Incremental Manifold Learning Via Tangent Space Alignment

Xiaoming Liu[1], Jianwei Yin[1], Zhilin Feng[1,2], and Jinxiang Dong[1]

[1] Department of Computer Science and Technology, Zhejiang University, China
[2] Zhijiang College, Zhejiang University of Technology, Hangzhou 310024, China
liuxiaoming@zju.edu.cn, zjuyjw@zju.edu.cn

Abstract. Several algorithms have been proposed to analysis the structure of high-dimensional data based on the notion of manifold learning. They have been used to extract the intrinsic characteristic of different type of high-dimensional data by performing nonlinear dimensionality reduction. Most of them operate in a "batch" mode and cannot be efficiently applied when data are collected sequentially. In this paper, we proposed an incremental version (ILTSA) of LTSA (Local Tangent Space Alignment), which is one of the key manifold learning algorithms. Besides, a landmark version of LTSA (LLTSA) is proposed, where landmarks are selected based on LASSO regression, which is well known to favor sparse approximations because it uses regularization with l_1 norm. Furthermore, an incremental version (ILLTSA) of LLTSA is also proposed. Experimental results on synthetic data and real word data sets demonstrate the effectivity of our algorithms.

Keywords: manifold learning, LTSA, incremental learning, LASSO.

1 Introduction

The purpose of dimensionality reduction is to transform a high-dimensional data set into a low-dimensional space, while retaining most of the underlying structure in the data. Dimensionality reduction has long been an important problem in the field of pattern classification, data mining and machine learning. It is important for several reasons, with the most important being to circumvent the curse of dimensionality: many classifiers perform poorly in a high-dimensional space given a small number of training samples. Dimensionality reduction can also be used to visualize the data by transforming the data into two or three dimensions.

Many dimension reduction algorithms have been proposed, and can be classified into two classes roughly: linear methods and nonlinear methods. PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) are the most popular linear dimensionality reduction methods. While they are easy understandable, simple to implemented and can catch the linear structures of data, they can not discover the nonlinear structures of the data. In reality, many high dimension data is embedded in a

F. Schwenker and S. Marinai (Eds.): ANNPR 2006, LNAI 4087, pp. 107 – 121, 2006.

low nonlinear manifold, and there are some cues that the low-dimensional embedding is consistent with human perception [1]. To address the shortcomings of the linear methods, kernel PCA method and kernel LDA method have been proposed by many researchers. Recently, there has been considerable interest in developing efficient algorithms, the so called manifold learning methods, to construct nonlinear low-dimensional manifolds from sample data points in high-dimensional spaces, and these methods have been regarded as effective approaches for nonlinear dimension reduction. In ISOMAP algorithm [2], pairwise geodesic distances of the data points instead of the Euclid distance are used with MDS (multidimensional scaling). The LLE (locally linear embedding) method [3] constructs a local geometric structure that is invariant to translations and orthogonal transformations in a neighborhood of each data point, and seeks to project the data points into a low-dimensional space that best preserves those local geometries. (A related method using Hessian matrices is presented in [4]). LTSA (local tangent space alignment) [5] methods constructs a local tangent space for each data point, and obtains the global low-dimensional embedding through affine transformation of the local tangent spaces.

Most of above nonlinear algorithms operate in a batch mode, meaning that all the data points need to be available during training. In applications like surveillance, where image data are collected sequentially, batch methods is computationally demanding: Repeating running the "batch" version whenever new data points become available is time consuming. Data accumulation is particularly beneficial to manifold learning algorithms due to their nonparametric nature. Another benefit for developing incremental methods is that the gradual changes in the data manifold can be detected. An incremental algorithm can be easily modified to be adaptive by incorporating "forgetting" effect. Another situation where incremental learning is useful is when there is an unbounded stream of possible data to learn from.

There have been some tries to create incremental manifold algorithms from their batch mode. In [6] Martin and Anil proposed two incremental algorithms considering the original ISOMAP and landmarked ISOMAP. An incremental LLE algorithm is proposed by Olga etc in [7]. In this paper, we have modified the LTSA algorithm so that it can update the low-dimensional representation of data points. Inspired by the landmarks using with the ISOMAP, we proposed an landmarked LTSA algorithm to reduce time complexity and memory requirement. Two incremental algorithms are proposed corresponding to the algorithms.

The main contribution of this study includes:

1. An landmark version of LTSA algorithm, where the landmark selection is based on LASSO [9]. This contrasts with previous work like [11], where random points are selected as landmark points.
2. Two incremental LTSA algorithms corresponding to original LTSA and landmark LTSA.
3. An incremental eigen-decomposition problem with increasing matrix size is solved by subspace iteration with Ritz acceleration. This is much efficient than solving a SVD problem from scratch.

2 LTSA

Given a set of data points $x_1,\ldots,x_N$ in a m-dimensional space R^m, LTSA assumes that the data lie on a (Reimannian) manifold and maps x_i to its d-dimensional representation τ_i in such a way that the local geometry information of x_i is reserved as much as possible. The local geometry information of x_i is defined as the local coordinates of the data points x_js in the neighborhood with respect to the tangent space of x_i. The power of LTSA can be demonstrated by the three-dimensional "Swiss-roll" data set in Fig.1, where points are colored according to their location on the manifold. When PCA is used to reduce the dimension to two (Figure 1b), points with different colors are mixed together, so disconnected regions on the manifold are mapped to similar locations. In LTSA (Figure 1c), the color of the points change gradually, indicating that the representation discover by LTSA faithfully corresponds to the structure of the curved manifold.

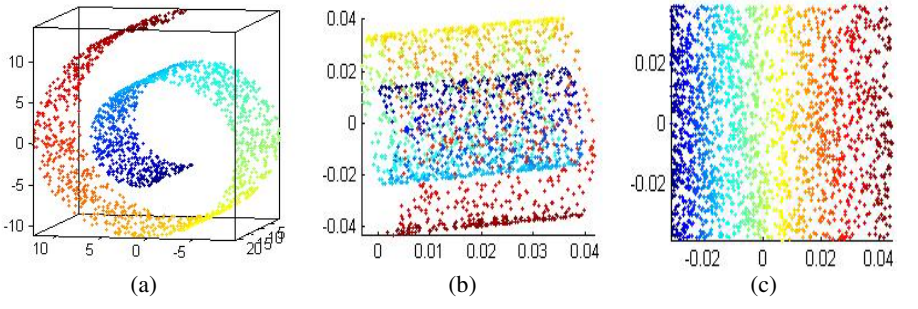

Fig. 1. LTSA on "Swiss roll" with 2,000 points, using knn neighborhood with k=8. (a) Points are colored according to their positions on the manifold. (b) Points with different colors are mixed together when they are plotted by the two PCA coordinates. (c) LTSA coordinates, a clear trend of the color is observed, indicating the structure of the manifold is recovered.

The LTSA algorithm has three stages. First, local information are extracted. LTSA requires the user to specify a parameter k, which is the number of neighborhoods used to construct local tangent spaces. For each x_i, let $X_i=[x_{i1},\ldots,x_{ik}]$ be a matrix consists of its k-nearest neighbors including x_i, say in terms of the Euclidean distance and $\overline{x_i}$ be the mean of X_i. LTSA determines X_i for each x_i firstly, then extracts local geometry information around x_i. Let $F = f(\Omega)$ is a parameterized manifold with $f:\Omega \subset R^d \to R^m$, while the Jacobi matrix of f at τ cannot be explicitly computed

without knowing the function *f*, the local tangent space T_τ at a fixed τ can be approximated using points in a neighbor set in the high-dimensional input space. Let Q_t be an orthonomal basis matrix of T_τ and $\theta_t^*(\bar{\tau})$ the local coordinate of $\bar{\tau}$ corresponding to T_τ. Then, Q_t and $\theta_t^*(\bar{\tau})$ can be calculated with SVD, so the local reconstruction error E_i can also be estimated.

LTSA proceeds to construct the alignment matrix. At the end of the first stage, a data point x_{ij} near x_i can be represented as $x_{ij} = \overline{x_i} + Q_i\theta_j^{(i)} + \varepsilon_j^{(i)}$, where $\varepsilon_j^{(i)} = (I - Q_i Q_i^T)(x_{ij} - \overline{x_i})$ denotes the reconstruction error. The global coordinates τ_i, i=1,…,N in the low-dimensional space are constructed based on the local coordinates $\theta_j^{(i)}$ which represents the local geometry. The local geometry information embedding by the $\theta_j^{(i)}$ are preserved as much as possible in the global coordinates.

The final step of LTSA recovers embedding coordinates τ_i. To uniquely determine T, the constraint $TT^T = I_d$ is imposed. The optimal T is given by the d eigenvectors of alignment matrix corresponding to the 2nd to d+1st small eigenvalues.

3 Incremental Version of LTSA (ILTSA)

The major computation cost of LTSA involves the computation of the smallest eigenvectors of the symmetric positive semidefined alignment matrix B. As new data arrive, these quantities usually do not change much: a new data point often changes the neighbors among only a subset of vertices, and the simple eigenvectors and eigenvalues of a slightly perturbed real symmetric matrix stay close to their original values. This justifies the reuse of the current transform matrix and coordinates for update. Compared to the incremental version of ISOMAP [6], the incremental LTSA is more suitable since it does not need the time consuming graph reconstruction problem, which is needed to calculate the geodesic distance between data points. More specificly, the structure of alignment matrix B in LTSA is highly local, and the influence of a new data is more local, which makes the updating of matrix very simple.

The problem of incremental LTSA can be described as follows. Assume that the low-dimensional coordinates t_i of x_i for the first n points are given. As a new sample x_{n+1} is observed, how should we update the existing set of t_i and find t_{n+1}? Our solution consists of three stages. The local geometry information are first updated in view of the new coming data x_{n+1}. The local coordinates of x_{n+1} with respect to subset of the

existing points are then used to estimate t_{n+1}. Finally, all t_i are updated in view of the coming data x_{n+1}.

The modification of the original LTSA for incremental updates will be described in section 3.1. In section 3.2, we proposed a new variant of LTSA (LLTSA) that utilizes the LASSO and LARS algorithms to select landmark points, because of LTSA is nonparametric, the data points themselves need to be stored, which limit the LTSA usage in huge data set. Compared to [6], where an incremental version of ISOMAP is proposed, the big difference is the landmarks are selected somewhat randomly, while in our method, the landmarks are selected following a more principal approach, as in [12]. An incremental version of LLTSA is also proposed in section 3.2.

3.1 Incremental LTSA (ILTSA)

When the new data x_{n+1} is observed, it only affects directly the coordinates of points which includes x_{n+1} in their k nearest neighborhoods, using X_A denotes this set of points. However, as the local tangent space of a point $x_i \in X_A$ is modified by the new point x_{n+1}, all the local coordinates of its neighbors need update. For each $x_i \in X_A$, let $X_i=[x_{i1},\ldots,x_{ik}]$ be a neighborhood matrix consisting of its k-nearest neighbors including x_i. The d-dimensional affine subspace approximation for data point in X_i is computed as

$$\min_{x,\theta,Q}\sum_{j=1}^{k}\left\|x_{ij}-(x+Q\theta_j)\right\|_2^2=\min_{x,\theta,Q}\left\|X_i-(xe^T+Q\Theta)\right\|_2^2 \tag{1}$$

where Q is of d columns and is orthonormal and $\Theta=[\theta_1,\ldots,\theta_k]$. Similar to PCA analysis, the optimal x is given by $\overline{x_i}$, the mean of all the x_{ij}'s and the optimal Q is given by Q_i, the matrix of d left singular vectors of $X_i(I-ee^T/k)$ corresponding to its d largest singular values, and Θ is given by Θ_i defined as

$$\Theta_i=Q_i^T X_i(I-ee^T/k)=[\theta_1^{(i)},\ldots,\theta_k^{(i)}],\ \theta_j^{(i)}=Q_i^T(x_{ij}-\overline{x_i}), \tag{2}$$

and $\theta_j^{(i)}$ incorporates local geometry information near x_i.

What we need is to construct the global coordinate τ_{n+1} in the low-dimensional space based on the given global coordinates τ_i, i=1,…,n, and the local coordinates $\theta_j^{(i)}$. In the same spirit of original LTSA, the principal of locating τ_{n+1} is to minimize the reconstruction errors $\varepsilon_j^{(i)}$, which is defined as

$$\varepsilon_j^{(i)}=\tau_{ij}-[\overline{\tau_i}+L_i\theta_j^{(i)}]\ ,\text{j=1,…,k},\ x_i\in X_A \tag{3}$$

where $\overline{\tau_i}$ is the mean of τ_{ij}'s, L_i is a local affine transformation matrix that need to be determined and $\varepsilon_j^{(i)}$ the local reconstruction error. Denoting $T_i = [\tau_{i1},...,\tau_{ik}]$ and $E_i = [\varepsilon_1^{(i)},...,\varepsilon_k^{(i)}]$, then we have $T_i = \frac{1}{k} T_i e e^T + L_i \Theta_i + E_i$, and the local reconstruction error matrix E_i then has the form

$$E_i = T_i (I - ee^T / k) - L_i \Theta_i . \tag{4}$$

To best preserve the local geometry information in the low-dimensional space, τ_i and L_i are sought to minimize the reconstruction errors $\varepsilon_{n+1}^{(i)}$, i.e.,

$$\sum_{x_i \in X_A} \left\| \varepsilon_{n+1}^{(i)} \right\|_2^2 = \sum_{x_i \in X_A} \left\| \tau_{n+1} (I - ee^T / k) - L_i \theta_{n+1}^{(i)} \right\|_2^2 = \min . \tag{5}$$

The optimal alignment matrix L_i that minimize the local reconstruction error $\| E_i \|_F$ for a fixed τ_i is given by $L_i = T_i (I - ee^T / k) \Theta_i^+$, and there for $E_i = T_i (I - \frac{1}{k} ee^T)(I - \Theta_i^+ \Theta_i)$.

To get the coordinates of τ_{n+1} given n known coordinates of x_i, i=1,…,n. We seek to minimize the local reconstruction error of x_{n+1} for each point $x_i \in X_A$, which is written as

$$\| \varepsilon_{n+1}^{(i)} \|_2^2 = \| \tau_{n+1}^{(i)} - [\overline{\tau_i} + L_i \theta_{n+1}^{(i)}] \|_2^2 = \| \tau_{n+1}^{(i)} - [\overline{\tau_i} + T_i \Theta_i^+ \theta_{n+1}^{(i)}] \|_2^2 . \tag{6}$$

As in LTSA, in the global low-dimensional coordinates, we want to minimize the reconstruction error:

$$\min_{\tau_{n+1}} \| \varepsilon_{n+1}^{(i)} \|_2^2 = \min_{\tau_{n+1}} \| \tau_{n+1} - [\overline{\tau_i} + T_i \Theta_i^+ \theta_{n+1}^{(i)}] \|_2^2, \; for \; x_i \in X_A , \tag{7}$$

τ_{n+1} is obtained by solving the above equations in the least square sense.

A related procedure is applied in [7] for LLE to calculate the coordinates of new data point. The eigenvalues of new data distance matrix are assumed the same as old data set. However, the assumption does not always hold in practice. In reality, if x_{n+1} is very near to a point x_i, the local geometry information of x_i will change enormously and so the eigenvalues. Our method does not assume the assumption, so it can overcome this situation.

After get the low-dimensional coordinates of new data point x_{n+1}, we need update the coordinates τ_i in view of the modified alignment matrix. This can be viewed as an incremental eigenvalue problem, since τ_i is obtained by eigen-decomposition. However, since the size of alignment matrix is increasing, traditional updating methods with same matrix size cannot be applied directly. An iterative scheme is used to update T by finding the eigenvales and eigenvectors of alignment matrix B_{new}. A good initial guess for the subspace of dominant eigenvectors of B_{new} is the column space of T^T. A better eigen-space is found by subspace iteration together with Rayleigh-Ritz acceleration [13]:

1. Compute $Z=B_{new}T^T$ and perform QR decomposition on Z, i.e., Z=QR and let V=Q.
2. Form $Z^*=V^TB_{new}V$ and perform eigen-decomposition of the d by d matrix Z*, let λ_i and u_i be the ith eigenvalue and the corresponding eigenvector.
3. $V_{new}=V[u_2 \ldots u_{d+1}]$ is the improved set of eigenvectors of B_{new}.

3.2 LTSA with Landmark Points

One drawback of the original LTSA is the quadratic memory requirement: the distance matrix is of size $O(n^2)$, making LTSA infeasible for large data sets. The same problem occurs in ISOMAP algorithm. In [11] landmark ISOMAP was proposed to reduce the memory requirement while lowering the computation cost and an incremental version of L-ISOMAP was proposed in [6]. In landmark ISOMAP, instead of finds all the pairwise geodesic distances, the methods finds a mapping that preserves the geodesic distances originating from a small set of landmark points. In the original L-ISOMAP, random points are used as landmark points. In the [6], the vectors corresponding to the largest d singular value of centered geodesic distance matrix are used as landmark points. Least Absolute value Subset Selection Operator(LASSO) [9] is a shrinkage and selection method for linear regression. It minimizes the usual sum of squared errors with a bound on the sum of the absolute values of the coefficients. Finding the LASSO solutions used to require solving a quadratic programming problem, until the development of the Least Angle Regression(LARS) procedure [10], which is much faster and not only gives the LASSO solutions but also provides an estimator of the risk as a function of the regularization tuning parameter. LASSO with the LARS are used in [12] to select landmarks for ISOMAP algorithm. We follows the similar procedures to select landmarks for LTSA algorithm.

3.2.1 Landmark Selection Based on LASSO and LARS

Let X be the n data points set in R^m, i.e., $X = [x_1 ... x_n]$, and T be the corresponding n d-dimensional point set in low-dimensional space. The sacristy of LTSA in achieved by finding an estimate $\hat{\beta}$ that minimizes the function

$$E = \| \theta - K\hat{\beta} \|^2 + \gamma \| \hat{\beta} \|_q^q \tag{8}$$

where K={k_{ij}}, $k_{ij} = \exp(-\frac{\| x_i - x_j \|}{2\sigma_k^2})^2$ is a Gaussian kernel, $\theta \in R^n$ and $\hat{\beta} \in R^n$, γ is a tunning parameter that controls the amount of regularization. $\hat{\beta}$ is the parameter column vector, and $\| \hat{\beta} \|_q$ denotes the l_q norm of $\hat{\beta}$, i.e., $\sqrt{\sum_{i=1}^{n} | \hat{\beta}_i |^q}$. For the most sparseness, the ideal value of q would be zero. However, minimizing E with the l_0 norm is prohibitive in computational terms. A sub-optimal strategy is to use q=1 instead. This is the usual formulation of a LASSO regressive problems, which is traditionally solved using quadratic programming. The recent development of the LARS method has made this unnecessary.

An important factor of the method is the choose of θ, which influences the process of landmark points selecting. In [12], the θ is chosen as $\theta = [\theta_1 ... \theta_n]^T$, where θ_j equals to the maximum principal angle between $T_{Xu}(M)$ and $T_{Xj}(M)$, x_j is the jth column of X and x_u is the mean of X, $T_{Xu}(M)$ and $T_{Xj}(M)$ are the tangent subspace at x_u and x_j respectively. The principal angles and efficient algorithms to compute them can be found in [14]. The local tangent subspace can be found by local SVD, which is calculated during the original LTSA, so there would be litter extra computational burden. A big difference between our method compared with the method in [12], is that the θ_j in our method is more local, here θ_j is defined as the maximum principal angle between $T_{\bar{X}_{ju}}$ and T_{xj}, where $\bar{x}_{ju}$ is the mean of X_j, which is the neighbor set of x_j. The choice is in the same spirit as LTSA, which in principal is more local compared with

ISOMAP, and also note the geometry information near a point x_i embedding in LTSA is determined by its near neighbors.

Briefly, LARS starts with $\hat{\beta} = 0$ and adds covariates (the column of K) to the model according to their correlation with the prediction error vector, $\theta - K\hat{\beta}$, setting the corresponding $\hat{\beta}_j$ to a value such that another covariate becomes equally correlated with the error and is, itself, add to the model. LARS then proceeds in a direction equiangular to all the active $\hat{\beta}_j$ and the process is repeated until all covariates have been added. There are a total of m steps, each of which adds a new $\hat{\beta}_j$, making it non-zero. With slight modification, these steps correspond to a sampling of the tuning parameter γ in (8) under LASSO. Furthermore, the risk is shown can be estimated as $R(\hat{\beta}_p) = \| \theta - K\hat{\beta}_p \|^2 / \bar{\sigma}^2 - m + 2p$, where p is the number of non-zero of $\hat{\beta}_j$, and $\bar{\sigma}^2$ can be found from the unconstrained least square solution.

The landmarks are the columns x_j of X with the same index j as non-zero element of $\hat{\beta}_j$, where $p = \arg\min_p R(\beta_p)$. There are $n' = p$ landmarks, as there are p non-zero elements in β_p.

3.2.2 Incremental Landmark LTSA (ILLTSA)

Without the generality, let the first u points, i.e., $x_1,\ldots,x_u$ be the landmark points, denote the point set with X_L. For a data point x_i, instead of finding the k minimal distance among all the data point X, the landmark LTSA(LLTSA) finds the k minimal distance neighbors among the small set landmark points X_L, and use this information to construct local tangent space. In the LLTSA, the size of distance matrix $D = \{d_{ij}\}$ is u*n, where d_{ij} is the distance between x_i(a landmark point) and x_j. The local tangent space of x_{n+1} is constructed with the local geometry information with X_L. The coordinate of the new point x_{n+1} is determined by solving a Least-Square problem similar to that in section 3.1. The difference is that the columns among X_L instead of X, are used. Finally,

subspace iteration together with Ritz acceleration is used to improve singular vector estimates. The steps are the following:

1. Perform SVD on the matrix BT, $U_1 S_1 V_1^T = BX$.

2. Perform SVD on the matrix $B^T U_1$, $U_2 S_2 V_2^T = B^T U_1$.

3. Set $T_{new} = U_2 (S_2)^{1/2}$ and $Q_{new} = U_1 (S_2)^{1/2}$.

Similarly, the updated coordinates are the eigenvectors corresponding to 2~d+1 smallest eigenvalues.

4 Experiments

In order to evaluate the methods proposed, we have conducted several experiments on synthetic data sets and real datasets. The main algorithm is implemented in Matlab. The running time is measured on a 2.1 GHz PC with 1G memory running Windows XP.

4.1 Incremental LTSA(ILTSA)

The accuracy and the efficiency of the basic incremental algorithm is evaluated by comparing it with the batch version on several data sets. The first experiment is on the 3 dimensional Swiss roll data set, the data set is also used in the original LTSA. Initialization is done by finding the coordinate estimate x_i for 100 randomly selected points using the original "batch" LTSA, with the neighborhood size k=8. Random points from the S-curve data set are then added one by one, until 2,000 points are accumulated. The incremental algorithm described in Section 3.1 is used to update the coordinates. Figure 2 shows several snapshots of the algorithm. In the first column, the circles and cross in the figures represent the coordinates estimated by the batch and the incremental version ILTSA respectively. The second column contains scatter plots, where the color of the points correspond to the coordinates of the first column. The third column illustrates the neighborhood structure graphs. Snapshots with 100, 500, 1,000 points are shown. The cross and the circles match very well, indicating that the coordinates updates by the incremental LTSA follow closely with the coordinates estimated by the batch version for different number of points.

To quantify the accuracy of the coordinate update of the incremental algorithm ILTSA, we adopt an error measure[6] defined as the square root of the mean square error between $\hat{\tau}_i^{(n)}$ and $\tau_i^{(n)}$, normalized by the total sample variance:

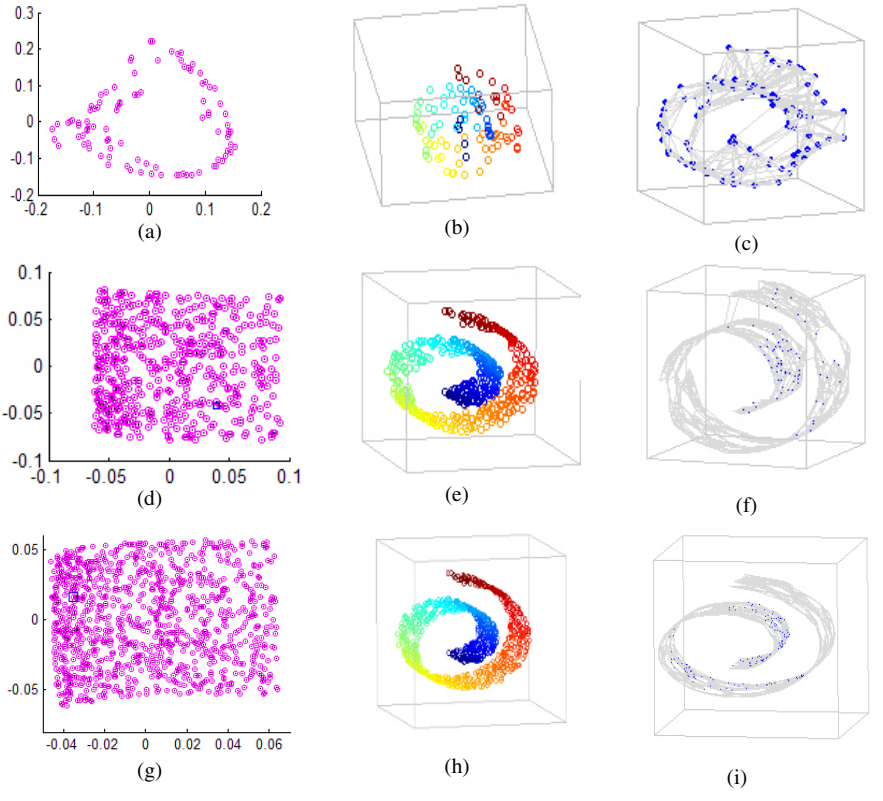

Fig. 2. Snapshots of "Swiss roll" for incremental LTSA. Incremental LTSA was initialized by running the "batch" LTSA with 100 points((a) to (c)). Snapshots with 500 and 1,000 are shown in (d) to (f) and (g) to (i) respectively.

$\varepsilon_n = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \| \tau_i^{(n)} - \hat{\tau}_i^{(n)} \|^2 / \sum_{i=1}^{n} \| \tau_i^{(n)} \|^2}$. ε_n against the number of data point n for Swiss roll data set is presented in Figure 3a. From the figure, we can see that the proposed updating method is fairly accurate with an average error of 0.08 percent. The computation time is show in Table 1. Our incremental approach has significant saving in main aspects of LTSA: the global coordinates update. Note that both the batch and incremental versions need the same number of distance computations.

Similar experimental procedure is applied to other data sets. The "S-curve" data set contains points in a 3D space with an effective dimensionality of two, which is a standard benchmark for manifold learning. The "rendered face" data set contains 698 face images with size 64*64 rendered at different illumination and pose conditions.

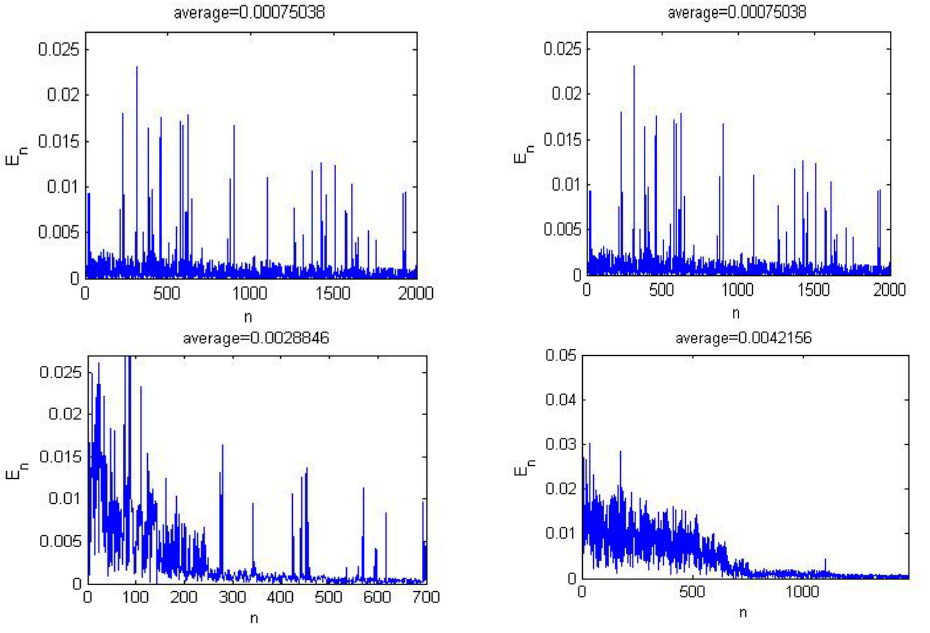

Fig. 3. Approximation error (ε_n) between the coordinates estimated by the basic incremental LTSA and the basic batch LTSA for different numbers of data points (n). (a) Swiss roll, (b) S-curve, (c) Rendered Faces. (d) MNIST digit 2.

Table 1. Runtime (Seconds) for Batch and incremental LTSA

	Swiss roll		S-curve		Rendered face		MNIST 2	
	Batch	Incr.	Batch	Incr.	Batch	Incr.	Batch	Incr.
Computing t_{n+1}	31.76	0.43	28.96	0.56	3.47	0.07	32.85	0.52
Updating t_i		5.12		5.53		0.86		4.38

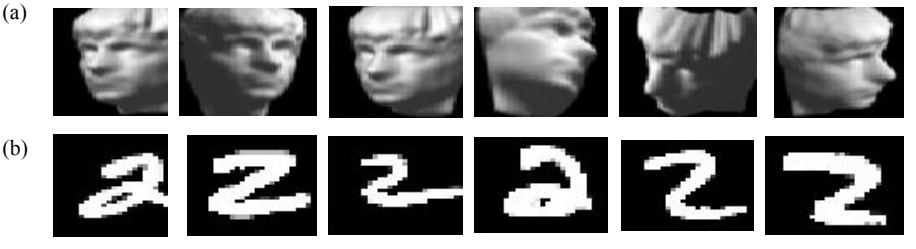

Fig. 4. Example images of data sets. (a) rendered face. (b) MNIST digit 2.

"MNIST digit 2" is a 576-dimensional data set derived from the digit images "2" from MNIST and contains 28 by 28 digit images. Some examples for "rendered face" and "MNIST digit 2" are shown in Figure 4. All the above data set are also used in [6]. The neighborhood size for MNIST digit 2 and "rendered face" is set to 10 to demonstrate that the proposed approach is efficient and accurate irrespective of the neighborhood used. The approximation error and the computation time for these data set are shown in Figure 3 and Table 1. We can see that the incremental LTSA is accurate and efficient for updating the coordinates in all these data sets.

4.2 Experiments on Landmark LTSA

A similar experimental procedure is applied to the incremental landmark LTSA described in Section 3.2 for Swiss roll, S-curve, rendered face, MNIST digit2 data sets. 300 randomly points from the data set are selected at start, points are then added one by one randomly until 5,000 points are accumulated. For the data set less than 5000, the procedure stops when all the data point are used. 100 points from the initial 300 points are selected to be the landmark points following the LASSO procedure in section 3.2.1. The snapshots for incremental LLTSA are fairly similar to those for incremental LTSA in Fig.2 and are omitted here. The approximation error and the computation time for the batch and incremental version of landmark LTSA are shown in Fig. 5 and Table 2 respectively. Once again, the coordinates estimated by the incremental version are accurate with respect to the batch version, and the computation time is much less.

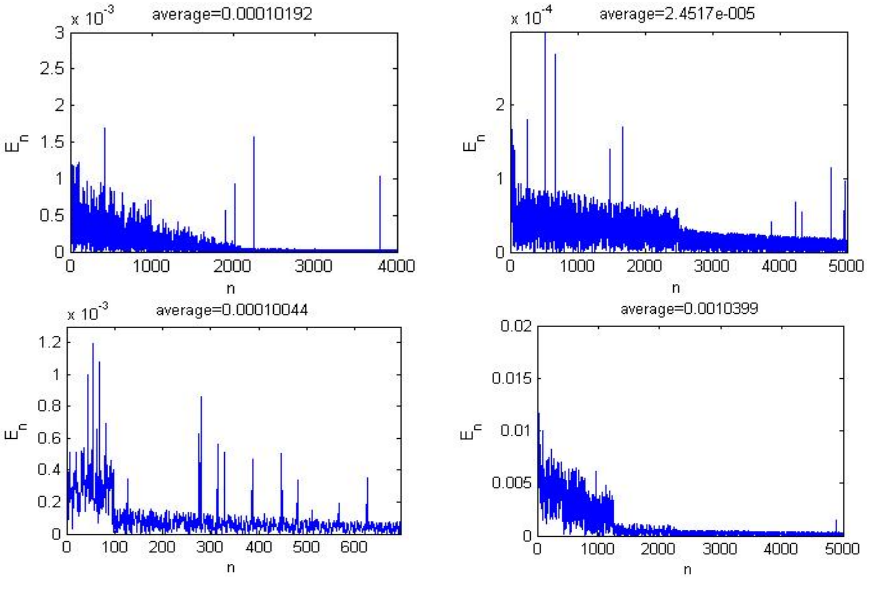

Fig. 5. Approximation error (ε_n) between the coordinates estimated by the incremental landmark LTSA and the batch landmark LTSA for different numbers of data points (n). (a) Swiss roll. (b) S-curve. (c) Rendered Faces. (d) MNIST digit 2.

Table 2. Runtime (Seconds) for Batch and incremental Landmark LTSA

	Swiss roll		S-curve		Rendered face		MNIST 2	
	Batch	Incr.	Batch	Incr.	Batch	Incr.	Batch	Incr.
Computing t_{n+1}	11.72	0.35	12.06	0.72	2.13	0.09	10.58	0.42
Updating t_i		3.21		3.54		0.87		2.76

5 Conclusion

Nonlinear dimensionality reduction is an important problem with applications in pattern recognition, computer vision and data mining. We have proposed an algorithm (ILTSA) for incremental nonlinear mapping problem by modifying the LTSA algorithm. The core idea is to efficiently reestimate the eigenvectors using the previous computation results. A landmark version of LTSA (LLTSA) is also proposed, where the landmark points are selected based on LASSO and LARS regression. The proposed algorithm finds geometrically meaningful landmarks and avoids expensive quadratic programming computations. Furthermore, an incremental LLTSA (ILLTSA) algorithm is also proposed for the landmark version of LTSA. The proposed methods have been validated on synthetic and real datasets.

Acknowledgement

The work has been supported by the National High-Tech. R&D Program for CIMS, China (No.2003AA411021) and the highlight R&D Program of Zhejiang Province (No.2004C11053, No. 2005C21078).

References

1. Seung, S., Daniel, D.L., The manifold ways of perception. Science, 2000, 290(5500): 2268-2269.
2. Tenenbaum, J.B., de Silva, V., and Langford, J.C.,, A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, 2000, 290(5500): 2319-2323.
3. Roweis, S.T. and Saul, L.K., Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290(5500): 2323-2326.
4. David, L.D., Caroe, G., Hessian eigenmaps Locally linear embedding techniques for high-dimensional data. Proceedings of the National Academy of Sciences of the United States of America, 2003. 100(10): 5591-5596.
5. Zhang, Z.Y., Zha, H.Y., Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM Journal of Scientific Computing, 2004. 26(1): 313-338.
6. Martin H.C. Law, A.K.J., Incremental Nonlinear Dimensionality Reduction by Manifold Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006. 28(3): p. 377-391.

7. Olga Kouropteva, O.O., Matti Pietikainen, Incremental locally linear embedding. Pattern Recognition, 2005. 38: p. 1764-1767.
8. ZhenYue Zhang, H.Z. A Domain Decomposition Method for Fast Manifold Learning. in Advances in Neural Information Processing Systems. 2006: MIT Press.
9. T. Hastie, R. Tibshirani, and J. H. Friedman. The Elements of Statistical Learning. Springer, 2001.
10. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. Annals of Statistics, 2003.
11. V. de Silva, J.B.T. Global versus Local Approaches to Nonlinear Dimensionality Reduction. in Advances in Neural Information Processing Systems 15. 2003.
12. Jorge Gomes da Silva, J.S.M., Jo?o Manuel Lage de Miranda Lemos. Selecting Landmark Points for Sparse Manifold Learning. in Advances in Neural Information Processing Systems. 2006. Vancouver, Canada: MIT Press.
13. G.H. Golub, C.F.V.L., Matrix Computations. 1996: Johns Hopkins University Press.
14. A. Bjorck, G.H.G., Numerical methods for computing angles between linear subspaces. Mathematics of Computation, 1973. 27(123): p. 579-594.