003), 1373–1396.

[16] D.L. Donoho and C. Grimes, Hessian Eigenmaps: Locally Linear Embedding Techniques for High Dimensional Data, *Proceedings of the National Academy of Sciences of the United States of America* **100**(10) (2003), 5591–5596.

[17] Z. Zhang and H. Zha, Principal manifold and nonlinear dimensionslity reduction via tangent space alignment, *SAIM Journal on Scientific Computing* **26**(1) (2004), 313–338.

[18] J.B. Tenenbaum, V.d. Silva, et al., A global geometric framework for nonlinear dimensionality reduction, *Science* **290**(5500) (2000), 2319–2323.

[19] J. Chen and Z.M. Ma, Locally Linear Embedding: A Review, *International Journal of Pattern Recognition and Artificial Intelligence* **25**(7) (2011), 985–1008.

[20] D. De Coste, Visualizing Mercer kernel feature spaces via kernelized locallylinear embeddings, *In Proceedings of the Eighth International Conference on Neural Information Processing*, Shanghai, China, (2001), 14–18.

[21] C. Varini, A. Degenhard and T.W. Nattkemper, ISOLLE: LLE with geodesic distance, *Neurocomputing* **69**(13) (2006), 1768–1771.

[22] Y. Pan, S.S. Ge and A. Al Mamun, Weighted locally linear embedding for dimension reduction, *Pattern Recognition* **42**(5) (2009).

[23] C.Y. Zhou and Y.Q. Chen, Improving nearest neighbor classification with cam weighted distance, *Pattern Recognition* **39**(4) (2006), 635–645.

[24] C. Zhang, F. Wen and J. Sun, A rank-order distance based clustering algorithm for face tagging, *IEEE Conference on Computer Vision and Pattern Recognition* (2011), 481–488.

[25] X.L. Sun and Y.G. Lu, Locally Linear Embedding based on Rank-order Distance, *International Conference on Pattern Recognition and Methods* (2016), 162–169.

[26] J. Wang and Z.Y. Zhang, Nonlinear embedding preserving multiple local-linearities, *Pattern Recognition* **43**(4) (2010), 1257–1268.

[27] X. He, D. Cai, S. Yan and H. Zhang, Neighborhood preserving embedding, *Proceedings in International Conference on Computer Vision (ICCV)* **2** (2005), 1208–1213.

[28] Y. Pang, L. Zhang, Z. Liu, N. Yu and H. Li, Neighborhood preserving projections (NPP): A novel linear dimension reduction method, *Proceedings in ICIC* (1) (2005), 117–125.

[29] E. Kokiopoulou and Y. Saad, Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(12) (2007), 2143–2156.

[30] H. Qiao and P. Zhang, et al., An Explicit Nonlinear Mapping for Manifold Learning, *IEEE Transactions on Cybernetics* **43**(1) (2013), 51–63.

[31] Z.H. Lai, W.K. Wong, Y. Xu, J. Yang and D. Zhang, Approximate Orthogonal Sparse Embedding for Dimensionality Reduction, *IEEE Transactions on Neural Networks and Learning Systems* **27**(4) (2015), 723–735.

[32] M.D. Wang, J. Yu, L.J. Niu and W.D. Sun, Unsupervised feature extraction for hyperspectral images using combined low rank representation and locally linear embedding, *IEEE ICASSP* (2017).

[33] M. Ma, X.F. Chen, X.L. Zhang, B.Q. Ding and S.B. Wang, Locally Linear Embedding on Grassmann Manifold for Performance Degradation Assessment of Bearings, *IEEE Transactions on Reliability* **66**(2) (2017), 467–477.

[34] Y.H. Cheng, B. Jiang, N.Y. Lu, T. Wang and Y. Xing, Incremental locally linear embedding- based fault detection for satellite attitude control systems, *Journal of The Franklin Institute* **353**(1) (2016), 17–36.

[35] Y.H. Liu, Y.S. Zhang, Z.W. Yu and M. Zeng, Incremental supervised locally linear embedding for machinery fault diagnosis, *Engineering Applications of Artificial Intelligence* **50** (2016), 60–70.

[36] Z. Qu, K. Xu, Z.G. Chen, X. He, Y.H. Xie, M.M. Liu, F. Li and S. Han, Fault Detection Method of Luojia1-01 Satellite Attitude Control System Based on Supervised Local Linear Embedding, *IEEE Access* **7** (2019), 105489–105502.

[37] S. Zhang, Z.M. Ma and W. Gan, Dimensionality reduction for tensor data based on local decision marginmaximization, *IEEE Transactions on Image Processing* **30** (2020), 234–248.

[38] https://cs.nyu.edu/roweis/data/.

[39] http://www.cs.toronto.edu/roweis/data.html.

[40] http://dataju.cn/Dataju/web/datasetInstanceDetail/62.