Incremental LLE Based on Back Propagation Neural Network

Yansheng Zhang [1, 2], Dong Ye[1], and Yuanhong Liu[2, *], Jianjun Xu[2]

[1]School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin, 150001, P. R. China;
[2]School of Information and Electrical Engineering, Northeast Petroleum University, Daqing, China.

*Corresponding author. E-mail: 39522496@qq.com

Abstract. Locally Linear Embedding (LLE) algorithm is one of promising NonLinear Dimensionality Reduction (NLDR) method for feature extraction. Like most NLDR algorithms, LLE operates in a batch or off-line mode, in other words, for newly coming samples, the old data augmented by the new samples must be completely recalculated by LLE algorithm, which is computationally intensive. Back propagation neural network (BP) is a nonlinear mapping method, and it can learn all the information of a dataset, further, when BP is trained well, it is effective to predict new data. Hence, in this paper, BP is combined with LLE (BPLLE) to deal with out-of-sample data. Four synthetic datasets and two real datasets are given to demonstrate that BPLLE is more valid than several classical incremental LLE algorithms.

Keywords: Dimension reduction; back propagation neural network; incremental LLE; nonlinear mapping.

1. Introduction

In the present world, most of signals are high-dimensional. It is widely acknowledged that high-dimensional signals containing amount of redundant information is difficult to be analysed and diagnosed in the original space [1, 2, 3, 4]. Consequently, it is necessary to project high-dimensional signals into a low-dimensional feature space. As an example, we sample 20000 points in each period from a rotating machine, that is, the run state of the machine in a period can be described by 20000 points. However, as we all know that most of the points are redundant, only few points can perfectly exhibit the running state of the rotating machine [5].

Recently, researchers have been focused on investigating efficient algorithms to obtain the low-dimensional representations of high-dimensional data [6, 7, 8, 9], and a great quantity of excellent algorithms have been proposed [10, 11]. Among these algorithms, LLE [12, 13] exhibits superior performances in dimension reduction. In LLE algorithm, the local geometry structure of original high-dimensional dataset is preserved in embedding space, which provides a better understanding of the internal structure of high-dimensional dataset. Moreover, LLE can avoid local minima and only few free parameters need to be set. Although LLE has so many advantages, it is inefficient for sequentially coming data.

In order to solve this problem, incremental LLE algorithms [14, 15] have been developed. Usually, they can roughly be divided into two types. One is that only the points whose neighbors are changed will be updated, and the low-dimensional representations of the new samples are obtained by recomputing original LLE [16], which is still inefficient. The other constructs an explicit mapping between a high-dimensional space and a low-dimensional space to deal with the new data. Bengio et al. [17] built a unified framework extending for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In this framework, an approximate function is generated by kernel trick. However, the accuracy of this method is mainly dependent on the density estimation of a raw dataset. Kouropteva et al. [18] proposed a non-parametric incremental LLE i.e., ILLE1 algorithm, where the authors suggested that the new data should be embedded by using a linear translation between the neighbors of a new sample in a high-dimensional space and a low-dimensional space. A similar algorithm ILLE2 raised in literature [19] proposed that the low dimension coordinate of a new data could be acquired by a linear combination of its low-dimensional neighbors. In ILLE1 and ILLE2 algorithms, the decomposition of cost matrix is avoided, but their accuracies are rough, and they are inefficient for searching the neighbors.

2. Previous works

LLE algorithm is firstly proposed by Roweis in 2000, which mainly includes three steps. At the beginning, the neighbors of each point are chosen by a similarity measure method; then the optimal reconstruction weights are calculated by solving a minimum square error function; at the end, the low-dimensional coordinates are obtained by preserving the reconstruction weights in embedding space.

2.1. Locally lineal embedding algorithm

Given a dataset $X=[x_1,x_2,\ldots,x_n]\in R^{D\times n},\ \ x_i\in R^D,i=1,2,,\ldots,n.$ $\mathbf{x}_i$ denotes the ith point, and n is the sample size. LLE maps $\boldsymbol{X}$ into $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_n]\in R^{d\times n}\quad (d\ll D)$, where $\boldsymbol{Y}$ is the low-dimensional coordinate of $\boldsymbol{X}$.

1. Finding the neighbors for each point. Let $X^i=[x_i^1,x_i^2,\ldots,x_i^k]$ denote the neighbors of $\mathbf{x}_i$, where k is the number of neighbors. $\mathbf{X}^i$ is chosen by K-NN method with using the Euclidean distance. The selection of k is a difficult task, and the specific description can refer to literature [21].

2. Calculating the reconstruction weight $W=(W_1,W_2,\ldots,W_n)^{n\times n}$ that best reconstruct each point from its k neighbors, where W_i represents the reconstruction weight of $\boldsymbol{x}_i,i=1,2,\ldots,n$. W_i can be calculated by minimizing a square error function:

$$\boldsymbol{W}_i=arg\ min(\left\|\boldsymbol{x}_i-\sum_{j=1}^{k}w_{ij}\boldsymbol{x}_i^j\right\|_2^2)\quad s\cdot t\quad \sum_{j=1}^{k}w_{ij}=1 \tag{1}$$

The constrained condition can ensure that the solution of Eq. (1) is translation invariance. If x_j is not a neighbor of $\boldsymbol{x}_i$, set $w_{ij}=0$. Employing Lagrange theorem on Eq. (1), the optimum solution can be obtained as below:

$$\boldsymbol{W_i}=\frac{\boldsymbol{G_i^{-1}e}}{\boldsymbol{e^{¢}G_i^{-1}e}}$$

Where $G_i=(x_i-x_i^1,x_i-x_i^2,\ldots,x_i-x_i^k)'(x_i-x_i^1,x_i-x_i^2,\ldots,x_i-x_i^k)$ It is worthy to note that W_i reflects a local geometrical relationship. If this relationship is linear, there will exist a linear mapping between the high-dimensional space and the low-dimensional space, which can be utilized to tackle new data.

3. Finally, computing the low-dimensional coordinate based on the reconstruction weights W by minimizing the cost function:

$$\boldsymbol{Y}=\sum_{i=1}^{n}\left\|\boldsymbol{y}_i-\sum_{j=1}^{n}w_{ij}\boldsymbol{y}_i^j\right\|_2^2\quad s\cdot t\quad \frac{1}{n}\boldsymbol{Y}'\boldsymbol{Y}=\boldsymbol{I},\ \ \boldsymbol{Y}'\boldsymbol{e}=0 \tag{2}$$

Where $\boldsymbol{y}_i^j$ is the jth neighbor of y_i. The constraints in Eq.(2) make the solution of the cost function be invariant to translations and rescaling. Y can be obtained by calculating the bottom $d+1$ eigenvectors of the cost matrix $\boldsymbol{M=(I-W)'(I-W)}$. According to the normalization constraint $\sum_{j=1}^{n}w_{ij}=1$, zero is a trivial solution that should be excluded, hence, the remaining d eigenvectors are the final low-dimensional coordinate $\boldsymbol{Y}$. Furthermore, $\boldsymbol{M}$ is a large sparse symmetric matrix, which is expensive to compute the eigenvalues and eigenvectors, so we should avoid to recalculating the matrix $\boldsymbol{M}$, when deals with new data.

2.2. Two classical incremental LLE algorithms

ILLE1 and ILLE2 are two classical incremental LLE algorithms, which all belong to linear method. The specific description of ILLE1 and ILLE2 are expressed as follows:

ILLE1: Let x_{n+1} denote a newly coming point, $\boldsymbol{X}^{n+1}=[\boldsymbol{x}_{n+1}^{1},\boldsymbol{x}_{n+1}^{2},\dots,\boldsymbol{x}_{n+1}^{k}]$ denote the neighbor of $\boldsymbol{x}_{n+1}$, and $\boldsymbol{Y}^{n+1}=[\boldsymbol{y}_{n+1}^{1},\boldsymbol{y}_{n+1}^{2},\dots,\boldsymbol{y}_{n+1}^{k}]$ represent the low-dimensional neighbor of x_{n+1} . For calculating low-dimensional coordinate $\boldsymbol{y}_{n+1}$, two hypothesises must be satisfied: 1) high-dimensional space is local linear; 2) the points that are close in the high-dimensional space must be close in the low one. Then the linear relationship between the high-dimensional space and the low-dimensional space can be approximately described as $\boldsymbol{Y}^{n+1}=\boldsymbol{A}\boldsymbol{X}^{n+1}$, where $\boldsymbol{A}\in R^{d\times D}$ is an unknown linear transformation matrix. Hence, we can obtain $\boldsymbol{A}$ by $\boldsymbol{A}=\boldsymbol{Y}^{n+1}(\boldsymbol{X}^{n+1})^{\dagger}$, where $(\boldsymbol{X}^{n+1})^{\dagger}$ is a pseudo inverse matrix. According to the above assumptions, y_{n+1} can be calculated by $\boldsymbol{y}_{n+1}=\boldsymbol{A}\boldsymbol{Y}^{n+1}$.

ILLE2: For one thing, we compute the weight, W_{n+1} , which best reconstruct the point $\boldsymbol{x}_{n+1}$ from its k neighbors in high-dimensional space. The reconstruction weight W_{n+1} can be computed by $W_{n+1}=(X)^{\dagger}x_{n+1}$. Then y_{n+1} is found by $\boldsymbol{y}_{n+1}=\sum_{j=1}^{k}w_{(n+1)j}\boldsymbol{y}_{n+1}^{j}=\boldsymbol{Y}^{n+1}\boldsymbol{W}_{n+1}$, where i $\boldsymbol{y}_{n+1}^{j}$ s the low-dimensional representation of $\boldsymbol{x}_{n+1}^{j}$.

Remark 1. **It is noted that, if we remove the constraint, $\sum_{j=1}^{k}W_{(n+1)j}=1$, ILLE1 and ILLE2 are essentially** equivalent.

Let $y1$ and $y2$ represent the output of ILL1 and ILL2, respectively.

ILLE1:

$$\boldsymbol{Y}^{n+1}=\boldsymbol{A}\boldsymbol{X}^{n+1}$$

$$\boldsymbol{A}=\boldsymbol{Y}^{n+1}(\boldsymbol{X}^{n+1})^{\dagger}$$

$$y1=\boldsymbol{A}\boldsymbol{x}_{n+1}=\boldsymbol{Y}^{n+1}(\boldsymbol{X}^{n+1})^{\dagger}\boldsymbol{x}_{n+1}$$

ILLE2:

$$\boldsymbol{x}_{n+1}=\boldsymbol{X}^{n+1}\boldsymbol{W}_{n+1}$$

$$\boldsymbol{W}_{n+1}=(\boldsymbol{X}_{n+1})^{\dagger}\boldsymbol{x}_{n+1}$$

$$y2=\boldsymbol{Y}^{n+1}\boldsymbol{W}_{n+1}=\boldsymbol{Y}^{n+1}(\boldsymbol{X}_{n+1})^{\dagger}\boldsymbol{x}_{n+1}$$

Then $\boldsymbol{y1}=\boldsymbol{y2}$

3. Incremental LLE based on BP

3.1. BPLLE algorithm

Ahmed et al.[22] proposed that a nonlinear mapping could be constructed from the embedding space to input space. Hence, any input data can be represented by the nonlinear mapping with the corresponding low-dimensional coordinate. Hence, it is natural to consider constructing a nonlinear mapping from high-dimensional space to embedding space. BP is a nonlinear mapping algorithm, which can approximate any function with expecting precision, and it can learn all the information from samples. So, it is suitable to employ BP to predict the low-dimensional representations of out-of-sample data. BP consists of input layer, hidden layers and output layer. Each layer may contain a number of neurons. A representative three layers of BP network structure and a neuron [23] are shown in Figure 1. A neuron actually is an activation function. The most commonly used neuron is sigmoidal function, and it can be written as follows:

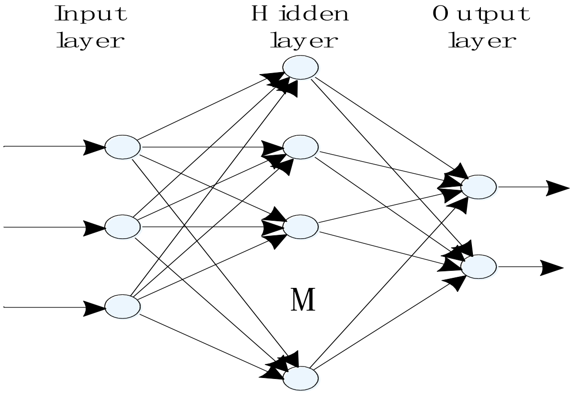
$$f(x)=\frac{1}{1+e^{-x}}$$

Where x is a input. Two neurons i,j are connected by a weight $\tilde{W}_{ij}$. After the network is determined, only the parameters $\tilde{\boldsymbol{W}}$ need to be computed, where $\tilde{\boldsymbol{W}}$ denotes all the weights of BP. Let $\boldsymbol{E}$ is the difference between the expectation value and the predicted value computed by BP, then $\tilde{W}$ can be obtained by minimizing $\boldsymbol{E}$ with an iterative process of gradient descent. Gradient descent can be described as follows:

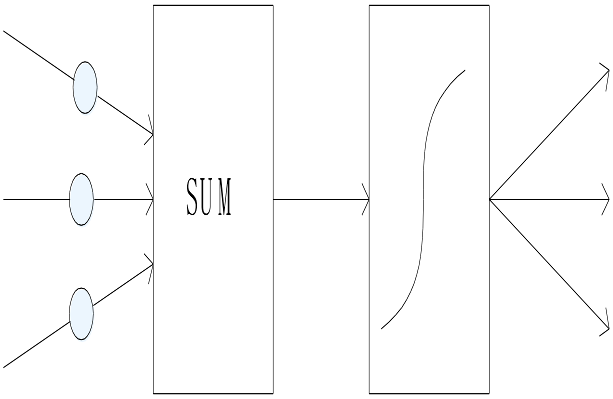
$$\nabla\boldsymbol{E}=\frac{\partial\boldsymbol{E}}{\tilde{\boldsymbol{W}}}$$

Then the new weight can be by update by:

$$\tilde{\boldsymbol{W}}(l+1)=\tilde{\boldsymbol{W}}(l)-\gamma_l\nabla\boldsymbol{E}$$



(a)BP neural network



(b) Neuron

Figure 1: Three layers BP neural network

Where l denotes the l th iteration, γ_l represents step size.

In BPLLE algorithm, LLE is only used to handle the raw dataset to obtain training data, while BP is employed to map the out-of-sample data to a low-dimensional space. The proposed method overcomes the shortcomings of ILLE1 and ILLE2. The specific description of BPLLE is shown in Algorithm 1.

Input: Raw dataset $\boldsymbol{X}$ and out-of-sample $\boldsymbol{x}_{n+1}$.

Output: low-dimensional coordinate $\boldsymbol{y}_{n+1}$ of $\boldsymbol{x}_{n+1}$.

(1) Compute the low coordinate $\boldsymbol{Y}$ of $\boldsymbol{X}$ by LLE, as described in sections 2.1.

(2) Select a suitable BP neural network structure, learning rate, and learning function.

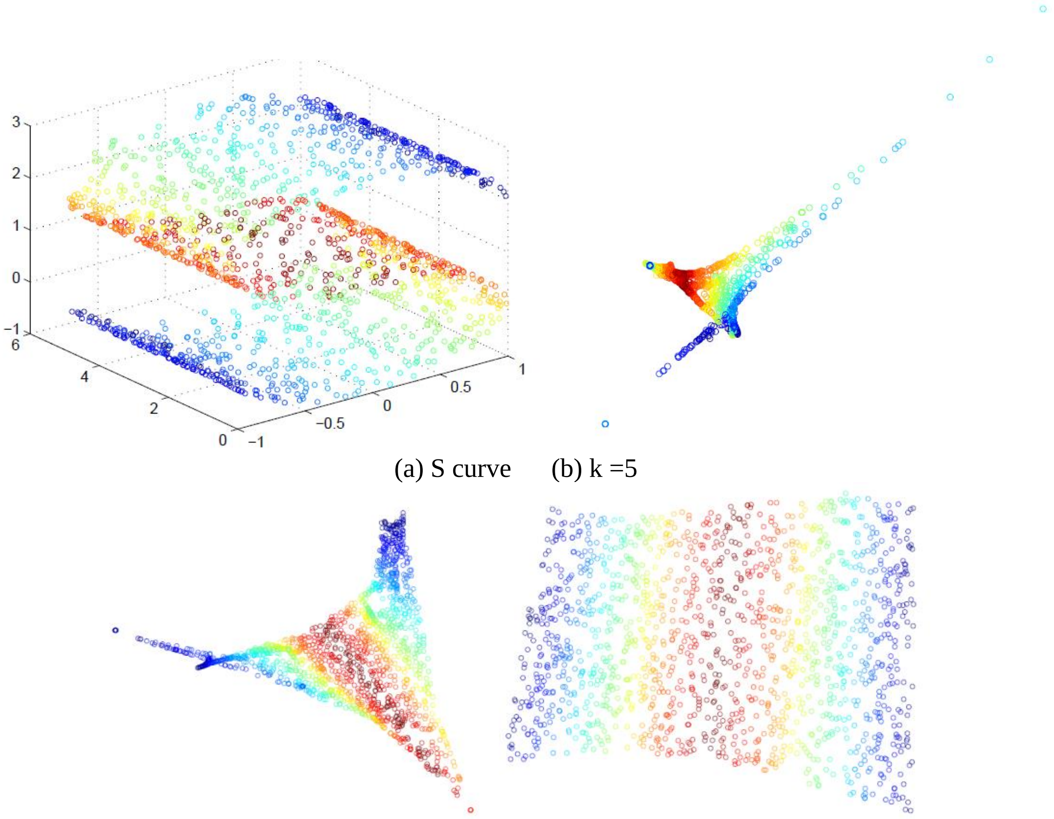
(3) Normalize the training dataset $\boldsymbol{X}$ and its corresponding low-dimensional coordinate $\boldsymbol{Y}$, then BP network is trained by the normalized data. If the trained BP cannot reach the expected result, return to (2), otherwise, go to the next step.

(4) The new low-dimensional coordinate $\boldsymbol{y}_{n+1}$ of out-of-sample data can be calculated by the trained

Figure 2. Calculating procedure of BPLLE algorithm 3.2 Analysis of BPLLE algorithm

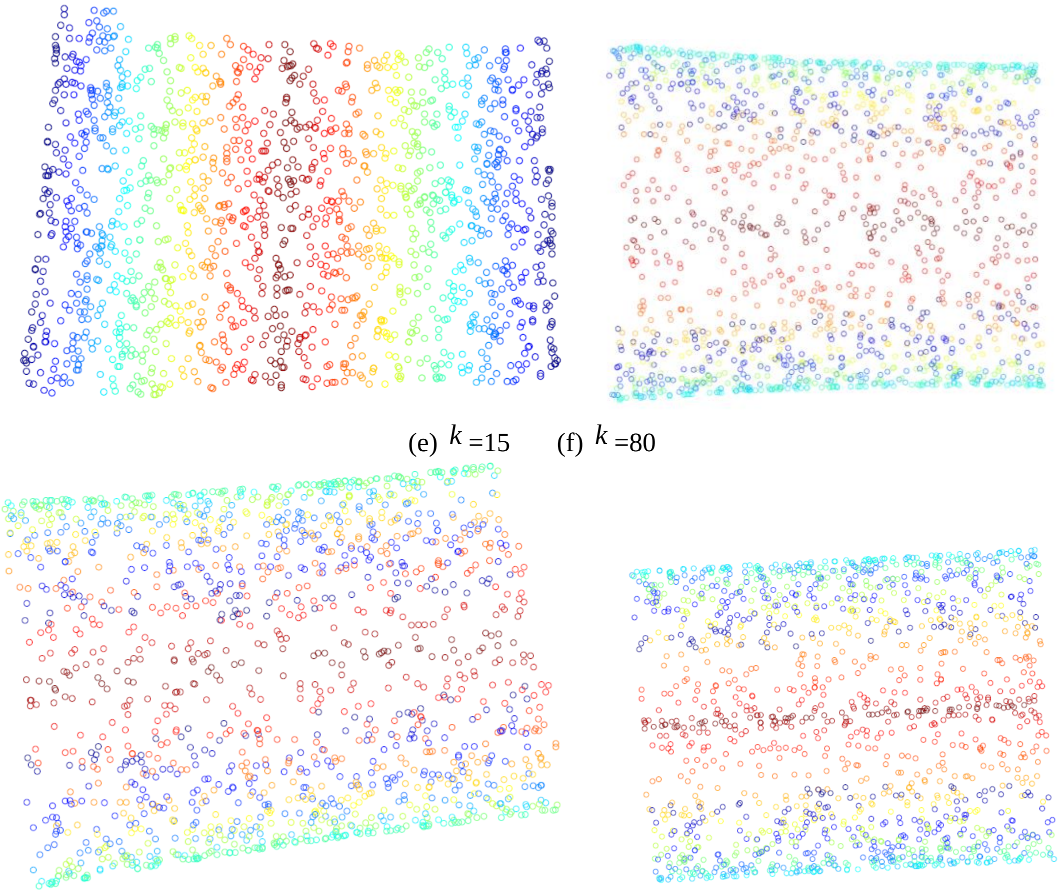
3.2. *Analysis of BPLLE algorithm*

3.2.1 *Analysis of LLE.* The final result of LLE is quite sensitive to the number of neighbors k and intrinsic dimensionality d, which have been investigated in host of literatures [24, 25]. If k is too small, the high-dimensional space will be divided into discrete subspace that the projection cannot express any global properties; if k is too large, samples in different subspace may be taken into the same one, in this way, the projection cannot show the intrinsic nonlinear structure of manifold. Furthermore, when k is large enough, LLE operates like PCA. All these facts are illustrated in Figure 3. The results are stable over a wide range of k, but do break down as k becomes too small or large.



(a) S curve (b) k =5

(c) k =6 (d) k =12



(e) k =15 (f) k =80

(g) k =100 (h) PCA

Figure 3: LLE operates S curve to 2 dimensions with different neighbors k

The optimal k can be estimated by the distribution of a dataset, but the distribution is difficult to be accurately obtained. A straightforward way for selecting k is to evaluate how well the high-dimensional property is preserved in the low one. Olga[26] suggested to calculate Spearman rank correlation coefficient ρ that assessed the correlation of rank order between two variables. ρ can be defined as below:

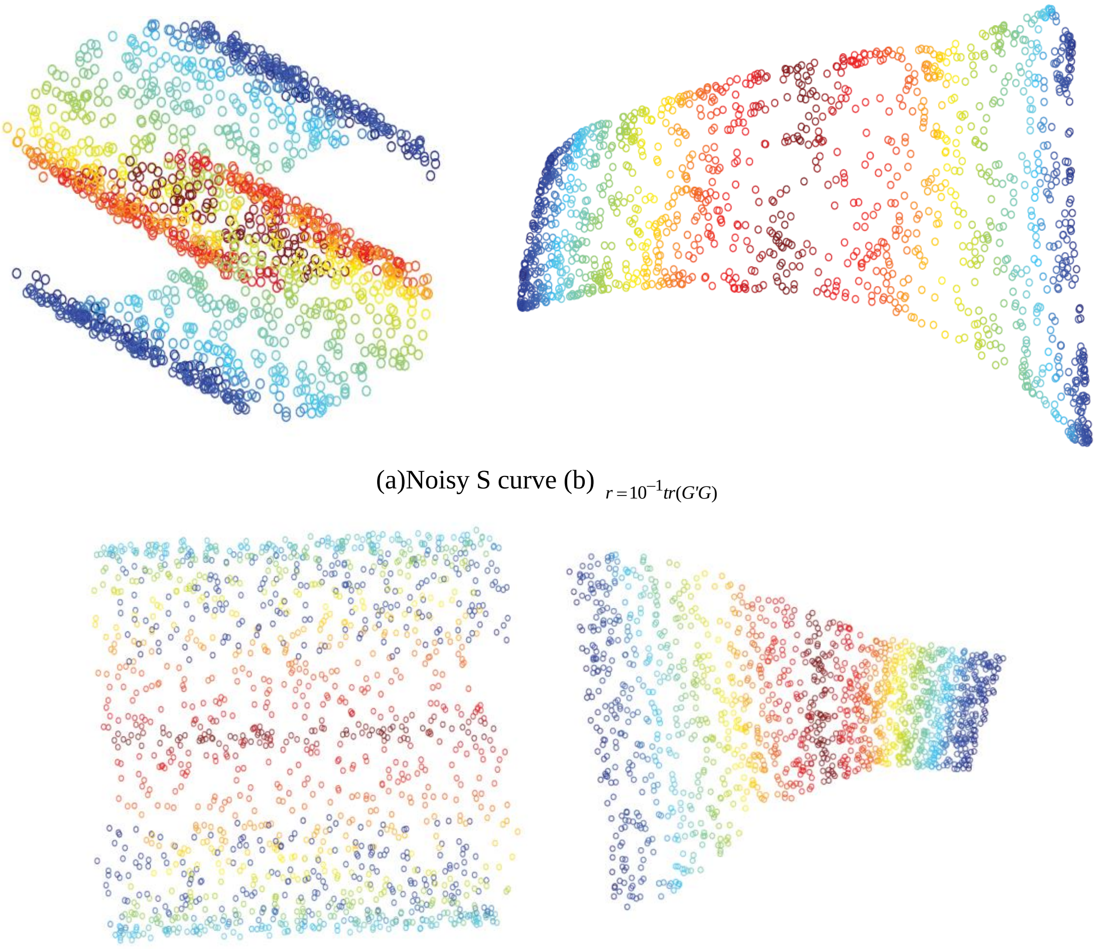
$$\rho = 1 - \frac{6\sum_{i=1}^{n}(d_x(i) - d_y(i))^2}{n(n^2 - 1)}$$

where $d_x(i)$ and $d_y(i)$ represent the ranks of pairwise distance in the high-dimensional space and the low-dimensional space, respectively. The larger value of ρ, the order of the two variables is more similar, and the best value is 1.

If the matrix $\boldsymbol{G}_i$ is singular, i.e., a small perturbation can have a very large effect on $\boldsymbol{W}$. It is advised that a regularization item $\boldsymbol{G}_i$ [22] should be added.

$$G_i = G_i + rI_k$$

where r is a regularization parameter, $I_k \in R^{k\times k}$ is an identity matrix. The value of r usually can be estimated by $r = \frac{\Delta}{k} tr(\boldsymbol{G}_i)$, where Δ is a small positive value[22], and $tr(\cdot)$ represents a matrix trace. We generate a dataset of S curve with signal-to-noise ratio (SNR) of 30db, and LLE with difference r is employed on the dataset. As it can be shown in Figure 4, the final result is sensitive to r .



(a)Noisy S curve (b) $r=10^{-1}tr(G'G)$

(c) $r=10^{-3}tr(G'G)$ (d) $r=10^{-6}tr(G'G)$

Figure 4: LLE operates on noisy S curve with different regularization item r

Intrinsic dimension d can be utilized to characterize the quantity of information. Liu et al.[27] showed that a small value of d may lose significant information, whereas a large one may remain too much redundant information. Correlation dimension algorithm is one of the most classical method to estimate d, and it can be computed by using the following equation[28]:

$$d = \lim_{\delta \to 0} \frac{logC(n,\delta)}{-log\delta}$$

where δ is a radius and $C(n,\delta)$ is correlation integral that is a statistical average. It can be obtained from:

$$C(n,\delta) = \frac{number\ \ of\ \ distances\ \ less\ \ than\ \ \delta}{sample\ \ size}$$

the computational cost of every step of LLE are listed in Table 1. As pointed out from Table 1, finding the neighbors and calculating the low-dimensional coordinate are computationally expensive in LLE. In practical application, LLE only provides the training dataset for BP, so the calculation speed of LLE does not impact the one of BPLLE.

Table 1: Each step computational cost of LLE

Step	Action performed	% of runtime
1	Find k neighbors	24
2	Calculate reconstruction weights matrix	17
3	Calculate low-dimensional coordinate	59

3.2.2 Analysis of BP. One of the most important parameter for BP is the network structure that control the complexity and accuracy of BP. The selection of BP network structure is to determine the number of hidden layers and neurons. A simple network structure has faster calculation speed, but it does not have enough capability to learn all patterns, which will reduce the performance of BP. A complex network structure can contribute to improving learning and predictive capability, however, it greatly increases the computational complexity. So far, there still is no effective method to decide the selection of BP network structure, and the most commonly used method is to repeat the experiment until a promising result is gotten. Moreover, the predictive capability of BP is related to the training data. If the training data contain all the classes of a dataset, BP can be trained well, then the predictive capability of BP will be improved. However, too many training data will lead to overfitting, which will decrease the performance of BP. In addition, the computational complexity and storage of BP rapidly increase, with the increasing dimensionality of dataset, and it may even cause the program to stop. Hence, appropriate network architecture, complete training data, and appropriate dimensionality of dataset are premises for BPLLE to work well. The computational process of BP can be divided into two steps: 1) train BP network; 2) predict new points. The computational cost of BP is mainly occupied by the first step, fortunately, similar to LLE, this step also belongs to an off-line mode. In the next section, we will proved that BPLLE has more faster speed than the ones of ILLE1 and ILLE2.

4. Experiments

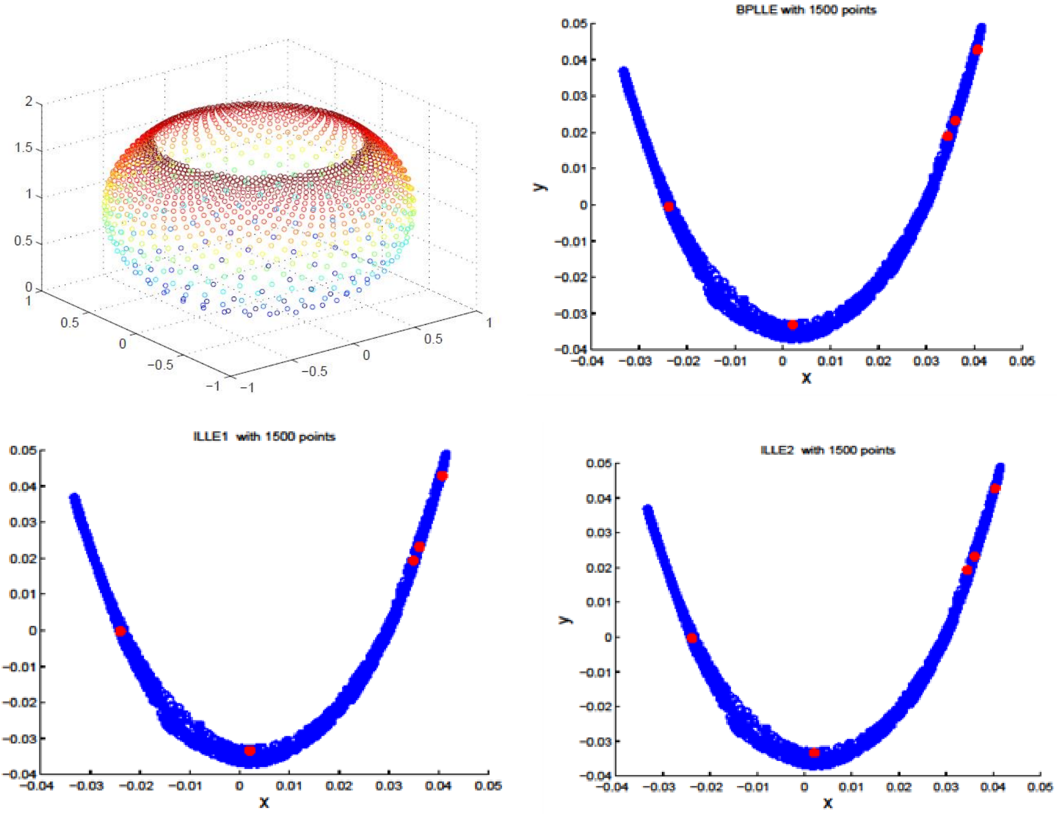
In this section, four synthetic datasets and two real datasets are utilized to verify BPLLE algorithm. All the datasets are summarized in Table 2. Swiss roll, Punctured Sphere and Toroidal Helix are three synthetic datasets that are the standard benchmarks for dimension reduction algorithms. The Random data is generated by random function in MATLAB as an approximation to a strong nonlinear dataset. For experimental study, each synthetic dataset is divided into two subsets: training and test data.

Table 2: Datasets operated on experiments

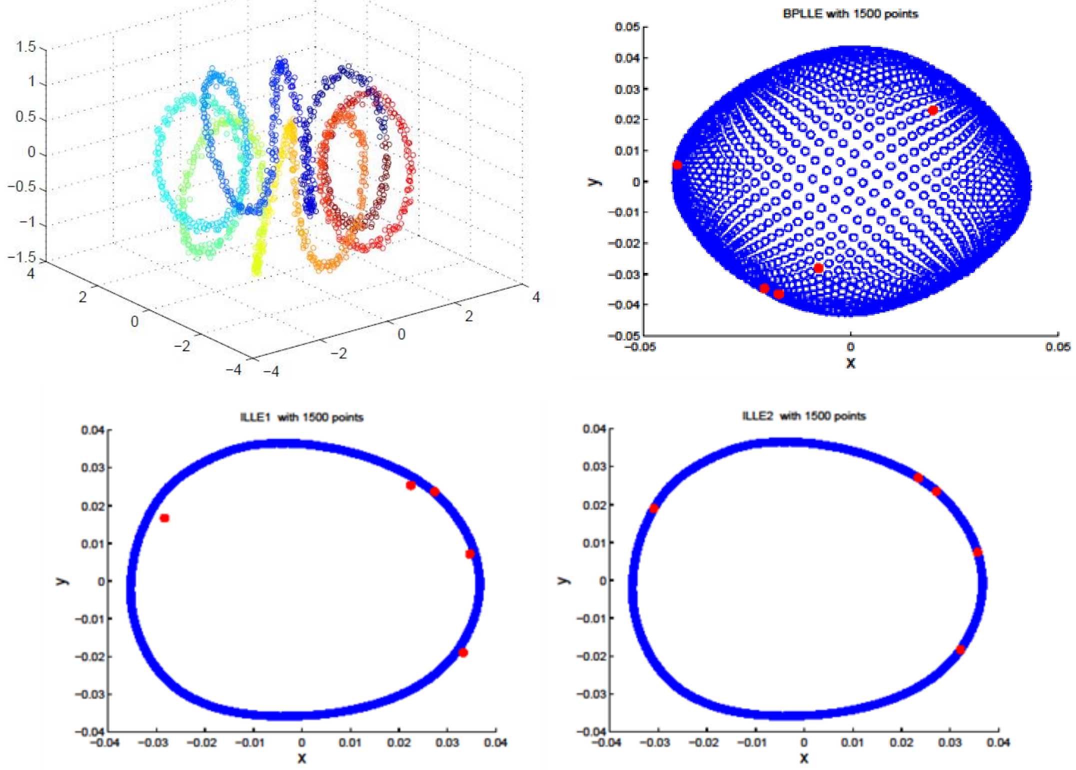
Dataset	N points	Dimensionality
Swiss roll	1500	3
Punctured Sphere	1500	3
Toroidal Helix	1500	3
Wine	178	13
Random data	1500	30
Iris data	150	4

4.1. Synthetic datasets

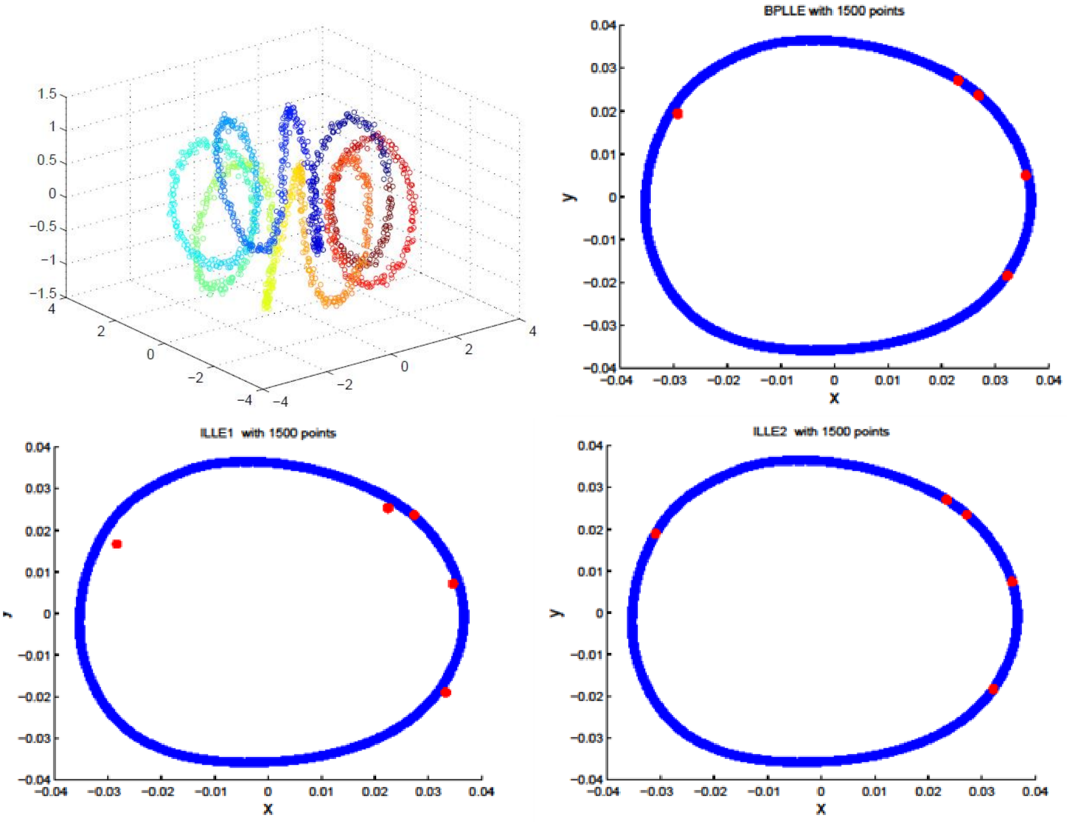
In order to prove that the proposed algorithm is feasible, a qualitatively analytic experiment is introduced. In each class dataset, 1495 points are collected as training data and the remaining 5 points as test data. First, the training dataset is mapped into embedding space; then the test data is tackled by three incremental methods: BPLLE, ILLE1, and ILLE2, respectively. The experimental results are shown in Figure 5. The blue 'o' points represent the low-dimensional coordinates of the training data, and the points filled with red color are the low-dimensional representations of the test data. It is obvious that all the test datasets processed by the three incremental LLE are visually almost the same, therefore, BPLLE is feasible to deal with out-of-sample data.



(a) BPLLE



(b) ILLE1



(c)ILLE2

Figure 5: Qualitative analysis of the incremental LLE

Furthermore, to indicate the advantage of the proposed method, we quantitatively analyse the three incremental methods by Spearman rank correlation that is a nonparametric method. We randomly collect 100 points from Swiss roll dataset as test data and 1400 samples as training data. Similar to the above experiment, the training dataset is mapped into a low-dimensional space, then the 100 test points are calculated during 20 iterations. The final results are indicated in Figure 6. As shown in Figure 6 (a), the red curve is always at the top of the green and black curves, i.e., the order of the pairwise distances is well preserved by BPLLE. Moreover, observing Figure 6(a), (b), and (c), it is quite clear that the three curves become more and more closer with the sample size increasing, i.e., the Spearman rank correlation values of ILLE1 and ILLE2 will gradually get close even exceed to the ones of BPLLE, when the sample size is large. This is because that the larger sample size, the data distribution is more compact and the local part of the dataset is more similar to linearity, which can improve the performance of ILLE1 and ILLE2. In addition, we estimate ρ for all the datasets, and the largest value of Spearman rank correlation value is recorded in each iteration. As shown in Table 3, BPLLE performs the best.

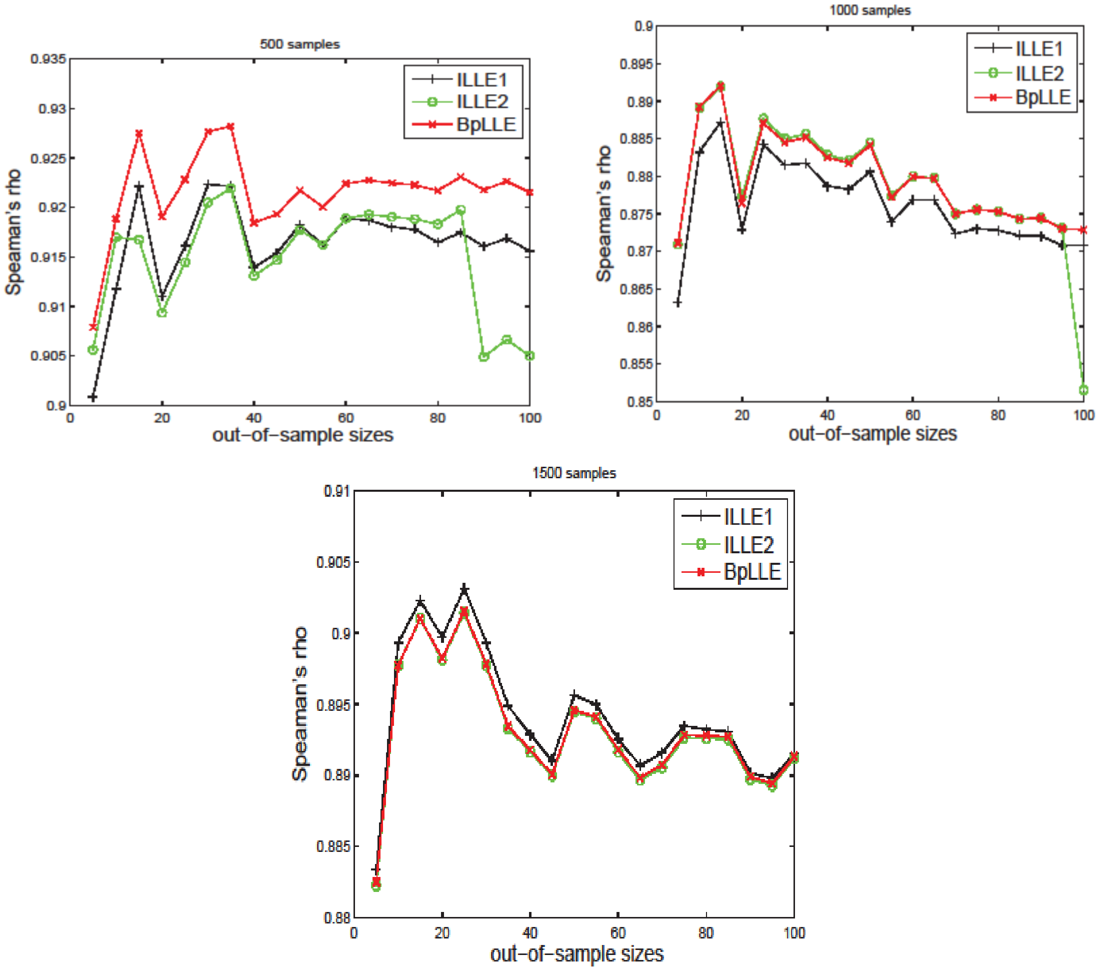


Figure 6: Spearman's rho measurement with different sample size

We redo the second experiment and record the computational cost of each algorithm. As shown in Figure 7, we can see that the green curve and the black curve are almost coincident, that is, ILLE1 and ILLE2 have the similar complexity. BPLLE is faster because it avoid looking for the neighbors of new data, and a trained well BP is efficient to deal with new data. Hnece, the complexity of BPLLE is the lowest.

Table 3: Spearman rank correlation value for the datasets

Dataset	Points/Per iteration	ILL1	ILL2	BPLLE
S curve	10	20	18	22
Wine	3	5	2	7
Random data	10	15	0	55

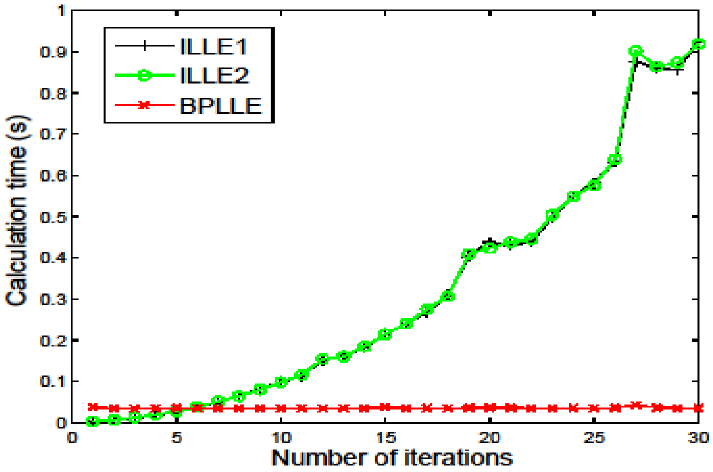


Figure 7: Time consuming calculations

4.2. Iris dataset datasets

Iris dataset presented in Fishers article is widely used in extensive literatures to verify recognition algorithms. Iris dataset can be divided into three categories, and each category contains 50 points. A three-dimensional plot obtained by arbitrarily selecting three dimensions of Iris dataset is shown in Figure 8(a), in which one category is linearly separable, but the other two categories are intertwined with each other, making it difficult for linearly separated in the original space. BPLLE, ILLE1, ILLE2 and LPP algorithms are utilized to Iris data with different number of training and test data, then SVM is employed to calculate the recognization accuracy. The final results are shown in Figure 8(b). Since LPP is a linear algorithm, its recognization accuracy is the lowest. The recognization accuracies of ILLE1and ILLE2 are almost the same, which can also demonstrate that both of the algorithms are equivalent. Although LLE is a nonlinear algorithm, ILLE1 and ILLE2 deal with out-of-sample data with linear mode, further, the final results are also depend on the selection of neighbors. As we all known, it is hard to accurately find the neighbors of a point in a high-dimensional space. However, a trained well BP handles out-of-sample data with nonlinear mode. Therefore, the recognization accuracies of ILLE1and ILLE2 are lower than the one of BPLLE.

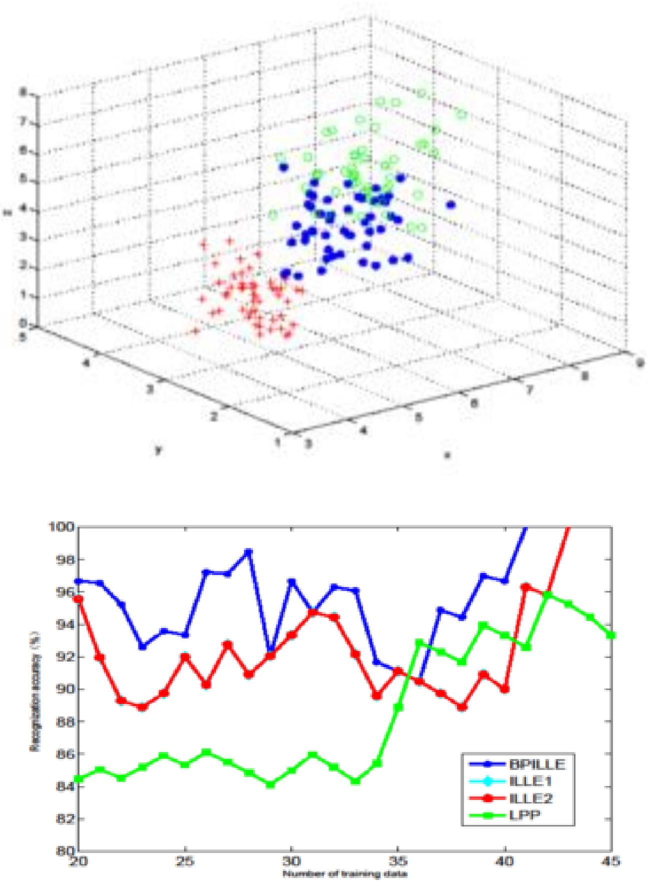


Figure 8: (a) Three dimension Iris data; (b) Recognition accuracy with different number of training data

5. Conclusion

In this paper, two classical incremental LLE algorithms are analyzed, and their drawbacks for out-of-sample data are pointed. In order to overcome these drawbacks, BPLLE is proposed. In BPLLE algorithm, training samples are mapped into low-dimensional space by LLE, and BP network is trained by the original data and their corresponding low-dimensional representations, finally out-of-sample data is processed by the trained BP. BPLLE can avoid searching neighbors of a new sample, by which the computational speed is greatly improved. In addition, BPLLE can learn knowledge from all samples, and has strong predictive capability, which is advantaged for handling nonlinear distribution dataset. Experimental results show that BPLLE is effective. However, there still are some open problems in BPLLE, such as how to choose the suitable BP neural network architecture, and BPLLE costs too much when the data dimensionality is too high. All these should be further studied.

Compliance with ethical standards

Conflict of interest: The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgements

Funding: This work is supported by the National Natural Science Foundation of China (No. 51774088).

References

[1] H. Dong, Z.Wang, S. X. Ding, and H. Gao, Finite-horizon estimation of randomly occurring faults for a class of nonlinear time-varying systems, Automatica, 2014, 50(2):3182-3189.

[2] Z. Zhou, N. Chawla, Y. Jin, and G. Williams, Big data opportunities and challenges: Discussions from data analytics perspectives, Computational Intelligence Magazine, IEEE, 2014, 9(4):62-74.

[3] Xu J.J., Gai D., Yan L.M. A NEW FAULT IDENTIFICATION AND DIAGNOSIS ON PUMP VALVES OF MEDICAL RECIPROCATING PUMPS. Basic & Clinical Pharmacology & Toxicology, 2016,118 (Suppl. 1), 38-38.

[4] S. Yin, X. Yang, and H. R. Karimi, Data-driven adaptive observer for fault diagnosis, Mathematical Problems in Engineering, 2012, 42(1):1-21.

[5] S. Yin, X. Li, H. Gao, and O. Kaynak, Data-based techniques focused on modern industry: an overview, IEEE Transactions on Industrial Electronics, 2014, 62(1):657-667.

[6] T. Khoat and N. Duy Khuong, An effective framework for supervised dimension reduction, Neurocomputing,2014, 139(0): 97- 407.

[7] Xu Jianjun, Wang Bao'e, Yan Limei, Li Zhanping. The Strategy of the Smart Home Energy Optimization Control of the Hybrid Energy Coordinated Control. Transactions of China Electrotechnical Society, 2017, 32(12) 214-223.

[8] Z. Su, B. Tang, L. Deng, and Z. Liu, Fault diagnosis method using supervised extended local tangent space alignment for dimension reduction,Measurement, 2015, 62(0):1-14.

[9] Z. Zhou and X. Liu, On multi-class cost-sensitive learning, Computational Intelligence, vol. 26, no. 3, pp. 232-257, 2010.

[10] J. Chen and Y. Liu, Locally linear embedding: a survey, Artificial Intelligence Review, 2011, 36(1):29-48.

[11] S. Yin, G.Wang, and X. Yang, Robust pls approach for kpi-related prediction and diagnosis against outliers and missing data, International Journal of Systems Science, 2014, 45(7):1375-1382.

[12] Xu, J., Huang, L., Yin, S. et al. All-fiber self-mixing interferometer for displacement measurement based on the quadrature demodulation technique. Opt Rev. 2018,25(1):40-45.

[13] X. Liu, T. Duygu, W. W. Michael, and S. Norbert, Locally linear embedding (lle) for mri based alzheimer's disease classification, NeuroImage, 2013, 83(0):148-157.

[14] Z. Su, B. Tang, J. Ma, and L. Deng, Fault diagnosis method based on incremental enhanced supervised locally linear embedding and adaptive nearest neighbor classifier, Measurement, 2014, 48(0):136-148.

[15] H. Li, H. Jiang, B. Roberto, X. Liao, L. Cheng, and F. Su, Incremental manifold learning by spectral embedding methods, Pattern Recognition Letters, 2011, 32(10):1447-1455.

[16] S. Schuon, M. Durkovic, K. Diepold, J. Scheuerle, and S. Markward, Truly incremental locally linear embedding, 1st International Workshop on Cognition for Technical Systems, 2008.

[17] B. Yoshua, P. Jean-Francois, and V. Pascal, Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, in In Advances in Neural Information Processing Systems. MIT Press, 2003, 177-184.

[18] O. Kouropteva, O. Okun, A. Hadid, and M. Soriano, Beyond locally linear embedding algorithm, Image Analysis, 2005, 3540(0):521-530.

[19] K. S. Lawrence and T. R. Sam, Think globally, _t locally: Unsupervised learning of nonlinear manifolds, Journal of Machine Learning Research, 2003, 4:119-155.

[20] S. Jrgen, Deep learning in neural networks: An overview, Neural Networks, 2015, 61,(0):85-117.

[21] O. Kouropteva and O. Okun, Selection of the optimal parameter value for the locally linear embedding algorithm, FSKD, 2002, 2:359-363.

[22] A. Elgammal and C.-S. Lee, Nonlinear manifold learning for dynamic shape and dynamic appearance, Computer Vision and Image Understanding, 2007, 106(1):31-46.

[23] P. Lopamudra and K. T. Sunil, Performance prediction of gravity concentrator by using artificial neural network-a case study, International Journal of Mining Science and Technology, 2014, 24(4):461-465.

[24] Y. Zhan, J. Yin, and X. Liu, Adaptive neighborhood selection based on local linearity for manifold learning, Pattern Recognition, 2011,48(4):576-583.

[25] Q. Jiang, P. Huang, and H. Li, Adaptive neighborhood selection method based on silhouette index for locally linear embedding. International Journal of Advancements in Computing Technology, 2013, 5(8):647-654.

[26] K. Olga and O. Oleg, Incremental locally linear embedding, Pattern Recognition, 2005, 38(10):1764-1767.

[27] Y. Liu, Z. Yu, M. Zeng, and S. Wang, Dimension estimation using weighted correlation dimension method, Discrete Dynamics in Nature and Society, 2015, 114(6):1-10.

[28] D. Mo and S. H. Huang, Fractal-based intrinsic dimension estimation and its application in dimensionality reduction, Knowledge and Data Engineering, IEEE Transactions on, 2012, 24(1):59-71.