Pattern Recognition Letters 30 (2009) 1457–1463

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Incremental Laplacian eigenmaps by preserving adjacent information between data points

Peng Jia [a], Junsong Yin [b], Xinsheng Huang [a], Dewen Hu [a,*]

[a] Department of Automatic Control, College of Mechatronics and Automation, National University of Defense Technology, Changsha, Hunan 410073, China
[b] Beijing Institute of Electronic System Engineering, 100141, China

ARTICLE INFO

Article history:
Received 13 May 2008
Received in revised form 22 June 2009
Available online 13 August 2009

Communicated by M.A. Girolami

Keywords:
Laplacian eigenmaps
Incremental learning
Locally linear construction
Nonlinear dimensionality reduction

ABSTRACT

Traditional nonlinear manifold learning methods have achieved great success in dimensionality reduction and feature extraction, most of which are batch modes. However, if new samples are observed, the batch methods need to be calculated repeatedly, which is computationally intensive, especially when the number or dimension of the input samples are large. This paper presents incremental learning algorithms for Laplacian eigenmaps, which computes the low-dimensional representation of data set by optimally preserving local neighborhood information in a certain sense. Sub-manifold analysis algorithm together with an alternative formulation of linear incremental method is proposed to learn the new samples incrementally. The locally linear reconstruction mechanism is introduced to update the existing samples' embedding results. The algorithms are easy to be implemented and the computation procedure is simple. Simulation results testify the efficiency and accuracy of the proposed algorithms.

1. Introduction

In pattern recognition and machine learning, dimensionality reduction aims to transform the high-dimensional data points in a low-dimensional space, while retaining most of the underlying structure in the data. It can be used to solve the curse of dimensionality (Bellman, 1961), and to accomplish data visualization. Besides some linear algorithms, such as principal component analysis (PCA) (Turk and Pentland, 1991) and linear discriminant analysis (LDA) (Belhumeur et al., 1997), manifold learning, which serves as a nonlinear method, has attracted more and more attention recently.

Several efficient manifold learning techniques have been proposed. Isometric feature mapping (ISOMAP) (Balasubramanian et al., 2002) estimates the geodesic distances on the manifold and uses them for projection. Locally linear embedding (LLE) (Roweis and Saul, 2000) projects data points to a low-dimensional space that preserves local geometric properties. Laplacian eigenmaps (LE) (Belkin and Niyogi, 2003) uses the weighted distance between two points as the loss function to get the dimension reduction results. Local tangent space alignment (LTSA) (Zhang and Zha, 2004) constructs a local tangent space for each point and obtains the global low-dimensional embedding results through affine transformation of the local tangent spaces. Yan et al. (2007) present a general formulation known as graph embedding to unify different dimensionality reduction algorithms within a common framework.

All of the above algorithms have been widely applied. However, serving as the batch methods, they require all training samples are given in advance. When samples are observed sequentially, batch method is computationally complex. This is because the batch method needs to be run repeatedly once new samples are observed. To overcome the problem, many researchers have been working on incremental learning algorithms. The problem of incremental learning can be stated as follows. Let $X = [x_1, x_2, \ldots, x_n]$ be a data set, where $x_i \in R^{d_1}$. Assume that the low-dimensional coordinate y_i of x_i for the first n training samples are given. When a new sample x_{n+1} is observed, incremental learning should figure out how to project x_{n+1} in the low-dimensional space and to update the existing samples' low-dimensional coordinates (Liu et al., 2006). Martin and Anil (2006) describe an incremental version of ISOMAP: the geodesic distances are updated, and an incremental eigen-decomposition problem is solved. Kouropteva et al. (2005) assume the eigenvalues of the cost matrix remain the same when a new data point arrives and the incremental learning problem of LLE is tackled by solving a $d_2 \times d_2$ minimization problem, where d_2 is the dimensionality of the low-dimensional embedding space. Bengio et al. (2004) cast MDS, ISOMAP, LLE and LE in a common framework, in which these algorithms are seen as learning eigenfunctions of a kernel, and try to generalize the dimensionality reduction results for the novel data points. An incremental manifold learning algorithm via TSA is proposed by Liu et al. (2006)

* Corresponding author. Fax: +86 731 84574992.
E-mail addresses: pengjia@nudt.edu.cn (P. Jia), jsyin@nudt.edu.cn (J. Yin), dwhu@nudt.edu.cn (D. Hu).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.08.005

The LTSA algorithm is modified, and an incremental eigen-decomposition problem with increasing matrix size is solved by subspace iteration with Ritz acceleration. The main ideas of the present incremental methods can be divided into two groups:

(i) Training-sample-independent. This kind of incremental algorithms can calculate the low-dimensional embedding of new samples from an existing class or a new class, such as incremental versions for subspace methods, e.g. PCA and LDA (Skočaj and Leonardis, 2002; Ye et al., 2005).
(ii) Training-sample-dependent. Some incremental algorithms, especially for the batch methods which try to preserve the local adjacent information within the data set, get the low-dimensional embedding of a new sample by using the adjacent information to the existing samples. Updating the adjacent information matrix is the indispensable step for incremental learning (Bengio et al., 2004; Kouropteva et al., 2005; Martin and Anil, 2006; Liu et al., 2006).

This paper presents incremental learning algorithm for LE, which is traditionally performed in a batch mode. An alternative formulation of linear incremental method and sub-manifold analysis algorithm are proposed to project the new sample. The locally linear reconstruction mechanism is introduced to add new adjacent information and revise the existing samples' low-dimensional embedding results. The algorithms are easy to be implemented and the computation procedure is simple. Simulation results testify the efficiency and accuracy of the proposed algorithms.

The rest of this paper is organized as follows. Section 2 reviews LE algorithm. Section 3 presents incremental LE. Section 4 shows some experimental results of the proposed algorithms. Section 5 provides some conclusions and discussion.

2. Laplacian eigenmaps

For convenience, we present the important notations used in this paper in Table 1.

$X = [x_1, x_2, \ldots, x_n]$ denotes the data set, where $x_i \in R^{d_1}$. LE builds a graph incorporating neighborhood information of the data set (Belkin and Niyogi, 2003). Using the notion of the Laplacian of the graph, the algorithm computes a low-dimensional representation of the data set by optimally preserving local neighborhood information in a certain sense, as shown in Fig. 1.

Construct a weighted graph with n nodes, one for each point, and a set of edges connecting neighboring points. The embedding map is now provided by computing the eigenvectors of the graph Laplacian. The algorithm procedure is formally stated below.

1. Constructing the adjacency graph. Let G denote a graph with n nodes. We put an edge between nodes i and j if x_i and x_j are close. There are two variations:
   (a) ε-neighborhoods. Node i and j are connected by an edge if $\|x_i - x_j\|^2 < \varepsilon, \varepsilon \in R$, where the norm is the usual Euclidean norm in R^{d_1}.

Table 1
Notations used in this paper.

Notations	Descriptions
X	Input data
y	The low-dimensional embedding
n	Number of training data points
d_1	Dimension of the training data
d_2	Dimension of the low-dimensional embedding space
W	The adjacent weights matrix
D	A diagonal weight matrix, $D_{ii} = \sum_j W_{ji}$
L	Laplacian matrix

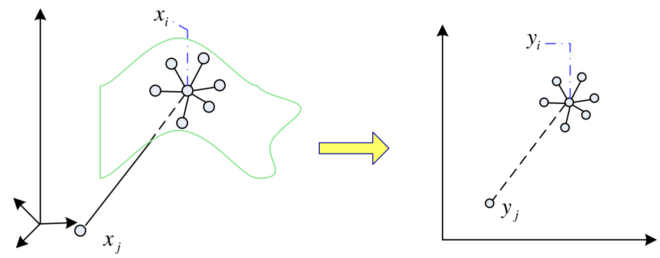

Fig. 1. A rationale sketch map of LE. When x_i and x_j are far from each other in the high-dimensional space, y_i and y_j are far from each other in the low-dimensional space as well.

   (b) k nearest neighbors. Node i and j are connected by an edge if i is among k nearest neighbors of j or j is among k nearest neighbors of i, $k \in N$.

2. Choosing the weights. Here are two variations for weighting the edges. W is a sparse symmetric $n \times n$ matrix with W_{ij} having the weight of the edge joining vertices i and j, and 0 if there is no such edge.
   (a) Heat kernel. If nodes i and j are connected, put

$$W_{ij} = \exp(-\|x_i - x_j\|^2/t), \quad t \in R \tag{1}$$

   The heat kernel reflects exactly the distance information between nodes i and j.
   (b) Simple-minded. $W_{ij} = 1$ if and only if vertices i and j are connected by an edge. This kind of weights only describe the simple connection information between nodes i and j.

3. Construct the object function. Consider the problem of mapping the weighted graph G to a low-dimensional space so that connected points stay as close together as possible. Let $y = (y_1, y_2, \ldots, y_n)$ be such a map. A reasonable criterion for choosing a good map is to minimize the following objective function:

$$\sum_{i,j} (y_i - y_j)^2 W_{ij} \tag{2}$$

There is a heavy penalty if neighboring points x_i and x_j are mapped far apart. Therefore, minimizing the objective function is an attempt to ensure that if x_i and x_j are close, then y_i and y_j are close as well.Let D be a diagonal weight matrix, whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$. Then the Laplacian matrix is $L = D - W$. It turns out that for any y, we have

$$\frac{1}{2} \sum_{i,j} (y_i - y_j)^2 W_{ij} = \mathrm{tr}(y^T L y) \tag{3}$$

The minimization problem can be reduced to find

$$\begin{aligned} &\arg\min_y \quad \mathrm{tr}(y^T L y) \\ &\text{s.t.} \quad y^T D y = 1 \\ &\qquad\; y^T D \mathbf{1} = 0 \end{aligned} \tag{4}$$

Since the bigger the D_{ii} is, the more important y_i is, there is a constraint as $y^T D y = 1$. Constraint $y^T D \mathbf{1} = 0$ is to eliminate the trivial solution $\mathbf{1}$ of problem (4).

4. Eigenmaps. Compute eigenvalues and eigenvectors for the generalized eigenvector problem,

$$Ly = \lambda D y \tag{5}$$

Let the column vectors $y_0, \ldots, y_{d_2-1}$ be the solutions of Eq. (4), ordered according to their eigenvalues, $0 = \lambda_0 \leqslant \lambda_1 \leqslant \ldots \leqslant \lambda_{d_2-1}$. We leave out the eigenvector y_0 corresponding to eigenvalue 0 and use the next d_2 eigenvectors for embedding in d_2-dimensional Euclidean space:

$$x_i \to (y_1(i), \ldots, y_{d_2}(i))$$

3. Incremental learning algorithms for Laplacian eigenmaps

When a new sample x_{n+1} is observed, it may change the local manifold distribution and the current training samples' neighboring information. Take the k nearest neighbors as an example, x_{n+1} may replace one of x_i's neighbors and change the local neighboring information of x_i, which can be seen in Fig. 2.

Therefore, in incremental learning, there are two kinds of samples that need to be considered: the novel sample x_{n+1}, the samples $x_{N(1)}, \ldots, x_{N(m)}$, whose neighbors have changed because of the insertion of x_{n+1}.

In this paper, we only consider the k nearest neighbors case. The ε-neighborhoods case can be deduced similarly. The incremental learning procedure includes the following steps.

3.1. The updating of adjacent matrix W

When x_{n+1} is observed, the adjacent matrix W needs to be updated first. There are two parts for the updating of W:

Part I: Construct the connection weights between x_{n+1} and $\{x_i\}_{i=1}^{n}$ if x_i is among k nearest neighbors of x_{n+1}. W is then extended to the scale of $(n+1) \times (n+1)$.

Part II: Reconstruct the connection weights of the sample points $x_{N(1)}, \ldots, x_{N(m)}$, whose neighbors have changed because of the insertion of x_{n+1}. Obviously, the low-dimensional embedding results $y_{N(1)}, \ldots, y_{N(m)}$ of $x_{N(1)}, \ldots, x_{N(m)}$ need to be updated.

3.2. Projection of x_{n+1}

We present an alternative formulation of linear incremental method and sub-manifold analysis algorithm to project x_{n+1} in the low-dimensional embedding space.

3.2.1. An alternative formulation of linear incremental method

Vector y_{n+1} should minimize the following target function:

$$\sum_{i=1}^{n} \|y_{n+1} - y_i\|^2 W_{(n+1)i} = \sum_i (y_{n+1} - y_i)^T (y_{n+1} - y_i) W_{(n+1)i} \tag{6}$$

Differentiate (6) by dy_{n+1}, and let the first order derivative be the equal of 0, we get:

$$-2\sum_i (y_{n+1} - y_i) W_{(n+1)i} = 0 \tag{7}$$

$$\text{i.e.} \quad (y_{n+1} - y_1, \ldots, y_{n+1} - y_n) \begin{pmatrix} W_{(n+1)1} \\ \vdots \\ W_{(n+1)n} \end{pmatrix} = 0 \tag{8}$$

Finally, we have:

$$y_{n+1} = \left[(y_1, \ldots, y_n) \begin{pmatrix} W_{(n+1)1} \\ \vdots \\ W_{(n+1)n} \end{pmatrix} \right] \Bigg/ \sum_{i=1}^{n} W_{(n+1)i} \tag{9}$$

3.2.2. Sub-manifold analysis method

Detect the k nearest neighborhoods of x_{n+1}. Let $X_S = [x_{S(1)}, \ldots, x_{S(k)}, x_{n+1}]$ denote this set of points including x_{n+1}, as shown in Fig. 3. Points $x_{S(1)}, \ldots, x_{S(k)}, x_{n+1}$ can be seen as a sub-manifold, and the incremental learning procedure is:

(1) Using LE on the sub-manifold. Construct the sub-adjacent matrix W_S by using the heat kernel:

$$W_S(i,j) = \begin{cases} \exp(-\|x_i - x_j\|^2/t), & t \in R, x_i \text{ is within } x_j\text{'s} \\ k \text{ nearest neighborhoods or } x_j \text{ is within } x_i\text{'s } k \\ \text{nearest neighborhoods on the sub-manifold} \\ 0, & \text{otherwise} \end{cases}$$

$$x_i, x_j \in X_S \tag{10}$$

Obviously, each point in X_S connects with all the other points. Calculate the $(k+1)$ by $(k+1)$ matrix D_S and sub-Laplacian matrix L_S:

$$D_S(i,i) = \sum_j W_S(j,i), \quad L_S = D_S - W_S$$

Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$L_S v = \lambda_S D_S v \tag{11}$$

Let the column vectors $v_0, \ldots, v_{d_2}$ be the solutions of Eq. (12), ordered according to their eigenvalues, $0 = \lambda_S^0 \leqslant \lambda_S^1 \leqslant \ldots \leqslant \lambda_S^{d_2}$. Leave out the eigenvector v_0 corresponding to eigenvalue 0 and use the next d_2 eigenvectors for embedding in d_2-dimensional Euclidean space, and then we get the low-dimensional coordinates for $x_{S(1)}, \ldots, x_{S(k)}, x_{n+1}$ on the sub-manifold:

$$x_i \to (v_1(i), \ldots, v_{d_2}(i)), \quad x_i \in X_S$$

(2) Calculating the global coordinate y_{n+1} for x_{n+1}. From v_{n+1} to y_{n+1}, the calculating procedure can be seen as a coordinate transformation problem. During the transformation, we preserve the relationships between x_{n+1} and its k nearest

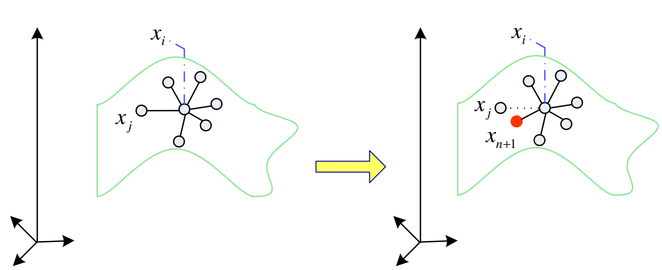

Fig. 2. x_{n+1} replaces x_j and becomes one of x_i's k nearest neighbors.

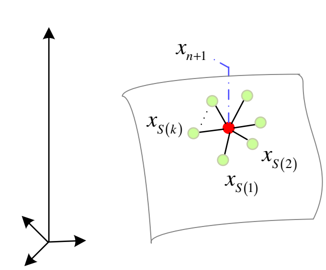

Fig. 3. A novel sample x_{n+1} and its k nearest neighborhoods $x_{S(1)}, \ldots x_{S(k)}$.

Table 2
The computing procedure of the incremental learning algorithm.

Input: x_{n+1}
Output: W^{new}, y_{n+1}, $y_{N(i)}^{new}$
Step 1: update the adjacent matrix and get W^{new}
Step 2: project x_{n+1} in the low-dimensional space and get y_{n+1} by using the differential method or the sub-manifold analysis method
Step 3: if there exist samples whose neighbors have changed because of the insertion of x_{n+1}, update $y_{N(i)}$ and get $y_{N(i)}^{new}$

neighborhoods $x_{S(1)}, \ldots, x_{S(k)}$. By using LE on the sub-manifold, the algorithm has detected the intrinsic structure information between x_{n+1} and $x_{S(1)}, \ldots, x_{S(k)}$. So compute the constrained weights matrix $C = [c_1, \ldots, c_k] \in R^k$ between x_{n+1} and $x_{S(1)}, \ldots, x_{S(k)}$, with the reconstruction error minimized:

$$\begin{aligned} &\min_{c_i} \left| v_{k+1} - \sum_{i=1}^{k} c_i v_i \right|^2 \\ &\text{s.t.} \quad \sum_i c_i = 1 \end{aligned} \tag{12}$$

Thus, the optimal weights c_i subject to the constraint can be calculated by solving a least squares problem. Then the global coordinate for x_{n+1} is computed with the weight vector C.

$$x_{n+1} \rightarrow y_{n+1} = \sum_{i=1}^{k} c_i y_{S(i)}$$

where $y_{S(1)}, \ldots, y_{S(k)}$ are the low-dimensional coordinates of $x_{S(1)}, \ldots, x_{S(k)}$.

3.3. Updating of the existing samples' embedding results

If there exist samples $x_{N(1)}, \ldots, x_{N(m)}$ whose neighbors have changed because of the insertion of x_{n+1}, besides calculating y_{n+1}, the embedding coordinates $y_{N(1)}, \ldots, y_{N(m)}$ need to be updated. We introduce the locally linear reconstruction mechanism to add novel adjacent information and revise the existing samples' low-dimensional results. The updating involves two stages:

(1) Locally fitting hyperplanes around $x_{N(i)}$, $i = 1, \ldots, m$, based on its k nearest neighbors $x_{N(i)}^1, \ldots, x_{N(i)}^k$, and calculating reconstruction weights. The cost function minimized is:

$$\varepsilon_i(w^{(i)}) = \left| x_{N(i)} - \sum_{j=1}^{k} w_j^{(i)} x_{N(i)}^j \right|^2 \tag{13}$$

Weights $w_j^{(i)}$ sum up to 1.

(2) Finding low-dimensional coordinates $y_{N(i)}$ for each $x_{N(i)}$ based on these weights. The weight $w^{(i)}$ is fixed and new $y_{N(i)}$ is obtained by:

$$y_{N(i)} = \sum_{j=1}^{k} w_j^{(i)} y_{N(i)}^j \tag{14}$$

It is noticeable that a new sample may not get the accurate low-dimensional embedding coordinates at once. But along with the incremental learning and updating process, the low-dimensional coordinates will be regulated. In the end, the unfolding manifold can uncover the intrinsic structure of the original manifold efficiently.

The computing procedure of the proposed incremental algorithm is shown in Table 2.

4. Simulations

4.1. Results on dimensionality reduction

This section presents the dimensionality reduction results of incremental LE on some nonlinear manifolds: Swiss roll and S-curve data. The weights matrix W is defined as

$$W_{ij} = \begin{cases} \exp(-\|x_i - x_j\|^2/t), \quad t \in R, \text{ node } i \text{ is among } k \\ \text{nearest neighbors of node } j \text{ or node } j \text{ is among } k \\ \text{nearest neighbors of node } i \\ 0, \quad \text{others} \end{cases} \quad i,j = 1,2,\ldots,n$$

The number of the nearest neighbors, k, is set to eight. One thousand and five hundred points are sampled randomly on the manifolds, of which the first 200 points serve as the training samples, and the rest are input in sequence to testify the efficiency of incremental LE.

In Figs. 4 and 5, the simulation results on Swiss Roll and S-curve data set are given. The 3D datasets are mapped to 2D by the incremental learning algorithm proposed in this paper. From the above two figures, several points are worthy of note.

(1) When the manifold is well sampled, namely the training samples can cover the whole input space, both the differential and sub-manifold analysis methods can project the novel samples in the low-dimensional embedding space accurately. This is because the whole manifold distribution is known, and the adjacent information of a novel sample is detected easily and properly. The embedding results can be computed reliably.

(2) The samples may not get the accurate low-dimensional embedding coordinates at once, e.g. the point which is marked with the dotted line in Fig. 5 (e). However, along with the incremental learning and updating process, the low-dimensional embedding results will be revised and get more accurate approximation to the theoretical solution. In Fig. 5 (f) – (h), the marked point is projected more accurately.

4.2. Applied to pattern classification of face images

4.2.1. The Frey Face dataset

In this experiment, the Frey Face dataset[1] has been chosen. It contains 1965 images of one individual with different poses and expressions. Each image is a gray picture of 20×28 pixels, and some typical images are shown in Fig. 6. six hundred images serve as the training samples and another 400 images are learned incrementally by the sub-manifold analysis algorithm. Fig. 7 shows some of the test images.

The unfolding results of the 600 training samples by using batch LE are shown in Fig. 8a. The horizontal axis represents the face is panning from left to right, and the vertical axis denotes that the

[1] Available at http://www.cs.toronto.edu/~roweis/data.html.

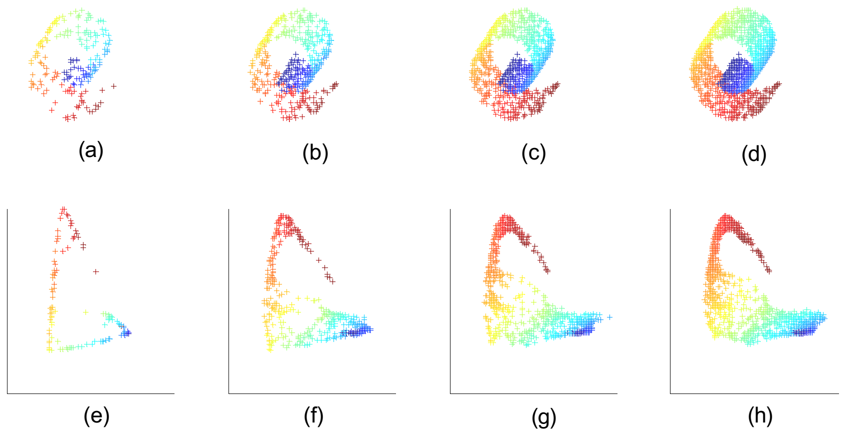

Fig. 4. Incremental dimensionality reduction results of Swiss Roll data set by using the differential method. (a) The initial 200 training samples; (b) 500 samples; (c) 1000 samples; (d) 1500 samples; (e) the unfolding results of the initial training samples by batch LE; (f) incremental learning results of the 500 samples; (g) incremental learning results of the 1000 samples; (h) incremental learning results of the 1500 samples.

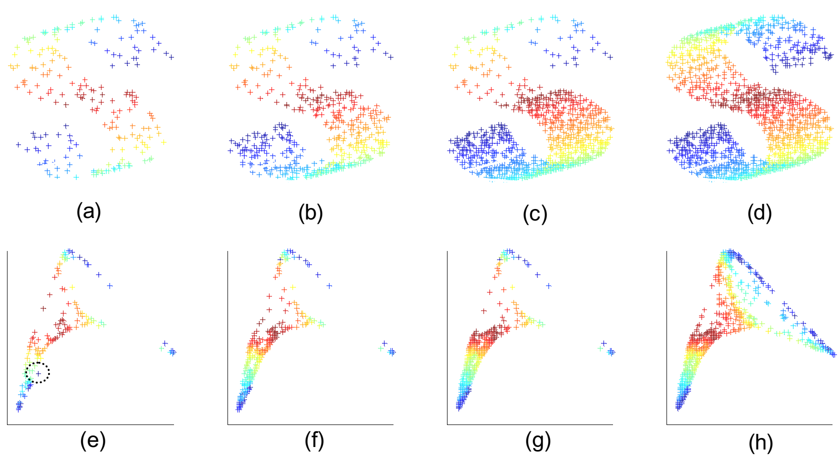

Fig. 5. Incremental dimensionality reduction results of S-curve data set by using the sub-manifold analysis method. (a) The initial 200 training samples; (b) 500 samples; (c) 1000 samples; (d) 1500 samples; (e) the unfolding results of the initial training samples by batch LE; (f) incremental learning results of the 500 samples; (g) incremental learning results of the 1000 samples; (h) incremental learning results of the 1500 samples.

Fig. 6. Typical images in Frey Face dataset.

Fig. 7. Some test images.

expression is changing from happy to angry. LE can detect the intrinsic structure of the input data set. Images with similar expressions and poses cluster together. Fig. 8b shows the incremental learning results by sub-manifold analysis algorithm and the embedding points of the test images shown in Fig. 7 are marked with red circles. Obviously, the test images are projected to the low-dimensional space accurately according to their poses and expressions. Although images nos. 1 to 5 in Fig. 7 show new expressions which are different from the training samples, the adjacent information of them is well detected and preserved by the incremental learning algorithm.

4.2.2. The ORL face dataset

This section will apply LE and the proposed algorithms to ORL face dataset.[2] The dataset contains 400 face images of 40 individuals. Each individual has 10 gray images with size 92×112. Some samples are shown in Fig. 9. In this scheme, one image from each subject is randomly selected for testing, while the rest images for training. The experiment is repeated 10 times so that every image of each subject can serve as the testing sample once. And the average recognition rates of various low-dimensional spaces are shown in Fig. 10. The testing images are learned incrementally by the proposed algorithms.

We use the nearest neighbors algorithm as the classifier. The number of the nearest neighbors in LE, k, is set to 4, and the weights matrix W is constructed in the simple-minded way. From the results we observe that the submanifold incremental learning

[2] Available at http://www.cs.toronto.edu/~roweis/data.html.

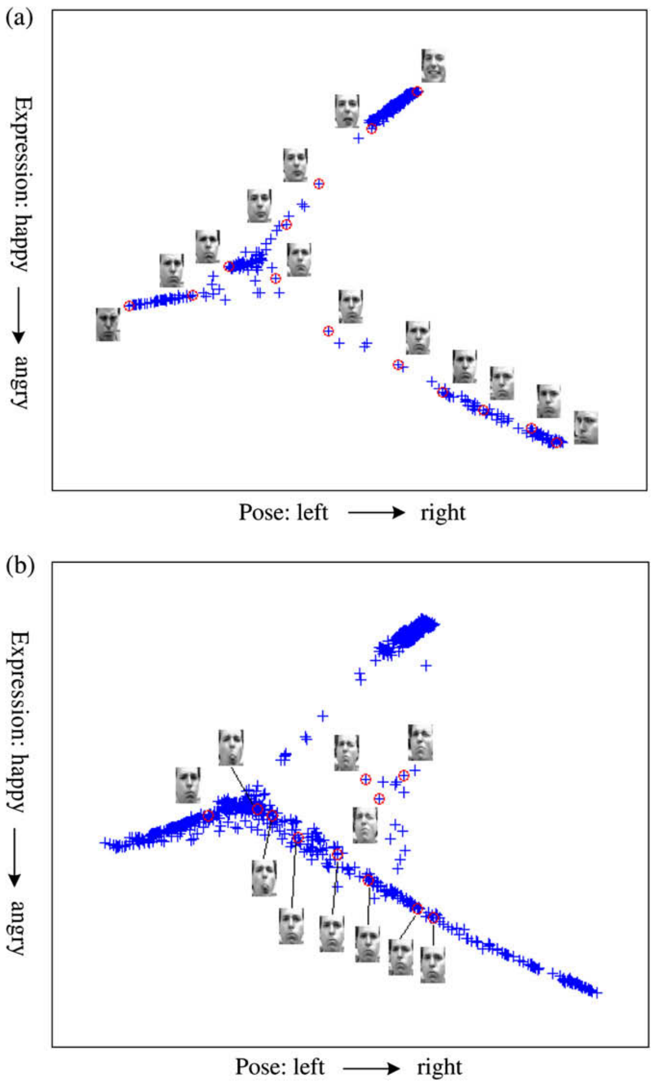

Fig. 8. (a) The unfolding results of the training samples by batch LE; (b) the incremental learning results by using sub-manifold analysis algorithm. The embedding points of the test images shown in Fig. 7 are marked with red circles. The horizontal axis represents the face is panning from left to right, and the vertical axis denotes that the expression is changing from happy to angry. The test images are projected to the low-dimensional space accurately according to their poses and expressions by the proposed algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 9. Face samples in ORL data set.

algorithms well detect the intrinsic structure information of the input manifold, and achieve the comparable classification accuracy to batch LE. We know that batch LE uses all the samples' adjacent information to extract features, while incremental LE uses only part of samples' adjacent information to get probable distribution of the original manifold. But when new samples are observed, the incremental learning and updating algorithm proposed in this paper will add new distance information and revise the existing unfolding results. Finally incremental LE achieves comparable or even better dimensionality reduction accuracy than batch LE.

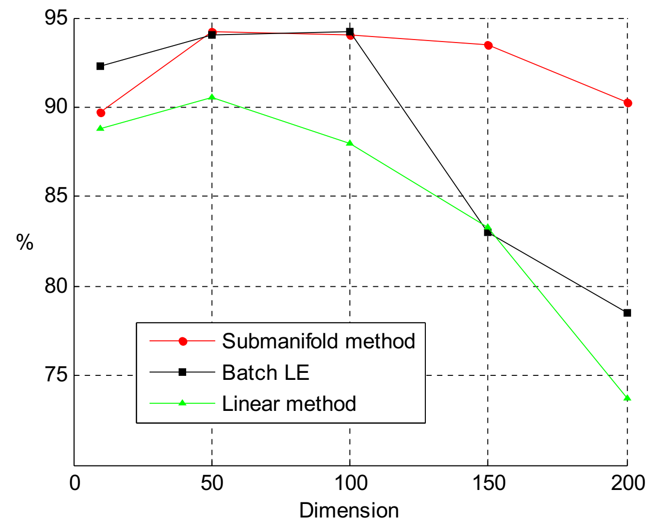

Fig. 10. Accuracy rate (%) for batch LE and incremental LE algorithms using different number of dimensions.

Fig. 11. Examples of binary digits 0, 2, 4, 6 and 7.

4.2.3. Binary Alphadigits

Binary Alphadigits dataset[2] contains binary digits of 0 through 9 and capital A through Z with size 20×16. There are 39 examples of each class. We use digits 0, 2, 4, 6 and 7 to testify the efficiency of the updating process of the existing samples. Some binary examples are shown in Fig. 11. Thirty samples of each class chosen randomly serve as the training set and the rest 9 samples as the testing set. The training samples are projected to different low-dimensional spaces by batch LE. The testing ones are learned by sub-manifold analysis algorithm. Using nearest neighbors classifier, when the updating of the existing samples is present and absent the recognition rates are shown in Table 3.

Obviously, when the updating of the existing samples is present, the recognition rate is boosted. The updating process makes the whole incremental algorithm have characteristics of an iterative method and the embedding results are more accurate.

4.3. Computational complexity analysis

Except updating the adjacent matrix W, the core computation process of the differential method involves some easy additions

Table 3
Comparison of recognition rates. The training samples are projected to different low-dimensional spaces by batch LE and the testing ones are learned by sub-manifold analysis algorithm. When the updating of the existing samples is present, the recognition rate is boosted.

Dimension	2	5	6	9	10	11	15	20	25
Incremental learning									
No updating	80	84.44	**91.11**	91.11	88.89	88.89	86.67	88.89	86.67
Updating the existing samples	**86.67**	**91.11**	**91.11**	91.11	91.11	**91.11**	**93.33**	**93.33**	**93.33**
Batch LE	82.22	86.67	88.89	**93.33**	**93.33**	**91.11**	88.89	88.89	86.67

and multiplications. The computation is simple. The sub-manifold method needs to solve a $(k+1)\times(k+1)$ eigenvector problem and a simple least squares problem, where k is the number of the nearest neighbors. The time complexity of solving the $(k+1)\times(k+1)$ eigenvector problem is $O((k+1)^3)$ (Golub and Van Loan, 1996). We use the 'fmincon' function in MATLAB software to solve the least squares problem, and the computation time is short enough for on-line learning.

5. Conclusion

This paper presents the incremental learning algorithm for LE. The sub-manifold analysis method and an alternative formulation of linear incremental method are proposed to project the new sample in the low-dimensional space. The locally linear reconstruction mechanism is introduced to add novel adjacent information and update the current samples' embedding results. Actually, the updating mechanism is similar to an iterative method. Every time a novel sample is observed, the current samples' embedding results are locally improved since only the samples whose neighbors change are updated. The computation procedure is simple, and the proposed incremental algorithm is easy to be generalized to other nonlinear manifold learning methods, such as LLE and ISOMAP. It is noticeable that the sub-manifold's properties are affected by the number of the nearest neighbors, k. The selection of k should satisfy that enough position information of the new sample can be obtained. Some efficient methods have been presented to select the value of k properly and adaptively (Wang et al., 2005; Lin and Zha, 2008).

Different from the iterative methods, the computation of incremental LE is simple, and it is able to accomplish on-line learning when samples are observed one by one. But the algorithm needs to be trained first. One of the directions of the future work is to improve the algorithm and apply it to real face recognition systems.

Acknowledgement

The authors would like to thank the action editor and the anonymous reviewers for their constructive comments. This work was supported in part by Natural Science Foundation of China (60675005, 60835005, 60736018), National Basic Research Program of China (2007CB311001), Program for New Century Excellent Talents in University (NCET-08-0147) and China Postdoctoral Science Foundation.

References

Balasubramanian, M., Schwartz, E.L., Tenenbaum, J.B., de Silva, V., Langford, J.C., 2002. The ISOMAP algorithm and topological stability. Science 295, 7.
Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J., 1997. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Machine Intell. 19 (7), 711–720.
Belkin, M., Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. 15 (6), 1373–1396.
Bellman, R., 1961. Adaptive Control Processes: A Guided Tour. Princeton University Press.
Bengio, Y., Paiement, J.-F., Vincent, P., 2004. Out-of-sample extensions for LLE, ISOMAP, MDS, eigenmaps and spectral clustering. Advances in Neural Information Processing Systems, vol. 16. MIT Press.
Golub, G.H., Van Loan, C.F., 1996. Matrix Computations, third ed. The Johns Hopkins Univ. Press, Baltimore, MD.
Kouropteva, O., Okun, O., Pietikäinen, M., 2005. Incremental locally linear embedding. Pattern Recognit. 38, 1764–1767.
Lin, T., Zha, H., 2008. Riemannian manifold learning. IEEE Trans. Pattern Anal. Machine Intell. 30 (5), 796–809.
Liu, X., Yin, J., Feng, Z., Dong, J., 2006. Incremental manifold learning via tangent space alignment. ANNPR 2006, LNAI 4087, pp. 107–121.
Martin, H.C.L., Anil, K.J., 2006. Incremental nonlinear dimensionality reduction by manifold learning. IEEE Trans. Pattern Anal. Machine Intell. 28 (3), 377–391.
Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326.
Skočaj, D., Leonardis, A., 2002. Weighted incremental subspace learning. In: Proceedings of Workshop on Cognitive Vision, Zurich.
Turk, M., Pentland, A.P., 1991. Face recognition using eigenfaces. In: IEEE Conf. on Computer Vision and Pattern Recognition.
Wang, J., Zhang, Z., Zha, H., 2005. Adaptive manifold learning. Proc. Adv. Neural Inform. Process. Systems 17, 1473–1480.
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 Trans. Pattern Anal. Machine Intell. 29, 40–51.
Ye, J., Li, Q., Xiong, H., Park, H., Janardan, R., Kumar, V., 2005. IDR/QR: An incremental dimension reduction algorithm via QR decomposition. IEEE Trans. Knowl. Data Eng. 17, 1208–1222.
Zhang, Z.Y., Zha, H.Y., 2004. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM J. Sci. Comput. 26 (1), 313–338.