A Manifold Learning Algorithm Based on Incremental Tangent Space Alignment

Chao Tan(✉) and Genlin Ji

School of Computer Science and Technology,
Nanjing Normal University, Nanjing, China
{chtan,glji}@njnu.edu.cn

Abstract. Manifold learning is developed to find the observed data's low dimension embeddings in high dimensional data space. As a type of effective nonlinear dimension reduction method, it has been widely applied to data mining, pattern recognition and machine learning. However, most existing manifold learning algorithms work in a "batch" mode and cannot effectively process data collected sequentially (or data streams). In order to explore the intrinsic low dimensional manifold structures in data streams on-line or incrementally, in this paper we propose a new manifold **L**earning algorithm based on **I**ncremental **T**angent **S**pace **A**lignment, *LITSA* for short. By constructing data points' local tangent spaces to preserve local coordinates incrementally, we can accurately obtain the low dimensional global coordinates. Experiments on both synthetic and real datasets show that the proposed algorithm can achieve a more accurate low-dimensional representation of the data than state-of-the-art incremental algorithms.

Keywords: Nonlinear dimension reduction · Manifold learning · Incremental tangent space

1 Introduction

Dimension reduction is a prerequisite for many tasks in data mining, pattern recognition and machine learning. Accordingly, developing effective methods for dimension reduction has been a hot research topic in recent years.

Specifically, manifold learning, a type of effective nonlinear dimension reduction method, aims to discover the low dimensional smooth manifold structure of observed high-dimensional data. In the past decade, a number of manifold learning algorithms are developed, including Isometric feature mapping (Isomap) [1], Laplacian eigenmaps algorithm [2], locally linear embedding (LLE) [3], Local Tangent Space Alignment (LTSA) [4] and so on. All these algorithms operated in a "batch" mode, i.e., the whole set of data must be collected before running the algorithm, which makes it ineffective to deal with data collected sequentially (or data streams). In reality, however, many applications need to handle realtime data streams where data are collected in turn. Therefore, incremental

© Springer International Publishing AG 2016
X. Sun et al. (Eds.): ICCCS 2016, Part II, LNCS 10040, pp. 541–552, 2016.
DOI: 10.1007/978-3-319-48674-1_48

manifold learning methods being able to continuously and efficiently update the manifold constructed on new-coming data and existing data without repeatedly computing the whole data set are required. Meanwhile, the incremental learning algorithms can be used to visualize the data manifold's changing process, which provides a great help to understand the feature of data structure.

In this paper we propose a new incremental manifold learning algorithm called *LITSA*, which finds the local principal components of the sample points by incremental PCA. *LITSA* is not subject to the size of sample points' covariance matrix, thus solving the large-scale matrix eigen decomposition problem in LTSA. In this method, in order to obtain its coordinates in a low dimension space, we construct the tangent space of each data sample in high dimension space incrementally. To avoid reconstructing the eigenspace repeatedly, we construct the local tangent space matrix incrementally in the light of the sample points' covariance matrix. Thus, the proposed algorithm can update the manifold structure directly via computing the eigenvectors of local tangent space matrix constructed on the basis of existing samples and newly-arrived data samples. The experimental results on both synthetic and real-world datasets show that the proposed algorithm can achieve a more accurate low-dimensional representation of the original data efficiently.

2 Related Work

The idea of existing incremental manifold learning algorithms can be roughly summed up into two categories.

The first one is based on the hypothesis that existing results are entirely correct, so the the new samples can be efficiently dealt with. These algorithms can efficiently deal with the new data. However, existing data often can not accurately reflect the intrinsic manifold structure. Especially in the case of non-uniform sampling, these algorithms may not provide low dimensional embedding of high-dimensional data correctly.

Algorithms in the second class will update the training data set's embedding coordinates when embedding the other samples. So they can better reflect the dataset's characteristics. By contrast, these methods can give more reliable results.

Existing typical incremental manifold learning algorithms include the incremental Isomap algorithm (IIsomap) [5], the incremental version of Laplacian eigenmaps (ILE) [6], the incremental LLE algorithm (ILLE) [7], and the incremental LTSA algorithm (ILTSA) [8]. Another incremental manifold learning algorithm via LTSA by Liu et al. [9] proposed a modified LTSA algorithm and an incremental eigen-decomposition problem with increasing matrix size.

The core idea of the IIsomap algorithm is to efficiently update the geodesic distances and re-estimate the eigenvectors, using the previous computation results. There exists the problem of disconnected neighborhood graphs when the data are undersampled or unevenly distributed, new points may change the current neighborhoods and local distribution of the manifold, the continuity of the neighborhood matrix is not guaranteed. The addition of a new sample can delete critical edges in the graph and subsequently change the geodesic distances dramatically. In this case, short-circuit or cavitation phenomenon will happen. The algorithm of ILE is traditionally performed in a batch mode. It introduces a locally linear reconstruction mechanism to add new adjacent information and revise the existing samples' low-dimensional embedding results. The sub-manifold method involved needs to solve a $(k+1) \times (k+1)$ eigenvector problem and the overall time complexity of solving the eigenvector problem is $O((k+1)^3)$. ILLE algorithm evaluated the mapping results of the new samples and re-calculated the projections of original samples. This algorithm assumed that the eigenvalues of the cost matrix remained the same when a new data point arrived and the incremental learning problem of LLE was tackled by solving a $d \times d$ minimization problem, where d was the dimensionality of the low-dimensional embedding space. In ILTSA the previous work of incremental algorithms mostly adopted iterative methods of whole process which need abundant repetitive computation, because one does not know what part of a iterative optimization problem has to be recomputed. The problem in method [9] is similar to ILTSA that the alignment matrix has to be reconstructed to include the new point, which is not very practical for large datasets. The other limitation of existing incremental methods is that there is no guarantee on the approximation error, so these approaches suffer from unpredicted approximation error [10]. Similar problems also exist in other incremental methods such as ILLE algorithm, the error in process of dimension reduction will become larger, because the eigenvalues of new cost matrix M_{new} in this algorithm have not been updated. As the number of new samples increases, the difference value between the first d smallest eigenvalues of the original cost matrix M will become larger, hence a larger error [11].

Given a set of N data points $\{x_i|i = 1, 2, \cdots, N\}$, Local Tangent Space Alignment (LTSA) assumes that the data points are sampled from a high dimensional manifold, i.e. located in a m-dimensional manifold $\mathbb{R}^m$. It maps x_i to the d-dimensional representation τ_i and reserves the local geometry information of x_i as much as possible, where $d < m$. Local geometry information of x_i is defined as the manifold constructed by its k nearest neighbor points to generate tangent space of x_i. All the data points in tangent spaces are then aligned to give their global coordinates in the low dimension manifold. Though LTSA can effectively evaluate a dataset's global mapping coordinates that reflects the data set's low-dimensional manifolds structure, it has two shortcomings: On the one hand, the size of the covariance matrix used for eigen decomposition in LTSA is equal to the number of samples, so it is inefficient to handle large datasets; On the other hand, it cannot deal with new sample point effectively for the high time complexity, so it is difficult for incremental learning.

Min et al. [12] proved that the local tangent space of a sample can be represented by the eigenvectors of the covariance matrix constructed by the samples in its neighborhood. The matrix's eigenvectors can be computed by local principal component analysis method. Therefore the problem of computing the sample points' projection coordinates in the low dimensional space can be transformed into solving the local principal component analysis problem.

Recently, a number of semi-supervised feature extraction algorithms [13–16] have come out, which combine semi-supervised techniques with local discriminant analysis approaches. While they have lost considerations of incremental learning on dynamic manifold, which has become a hot topic in big data stream nowadays.

The algorithm proposed in this paper can overcome these shortcomings. It finds the local principal components of the sample points by incremental PCA [17,18], which is not subject to the size of sample points' covariance matrix, thus solving the large-scale matrix eigen decomposition problem in LTSA. In addition, by taking the adaptive factor into account, our algorithm is able to deal with new arrived sample points effectively.

3 Manifold Learning Algorithm Based on Incremental Tangent Space Alignment

3.1 Update New Covariance Matrix After Inserting New Point

Firstly, we construct new local tangent spaces for existing sample points and newly arrived one. Given a set of data points $X = [x_1, \cdots, x_N]$ sampled from non-linear manifold $\mathbb{R}^m$, they will be mapped from m dimension space to d dimension space. Suppose $X_i = [x_{i_1}, \cdots, x_{i_k}]$ as a matrix containing k nearest neighbours of x_i (in Euclidean distance). Traditional PCA searches for vectors c, T and matrix U to project X_i to low dimension manifold $\mathbb{R}^d$. In order to get the optimal solution, we minimize the reconstruct error: $\min ||E|| = \min\limits_{c,U,T} ||X - (ce^T + UT)||_F$, in which c is the mean value of X, indicated as: Xe/N. Matrix $U \in \mathbb{R}^{m \times d}$ is a set of orthonormal basis of affine subspace, singular value decomposition in linear condition can be used to solve this problem. While in nonlinear conditions, especially in realtime environment, situation is more complicated. Local linear array must be used in realization to solve incremental non-linear mapping problem.

When a new point arrives, denoted by x_{new}, we prepare to update the local information of the new point from eigenvectors of existing points. Suppose the eigenspace of existing N samples has already been constructed, with k-neighbor points of each point. The new point's coordinate X_{new} is projected into existing eigenspace through following form: $w_i = u_i^T(X_{new} - \overline{X}_N), i = 1, 2, \cdots, N$,where u_i is the i^{th} eigenvector in eigenspace which consists of N sample points, $\overline{X}_N$ is mean of the N samples, w_i is the i^{th} coefficient of the new point in the eigenspace.

Then we can reconstruct the new point's coordinate based on these coefficients: $X_{new} = w_i u_i + \overline{X}_N$. To solve incremental nonlinear mapping problem, the first step is to estimate mean value and construct the sample points' covariance matrix, thereby to determine the sample points' eigenvectors in eigenspace.

Let M_N be the mean value of the existing N samples in m dimension space. In this paper we propose a novel formulation M_{new} in order to describe the geometric information of the existing sample points' covariance matrix after inserting the new point:

$$M_{new} = \alpha M_N + (1-\alpha) x_{new} e^T, \tag{1}$$

where α is an adaptive factor controlling how exited samples influence the estimation of M_{new}'s value, e denotes the identity vector. How to choose α is based on the incremental environment: if the samples in current environment update fast, α is set to be small; if slow, α has larger value. We set $(1-\alpha)$ as the *learning rate* which depends on the nature of data sets in realtime practice and requires a $trial - and - error$ procedure to determine, while it is impractical for "on-line" applications [19]. This process lies on the measurement of observed data samples X_{new}. From [20] we know the sample mean is an efficient estimate of mean distribution. The use of samples' mean value M_N also inspires us to divide the learning rate by $\frac{1}{N}$ here. So we set $(1-\alpha) = \frac{1}{N}$ as sample mean. Although $\alpha = \frac{N-1}{N}$ added on M_N is close to 1 when N is large enough, it is very important for fast convergence with primary samples. If the estimate does not converge well at the beginning, it is harder to be pulled back later when N is large. We can also know that the estimate of M_{new} has a fairly low error variance and high statistical efficiency through experiments in subsequent section.

Traditional PCA (Principle Component Analysis) computes the eigensolutions by solving the SVD (Singular Value Decomposition) of the covariance matrix: $C = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^T(x_i - \bar{x})$. The corresponding expression in matrix form can be written as: $C = (X_i - \bar{x_i}\, e^T)^T (X_i - \bar{x_i}\, e^T)$. Compute C's eigenvectors corresponding to d first eigenvalues which span a subspace of d dimension. We only need to save M_{new} and existing covariance matrix C_n because existed samples can be discarded during recursive updating process, thereby the algorithm's complexity can be greatly decreased. Similarly we can obtain the iterative estimation of covariance matrix given as below:

$$C_n = \alpha C_{n-1} + (1-\alpha)(X_n - \overline{X}_n)^T (X_n - \overline{X}_n) \tag{2}$$

After substituting α with $\frac{N-1}{N}$ and inserting new sample x_{new}, we get the updating form C_{new} for C as follows, where X_{new} is the matrix form of $x_{new} e^T$.

$$C_{new} = \frac{N-1}{N} C + \frac{1}{N} (X_{new} - M_{new})^T (X_{new} - M_{new}) \tag{3}$$

According to PCA, the optimal solution of $C = (X_i - \bar{x_i}\, e^T)^T (X_i - \bar{x_i}\, e^T)$ is given by SVD of $(X_i - \bar{x_i}\, e^T)$. Let $\lambda_i (i = 1, \cdots, d)$ be the d first eigenvalues of C, v_i as the corresponding eigenvectors, so: $C \approx U_{Nd} \Lambda_{dd} U_{Nd}^T = \sum_{i=1}^{d} \lambda_i v_i v_i^T$, where the column vectors U_{Nd} are the eigenvectors v_i of C, and diagonal matrix Λ_{dd} is comprised of $C's$ eigenvalues λ_i.

Now we can get the incremental C_{new} in matrix form and its expression model:

$$C_{new} = \frac{N-1}{N} U \Lambda U^T + \frac{1}{N}(X_{new} - M_{new})^T (X_{new} - M_{new}) \tag{4}$$

To facilitate the calculation, the above formula can be written as:

$$C_{new} = \frac{N-1}{N} \sum_{i=1}^{d} \lambda_i v_i v_i^T + \frac{1}{N}(X_{new} - M_{new})^T (X_{new} - M_{new}) \tag{5}$$

where:

$$X_{new} - M_{new} = x_{new} e^T - \frac{N-1}{N} \bar{x_i}\, e^T - \frac{1}{N} x_{new} e^T = \frac{N-1}{N}(x_{new} - \bar{x_i}) e^T \tag{6}$$

3.2 Construct the Local Tangent Space in Incremental Form

For the updating of all the points' embedding coordinates in low dimensional space by an incremental way, we proceed in the following steps.

Firstly, we perform eigen-decomposition for C_{new} and obtain its first d eigenvalues λ_i and eigenvectors v_i, $i = 1, 2, ..., d$. Then we construct $d + 1$ vectors based on v_i and λ_i: $A = [y_1, y_2, \cdots, y_{d+1}]$ of $k \times (d + 1)$ size, where: $y_i = v_i \sqrt{\frac{N-1}{N} \lambda_i}, i = 1, \cdots, d, y_{d+1} = \sqrt{\frac{1}{N}} (X_{new} - M_{new})^T$. It can satisfy the condition of $C_{new} = AA^T$, which indicates the local tangent space matrix of neighbour points of the new sample including itself.

Secondly, an inner-product matrix B of $k \times k$ size can be formulated as: $B = AA^T$, much smaller than C_{new}. The advantage of this approach is that we can construct local tangent space matrix incrementally instead of recomputing the new covariance matrix C_{new}. The matrix B is the orthogonal projector onto the subspace spanned by the columns of A. So the local coordinates of new point x_{new} in the incremental subspace can be obtained by computing the d first singular vectors of A [21], equivalently the d eigenvectors $u_1, \cdots, u_d$ corresponding to the d smallest eigenvalues of B. Set the eigenvectors and eigenvalues of B as: $\{v_n^i\}$ and $\{\lambda_n^i\}$, we have: $Bv_n^i = \lambda_n^i v_n^i$, $i = 1, 2, ..., k$. Then, $A^T A v_n^i = \lambda_n^i v_n^i$, multiplied by A on both sides: $AA^T A v_n^i = \lambda_n^i A v_n^i$. Let $u_n^i = A v_n^i$, then: $AA^T u_n^i = \lambda_n^i u_n^i$, i.e.: $C_{new} u_n^i = \lambda_n^i u_n^i$. So we have C_{new}'s eigenvectors: $u_n^i = A v_n^i$.

With eigenvectors of the covariance matrix in incremental form, new sample point x_{new}'s local coordinate can be calculated by formulation Eq. (7):

$$x_{new} = w_i u_i + \bar{X}_N = \sqrt{\lambda_n^i} u_n^i + \bar{X}_N = \sqrt{\lambda_n^i} A v_n^i + \bar{X}_N \tag{7}$$

3.3 Compute the Low Dimensional Global Coordinates

Now we consider the incremental updating situation. The local coordinates of new sample point in Eq. (7) can be represented in following matrix form:

$$X_{new} = WU + \overline{X}_N, \tag{8}$$

where the low-rank matrix WU is the d-dimensional optimal approximation of original coordinates, U is orthonormal and W has k column vectors: $U = [u_1, \cdots, u_d]$, $W = [w_1, \cdots, w_k]$. $\overline{X}_N$ is the mean of X_N.

Since the structure of global coordinates is linear in low dimensional feature space, the new sample point's local coordinate can be approximated by Eq. (8). Then we can construct global coordinates according to the local coordinates:

$$Y_i = \frac{1}{k} Y_i ee^T + L_i(WU + \overline{X}_N) + E_i, i = 1, ..., N. \tag{9}$$

Y_i denotes the global coordinates in low dimensional space, L_i and E_i denote local affine transformation matrix and reconstruction error respectively after inserting new data point. So,

$$E_i = Y_i(I - \frac{1}{k} ee^T) - L_i(WU + \bar{X_N}) \tag{10}$$

To minimize the reconstruction error we can describe Eq. (10) as: $\min\limits_{Y_i, L_i} Y_i(I - \frac{1}{k}ee^T) - L_i(WU + \bar{X_N})$, then:

$$L_i = Y_i(I - \frac{1}{k} ee^T)(WU + \bar{X_N})^\dagger \tag{11}$$

where: $(WU + \bar{X_N})^\dagger = ((WU + \bar{X_N})^T (WU + \bar{X_N}))^{-1} (WU + \bar{X_N})^T$ is pseudo-inverse of $(WU + \bar{X_N})$. So:

$$E_i = Y_i(I - \frac{1}{k} ee^T)(I - (WU + \bar{X})^\dagger (WU + \bar{X})) \tag{12}$$

Suppose $H_i = (I - \frac{1}{k}ee^T)(I - (WU + \overline{X})^\dagger (WU + \overline{X}))$, to determine Y_i uniquely $Y_i Y_i^T = I_d$ is the constraint condition. Let $Y = [Y_1, \cdots, Y_N]$ and S_i be the 0-1 selection matrix such that $YS_i = Y_i$, we have:

$$||E_i||_F^2 = ||Y_i H_i||_F^2 = ||YSH||_F^2 = Tr(H^T S^T Y^T YSH) = Tr(H^T S^T SH) \tag{13}$$

Then we minimize the reconstruction error: $\min ||E_i||_F^2 = \min Tr$ $(H^T S^T SH)$, Y in Eq. (13) is constituted with d eigenvectors corresponding to the 2^{nd} to $d + 1^{st}$ smallest eigenvalues of matrix D: $D = H^T S^T SH$, where $S = [S_1, \cdots, S_N]$ and $H = diag(H_1, \cdots, H_N)$.

In order to solve the eigenvalues and eigenvectors of D locally and linearly, we can construct U_i based on eigenvectors $u_1, ..., u_d$ of B: $U_i = [e/\sqrt{k}, u_1, \cdots, u_d]$, then, $H_i = I - U_i U_i^T$. After new data points arrive, we update incremental alignment matrix: $D(I_i, I_i) \leftarrow D(I_i, I_i) + I - U_i U_i^T$, then compute the 2^{nd} to the $(d+1)^{th}$ smallest eigenvalues and the corresponding eigenvectors $[t_2, \cdots, t_{d+1}]$ of D, which correspond to the optimal low dimensional global coordinates of data points.

4 Implementation of LITSA

Given a dataset with N m-dimensional points $X = [x_1, \cdots, x_N]$ sampled from manifold $\mathbb{R}^m$ and a new point x_n which will be inserted into X. The proposed algorithm projects these points into low dimensional space, provides d-dimensional $(d < m)$ coordinates $y_1, ..., y_N$ and the new point's low dimensional mapped point y_n in an incremental way.

Step 1. Update new covariance matrix.

- Determine k nearest neighbours of each point x_i (Euclidean distance) and compose the matrix X_i, $i = 1, ..., N$;
- Construct the covariance matrix C_n in Eq. (2) iteratively, which represents the incremental tangent space of each data point;
- Construct matrix A and B;
- Compute the eigenvectors $u_1, ..., u_d$ corresponding to the d smallest eigenvalues of B;
- Set $U_i = [e/\sqrt{k}, u_1, \cdots, u_d]$.

Step 2. Construct alignment matrix in incremental form.

Set $I_i = \{i_1, \cdots, i_k\}$ as k nearest neighborhood's index set of data point x_i , compute alignment matrix in incremental form:

```
Initialize $D \leftarrow 0$;
for $i = 1 \rightarrow N$ do
    $D(I_i, I_i) \leftarrow D(I_i, I_i) + I - U_i U_i^T$
end for
```

Step 3. Obtain the low dimensional coordinates.

Make eigen-decomposition of matrix D, compute the 2^{nd} to the $(d+1)^{th}$ smallest eigenvalues and the corresponding eigenvectors $[t_2, \cdots, t_{d+1}]$ which correspond to the optimal low dimensional global coordinates $y_1, ..., y_N$ of each point in original dataset including the new point's low dimensional coordinate y_n.

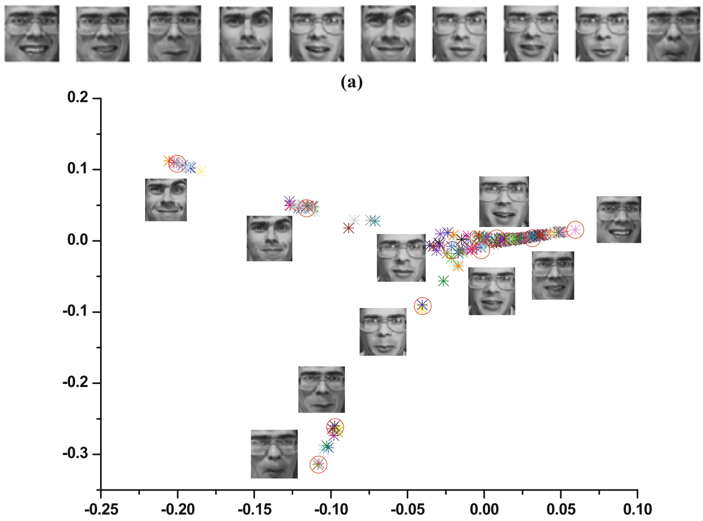
5 Experiments

5.1 Experiments on Olivetti-Faces

In this section we will apply LTSA and the proposed algorithm LITSA to process *Olivetti-faces* dataset[1] which contains 400 face images with 64 × 64 pixels. Some samples are shown in Fig. 1(a). One image of them is randomly selected for testing, while the rest ones serve as training samples. Figure 1(b) shows the dimension reduction results of training samples by LITSA.

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

The horizontal axis represents the face expression changing from unhappy to happy and the vertical axis denotes the face images from without glasses to wearing glasses. The testing images shown in Fig. 1(a) are projected to the low-dimensional space accurately according to their characteristics, because images with similar expressions and characteristics have clustered together. This experiment intuitively shows the effectiveness of LITSA as manifold learning dimension reduction algorithm, which accurately detects and preserves the intrinsic structure of original high dimensional data sets.


(a)

(b)

Fig. 1. (a) Some test images, (b) The unfolding results of the *Olivetti-faces* dataset by LITSA

Dimension reduction is often used as a preprocess step of classification in many machine learning. So dimension reduction always plays an important role in such classification tasks. In this section we display classification effect after dimensional reduction by several incremental algorithms on *Olivetti-faces* dataset, where 40 facial images including 5 images of each individual are randomly selected for testing and the rest ones for training. The experiment is repeated for 10 times. After reducing the dataset to different dimensions, we classify them using k-Nearest Neighbors classifier (k-NN, the nearest neighbors number k is set to be 5). From the classification accuracy rates shown in Fig. 2 we can observe that our algorithm LITSA well detects the intrinsic structure information of the input manifold and achieves better classification precision than other incremental algorithms. It indicates that our algorithm has played a positive role to the subsequent classification learning, has vital significance in many applications such as pattern recognition, image processing and so on.

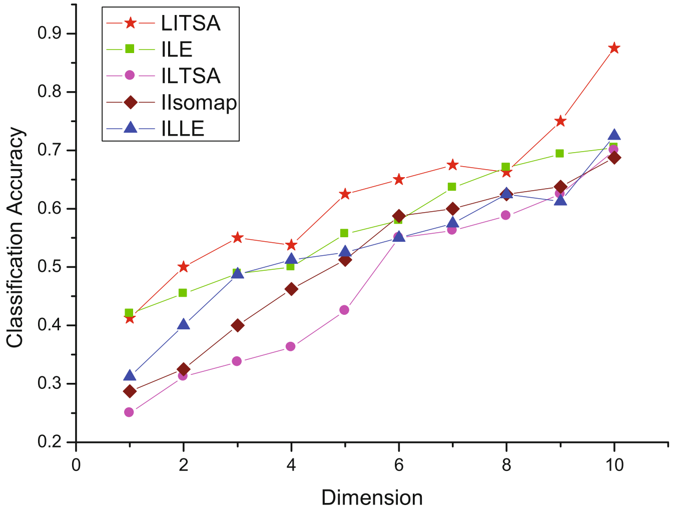


Fig. 2. Accuracy results of kNN classification after dimensional reduction on 10 dimensions

5.2 Local Reconstruction Performance on Rendered Face

In this section we consider the experimental results on Rendered face dataset [1]. This dataset contains 698 facial sculpture images with 64 × 64 pixels. The facial images have two groups of pose parameters which are up to down and left to right.

We compare the reconstruction performance between the proposed algorithm LITSA and other incremental algorithms on Rendered face dataset. To recover these implicit parameters we use matrix P to represent pose parameters. By making affine transformation on the dimensional reduction projection results T by these algorithms mentioned above, we can get the affine transformation expression: $Y = ce^T + LT$. Define the following expression: $||P - Y||_F^2 = \min_{c,L} ||P - (ce^T + LT)||_F^2$. Then the relevant reconstruction error can be measured by: $error = \frac{||P-Y||_F^2}{||P||_F^2}$.

Various reconstruction error computed by IIsomap, ILTSA, ILLE, ILE and LITSA along with neighborhood size k are shown in Table 1, in which LITSA keep the minimum reconstruction error. Reconstruction error of IIsomap after reducing dimensions already exceed the comparison range of other algorithms, which indicates that our method has leading position and superiority among these algorithms with reconstruction error in small area.

To sum up, the algorithm LITSA has showed obvious superiority both in intuitive and mathematical sense, through the experiments of this section on local reconstruction performance and reconstruction error measure.

Table 1. Reconstruction error computed by IIsomap, ILTSA, ILLE, ILE and LITSA

k	$IIsomap$	$ILTSA$	$ILLE$	ILE	$LITSA$
8	0.79	0.0358	0.0088	0.0091	0.0084
9	0.69	0.0352	0.0233	0.0096	0.0087
10	0.58	0.0387	0.0170	0.0096	0.0090
11	0.49	0.0397	0.0230	0.0098	0.0091
12	0.39	0.0381	0.0264	0.0102	0.0093
13	0.26	0.0373	0.0235	0.0105	0.0096
14	0.22	0.0373	0.0191	0.0107	0.0095
15	0.18	0.0385	0.0178	0.0111	0.0096
16	0.19	0.0393	0.0195	0.0116	0.0102
17	0.20	0.0398	0.0210	0.0120	0.0108

6 Conclusion

This paper proposes a new incremental manifold learning algorithm named LITSA based on incremental tangent space construction. Compared with LTSA algorithm and other incremental manifold learning methods, the innovation of this algorithm can be concluded as three aspects: (1) A novel formulation M_{new} is proposed in order to insert the new point's geometric information into existing sample points' covariance matrix. (2) We construct the incremental expression of C_{new} by inserting $(X_{new} - M_{new})$. (3) Inspired by the idea of literature [15], after constructing smaller size inner product matrix $B = AA^T$, we can immediately obtain the local coordinates of dataset inserted with new sample point by computing the eigenvectors and eigenvalues of matrix B. Extensive experiments have been performed in order to evaluate our algorithm's performance both on artificial and actual datasets, demonstrating its ability of dimension reduction effect and recognition accuracy on large scale datasets.

Acknowledgements. This work is supported by National Natural Science Foundation of China under Grants No. 41471371 and the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant No. 15KJB520022.

References

1. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
3. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
4. Zhang, Z.Y., Zha, H.Y.: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM J. Sci. Comput. **26**, 313–338 (2004)

5. Law, M.H.C., Jain, A.K.: Incremental nonlinear dimensionality reduction by manifold learning. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 377–391 (2006)
6. Jia, P., Yin, J., et al.: Incremental Laplacian eigenmaps by preserving adjacent information between data points. Pattern Recogn. Lett. **30**, 1457–1463 (2009)
7. Kouropteva, O., Okun, O., et al.: Incremental locally linear embedding. Pattern Recogn. **38**, 1764–1767 (2005)
8. Abdel-Mannan, O., Ben Hamza, A., et al.: Incremental line tangent space alignment algorithm. In: Proceedings of Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1329–1332. IEEE (2007)
9. Liu, X., Yin, J., Feng, Z., Dong, J.: Incremental manifold learning via tangent space alignment. In: Schwenker, F., Marinai, S. (eds.) ANNPR 2006. LNCS, vol. 4087, pp. 107–121. Springer, Heidelberg (2006). doi:10.1007/11829898_10
10. Zhao, D.F., Yang, L.: Incremental isometric embedding of high-dimensional data using connected neighborhood graphs. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 86–98 (2009)
11. Abdel-Mannan, O., Ben Hamza, A., et al.: Incremental Hessian locally linear embedding algorithm. In: Proceedings of 9th International Symposium on Signal Processing and Its Applications (ISSPA), pp. 1–4. IEEE (2007)
12. Min, W.L., Lu, L., He, X.F.: Locality pursuit embedding. Pattern Recogn. **37**(4), 781–788 (2004)
13. Ling, G, Han, P., et al.: Face recognition via semi-supervised discriminant local analysis. In: Proceedings of International Conference on Signal and Image Processing Applications (ICSIPA), pp. 292–297. IEEE (2015)
14. Li, B., Du, J., Zhang, X.: Feature extraction using maximum nonparametric margin projection. Neurocomputing **188**, 225–232 (2016)
15. Han, P., Yin, O., Ling, G.: Semi-supervised generic descriptor in face recognition. In: Proceedings of International Colloquium on Signal Processing and Its Applications (CSPA), pp. 21–25. IEEE (2015)
16. Xu, Y., Huang, H., et al.: Semi-supervised local fisher discriminant analysis for speaker verification. Adv. Inf. Sci. Serv. Sci. **6**, 1–11 (2014)
17. Artac, M., Jogan, M., Leonardis, A.: Incremental PCA for on-line visual learning and recognition. In: Proceedings of International Conference on Pattern Recognition (ICPR), pp. 781–784. IEEE (2002)
18. Zhao, H., Yuen, P.C., Kwok, J.T.: A novel incremental principal component analysis and its application for face recognition. IEEE Trans. Syst. **36**(4), 873–886 (2006)
19. Li, Y.M.: On incremental and robust subspace learning. Pattern Recogn. **37**(7), 1509–1518 (2004)
20. Weng, J.Y., Zhang, Y.L., Hwang, W.S.: Candid covariance-free incremental principal component analysis. IEEE Trans. Pattern Anal. Mach. Intell. **25**(8), 1034–1040 (2003)
21. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press, Baltimore (2012)