DIMENSIONALITY REDUCTION OF FACE ORDER PROBLEM USING NONLINEAR EMBEDDING METHODS

A PREPRINT

CHENG Haoyi
Department of Mechanical and Aerospace Engineering
Hong Kong University of Science and Technology
Hong Kong, Clear Water Bay, China
hchengaj@connect.ust.hk

QUAN Xueyang
Department of Mathematics
Hong Kong University of Science and Technology
Hong Kong, Clear Water Bay, China
xquan@connect.ust.hk

YU Zhiyuan
Department of Mathematics
Hong Kong University of Science and Technology
Hong Kong, Clear Water Bay, China
zhiyuan.yu@connect.ust.hk

May 21, 2021

ABSTRACT

The objective of this project is to order 33 face images of the same person in different directions. Total six main manifold learning algorithms were employed (including Diffusion Map, MDS, LLE, t-SNE, LTSA and ISOMAP). We also conducted hyper-parameters test to discover the further influence of nearest neighbour number on model performance (changing the number of nearest neighbour from 2 to 32). According to the experiment results and analysis, we discovered that locally linear embedding performed better than others in which number of components and neighbors are 2 and 5, respectively. And LTSA algorithm showed best performance among all these six methods, the first eigenvalue play the most important role on this. Moreover, it was found that when the neighbour number equals to 5, nonlinear algorithm provided the best performance.

Keywords Order face images · Dimensionality reduction · Nonlinear embedding methods · Manifold learning

1 Introduction

Ordering orientation of unstructured images is an interesting and important task. It helps estimate the right motion of object or camera which is essential to many computer vision applications, like object tracking, multi-view reconstruction, semantic understanding and so on. Since images with similar orientations usually share similar contents, this task can be viewed as a similarity measurement and dimensionality reduction problem for image data. In this report, we order 33 face images from the same person with different orientations via 6 manifold learning methods including Diffusion map, MDS, ISOMAP, LLE, LSTA and TSNE. The order results are based on the first eigenvector of image embeddings obtained from these methods.

This paper is organized as the following sections: section 2 introduces the detailed information of the face dataset and ground truth rank of face order. In the section 3, we describe the basic knowledge of six main manifold learning algorithms. Then, we give the detailed results and discussions of this face-order project in section 4. In last section, total conclusions are showed clearly. All the codes in this report are achieved by Python and Scikit-learn Package.

2 Datasets

The face dataset we used in this project contains 33 face images of the same person with different orientations ($X \in \mathbb{R}^{112\times92\times33}$). The dataset can be obtained with the following website:

`https://github.com/yao-lab/yao-lab.github.io/blob/master/data/face.mat`

However, we found that the id number of raw dataset does not match with the face rotation, so we adjust the order of these images with labelled rank number in Figure 1 and Table 1. In this project, we set the ranked image id in Table 1 as the ground truth. The metric we use in this project is one-hot encoding and Frobenius norm which can normalize the value of rank.

Figure 1: The ground truth ordered face images

Table 1: Labelled rank number of ground truth face images

Image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Rank	9	13	19	32	6	18	28	7	17	1	5	16	12	10	4	21	22
Image	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
Rank	26	33	11	2	24	3	27	29	23	14	30	31	20	15	25	8	

3 Methodology

3.1 Diffusion Map (DM)

Diffusion map is a nonlinear dimensionality reduction method based on the underlying geometry structure of dataset. Given high dimensional dataset $X \in \mathbb{R}^{n\times p}$ where each row represents a data point $x_i \in \mathbb{R}^p$ and Gaussian kernel:

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{\alpha}\right) \tag{1}$$

The diffusion matrix P which represents the transition probability of data points is defined as

$$P = D^{-1}K \tag{2}$$

where $K_{ij} = k(x_i, x_j)$, $D = \text{diag}(\sum K)$. Diffusion map re-organizes dataset X according to diffusion matrix by mapping x_i into p_i (row of matrix P). The Euclidean distance between two mapped points p_i and p_j is called diffusion distance. Eigende composition is applied on diffusion matrix P to use less dimensions associated with dominant eigenvectors to approximate diffusion distance.

3.2 Multi-Dimensional Scaling (MDS)

Multi-Dimensional Scaling (MDS) aims to find a subspace that preserves pairwise Euclidean distance of original data points best. Given high dimensional dataset $X \in \mathbb{R}^{n\times p}$ where each row represents a data point $x_i \in \mathbb{R}^p$, its dissimilarity matrix is defined as $D_{ij} = ||x_i - x_j||_2$. MDS can be viewed as an optimization problem as following

$$\underset{p_1,\ldots,p_n}{\operatorname{argmin}} \sum_{i<j} \left(\|p_i - p_j\|_2 - D_{ij}\right)^2 \tag{3}$$

Here $p_1, ..., p_n$ is minimizer in subspace $\mathbb{R}^m, m < p$. To solve this problem, we first apply double centering on D which gives us a new matrix $B = -\frac{1}{2}CD^{(2)}C$. C is the centering matrix. Then we apply eigen-decomposition on B and choose m largest eigenvalues λ_i and corresponding eigenvectors u_i. The minimizer matrix is defined as $P = U\Lambda^{\frac{1}{2}}$, where $U = [u_1, ..., u_m]$, $\Lambda = \text{diag}(\lambda_1, ...\lambda_m)$.

3.3 Isometric Mapping (ISOMAP)

ISOMAP is a nonlinear dimensionality reduction method proposed by B. Tenenbaum et al. (2000), and this algorithm can be regarded as an extension of the multidimensional scaling (MDS). The main idea in this algorithm is based on MDS and preserving the intrinsic geometry of the data, details are stated as the following,

Firstly, constructing a neighboring graph G over all the data. If points i and j are closer than ϵ or i is one of the K nearest neighbors of j, then the two points are connected in graph G with distance (edges of weight) $d_X(i, j)$.

Secondly, computing the shortest paths distance $d_G(i, j)$ in G to approximate the geodesic distance $d_M(i, j)$ on manifold M. Initially, if i, j are in neighbor, let $d_G(i, j) = d_X(i, j)$. There are two simple algorithms for calculating $d_G(i, j)$, Floyd's Algorithm and Dijkstra's Algorithm. The key idea is that for the neighboring points, (since the neighboring two points are almost in the same tangent space,) the Euclidean distance is used to represent the geodesic distance. For the faraway points, the shortest paths distance is used in the neighboring graph to approximate the geodesic distance.

Thirdly, constructing a lower dimensional embedding. Apply the classical MDS to distance matrix $D_G = \{d_G(i, j)\}$.

3.4 Locally Linear Embedding (LLE)

ISOMAP introduced above is a global method, which is very expensive. And since the manifold is locally like the Euclindean space, one smart way is choosing to fit locally and aligning globally. Based on this idea, T. Roweis and K. Sau (2000) proposed an algorithm, Locally Linear Embedding (LLE), which uses locally linear fits to recover global nonlinear structure. Thus, the whole algorithm contains two parts, one is **local fit**, the other is **global alignment**.

For the **local fit** part, firstly, construct a neighborhood graph $G = (V, E)$ such that $V = \{x_i : i = 1, ..., n\}$ and $E = \{(i, j) : j \in N_i\}$ (for instance, ϵ-neighbors or k-nearest neighbors). The next step is to do the local fit. That is, for arbitrary x_i and N_i, minimize the following embedding cost function,

$$\min \left\| x_i - \sum_{j\in N_i} w_{ij} x_j \right\|^2 \tag{4}$$

$$\text{s.t.} \sum_{j\in N_i} w_{ij} = 1 \tag{5}$$

(The Lagrange multiplier method can be applied to solve this optimization problem.)
For the **global alignment** part, a weight matrix $W \in \mathbb{R}^{n\times n}$ is defined as,

$$W_{ij} = \begin{cases} w_{ij}, & j \in N_i \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Then minimize the following global embedding cost function to obtain the embedding coordinates $Y \in \mathbb{R}^{d\times n}$,

$$\min_Y \sum_i \left\| y_i - \sum_{j=1}^{n} W_{ij} y_j \right\|^2 \tag{7}$$

Denote the kernel matrix K as $(I-W)^\top(I-W)$.

Find the $(d+1)$ smallest eigenvectors of K and their corresponding eigenvalues. And set the d-th row of Y to be the $(d+1)$ smallest eigenvector.

3.5 Local Tangent Space Alignment (LTSA)

LLE focuses on preserving the neighborhood distances, however, for Local tangent space alignment method (LTSA), it aims at characterizing the local geometry for every neighborhood in its tangent space. And then solve a global optimization problem to align these tangent spaces, and then achieve learning the embedding.

So here are some key steps in this algorithm, given these information, firstly search the nearest neighbors, then align the tangent space to obtain kernel matrix K, finally do the eigenvalue decomposition on K. Find the smallest (d+1) eigenvectors and drop the smallest one. Obtain a d-embedding from the remaining d smallest eigenvectors.

Firstly, search the nearest neighbors. For a set of data $x_1, x_2, ..., x_n \in \mathbb{R}^p$, compute the k-nearest neighbours of x_i and compute local SVD on neighbourhood of x_i.

$$X^{(i)} = [x_{i_1} - \mu_i, ..., x_{i_k} - \mu_i]^{p\times k} \quad x_{i_j} \in N(x_i) \quad \mu_i = \sum_{j=1}^{k} x_{i_j} \tag{8}$$

$$X^{(i)} = U^{(i)} \sum (V^{(i)})^\top \tag{9}$$

Then, align the tangent space. Obtain the kernel matrix.

$$K^{n\times n} = \sum_{i=1}^{n} S_i W_i W_i^\top S_i^\top \quad W_i^{k\times k} = I - G_i G_i^\top \tag{10}$$

$$[x_1, x_2, ..., x_n] S_i^{n\times k} = [x_{i_1}, ..., x_{i_k}] \quad G_i = [1/\sqrt{k}, V_1^{(i)}, ..., V_d^{(i)}]^{k\times(d+1)} \tag{11}$$

Finally, do eigenvalue decomposition on kernel matrix $K = U\lambda U^\top$. Find smallest (d+1) eigenvectors and drop the smallest one. Obtain a d-embedding from the remaining d smallest eigenvectors.

3.6 t-distributed Stochastic Neighbor Embedding (t-SNE)

T-distributed stochastic neighbor embedding (t-SNE) is a statistical method for visualizing high-dimensional data by giving each data point a location in a two or three-dimensional map. It is based on Stochastic Neighbor Embedding originally developed by Sam Roweis and Geoffrey Hinton, where Laurens van der Maaten proposed the t-distributed variant. It is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability.

Once we convert a set of high-dimensional data into a matrix of pairwise similarities, t-distributed Stochastic Neighbor Embedding (t-SNE) is used to visualize the resulting similarity data. t-SNE can capture the majority of local structures for high-dimensional data, meanwhile, it also reveals global structure well.

The t-SNE algorithm comprises two main stages. First, t-SNE constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects are assigned a higher probability while dissimilar points are assigned a lower probability. Second, t-SNE defines a similar probability distribution over the points in the low-dimensional map, and it minimizes the Kullback–Leibler divergence (KL divergence) between the two distributions with respect to the locations of the points in the map. While the original algorithm uses the Euclidean distance between objects as the base of its similarity metric, this can be changed as appropriate.

The key steps in this algorithm are list as following: 1) Compute pairwise affinities with perplexity, 2) Sample initial solution, 3) Compute low-dimensional affinities and gradient

4 Results and Discussions

4.1 Performance Evaluation

This project applied six manifold learning algorithms to order these 33 face images, and obtained the rank results which is presented in the Table 2 as following. We defined the face image order is from left to right (we give the number of nearest neighbour as 5 in these conditions).

Table 2: The rank results of various manifold learning methods

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Ground Truth	10	21	23	15	11	5	8	33	1	14	20	13	2	27	31	12	9
DM	10	21	23	15	11	5	8	1	33	14	20	13	2	27	31	12	9
MDS	10	21	23	19	14	20	5	11	33	8	4	13	29	1	28	15	25
ISOMAP	10	21	23	15	5	11	1	33	14	8	20	13	2	27	31	12	9
LLE	10	21	23	15	5	11	8	1	33	14	20	13	2	27	31	12	9
LTSA	10	21	23	15	11	5	33	8	1	14	20	13	2	31	27	12	9
t-SNE	21	10	3	31	26	23	15	30	6	27	22	13	17	9	1	14	16

Rank	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Ground Truth	6	3	30	16	17	26	22	32	18	24	7	25	28	29	4	19
DM	6	3	30	16	17	22	26	32	18	24	7	25	29	28	4	19
MDS	27	2	7	24	9	18	31	12	32	6	17	26	30	16	3	22
ISOMAP	6	3	30	16	17	26	22	32	18	24	7	25	28	29	4	19
LLE	6	3	30	16	17	22	26	32	18	24	7	25	28	29	4	19
LTSA	6	3	30	16	17	26	22	32	18	24	7	25	25	29	4	19
t-SNE	18	32	5	12	33	28	2	25	7	8	11	29	24	20	19	4

In order to evaluate the performances for face-order project of these six manifold learning algorithms, we additionally proposed the metrics to calculate the rank position differences of the same face image in dataset. As we see, the absolute error seems not the good method to evaluate the accuracy of these algorithms with discrete data, so the one-hot matrix is applied in this paper to evaluate the differences between our results and ground (showed in Table 3). From Table.3,

Table 3: One-hot error of different algorithms

Algorithm	Diffusion Map	MDS	ISOMAP	LLE	LTSA	t-SNE
Error	3.46410	7.74560	3.16228	3.46410	2.82843	7.87401

we can find that LTSA algorithm obtains the best accuracy among all these six methods, with one-hot error–2.82843, which represents LTSA performs better in this face-order project. And moreover, the one-hot error of MDS and t-SNE is 7.74560 and 7.87401, respectively. They are much larger than any other algorithms, seems that they cannot achieve this task very well.According to other former studies, it is obvious that LTSA actually performs best, and followed by ISOMAP, LLE Diffusion Map methods. Without suspense, MDS and t-SNE show worst performance in all similar tasks. Thus, maybe LTSA or LLE are fit for image-order project.

4.2 Visualization results

In this subsection, we give the visualization results of our experiments and compare the intuitive effect of different algorithms, which enable to provide detailed information of each performance.

4.2.1 Diffusion Map (DM)

Figure.2 represents the 2D embedding graph results of Diffusion Map algorithm with first two eigenvectors. The scatter plots of DM shows a "V" shape and the image of front face is nearly at the middle position which intuitively reveals the good order performance of Diffusion Map Method.

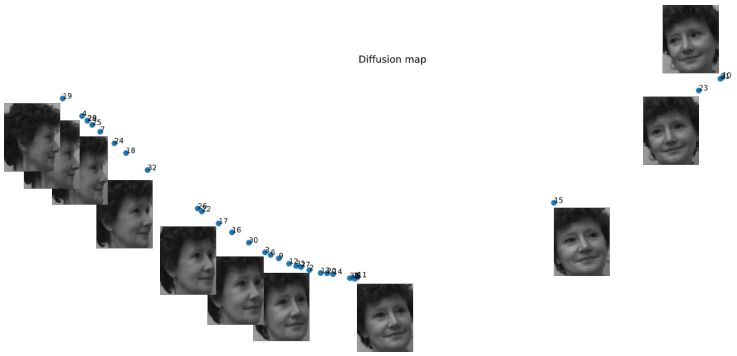


Figure 2: 2D embedding graph by Diffusion Map

4.2.2 MDS and ISOMAP

Both MDS and ISOMAP methods are the global, iterative and distance preserved algorithm. So we put them togather and compare the visualization results of them on following section. Figure 3 gives the 2D embedding graphs by MDS and ISOMAP. First 2 eigenvector is applied in these two algorithms. We can discover that although MDS extract the hair feature and face feature (first 2 components) exactly, meaning face gradually turn from left to right direction, it is unable to correctly obtain the distance among different principal components. Some images containing the similar features are divided into different subgroups, so there is no doubt that MDS shows larger error and order the face images in poor performance. On the contrary, ISOMAP not only extracts first two principal features perfectly as MDS, but also discover the potential/relative features behind principals to sort face images in detail. As for ISOMAP in Figure 3, it extract "face feature (face area)" to achieve classifying images from left to right (seems three regions), and ISOMAP then utilize "hair pattern" to delicately sort the images in "middle region". Thus, ISOMAP represents a really better results.

4.2.3 LLE and LSTA

We give the 2D embedding graph by LLE and LTSA in Figure 4 and the ordered image in Figure 5. Compare Figure 4(a) and Figure 2(a), we can find that although scatter plot of DM shows "V" shape and LLE shows different "A" shape, both two algorithms obtain the same performance in this task (one-hot error is totally the same). Thus, LLE is also the same as DM, which classify the first two components successfully and get the same discrepancy between various points. To achieve this task, DM and LLE give the totally same process to 33 face images behind the surface.

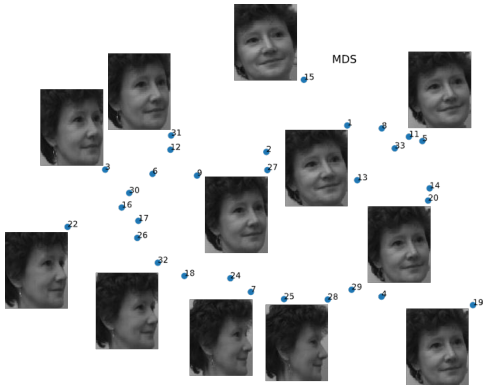
LLE and LTSA are both the algorithm which preserve the local properties. Therefore, we also compare these two nonlinear methods. It seems that LTSA first extract "face area" as first principal component to divided images into two subgroups. Then it applied "hair pattern" as second components (see in Figure 4(b) "right subgroup") to obtain the detailed order number of face images. Take ISOMAP into consideration, we discovered that ISOMAP roughly classifies 33 face images into three regions under first principal components, but LTSA classifies them into two. With this reason, LTSA represents better performance than ISOMAP, and also be the best among all. Above all, we can conclude that the better result the first principal extract, the better performance the nonlinear embedding method obtain.

4.2.4 t-SNE

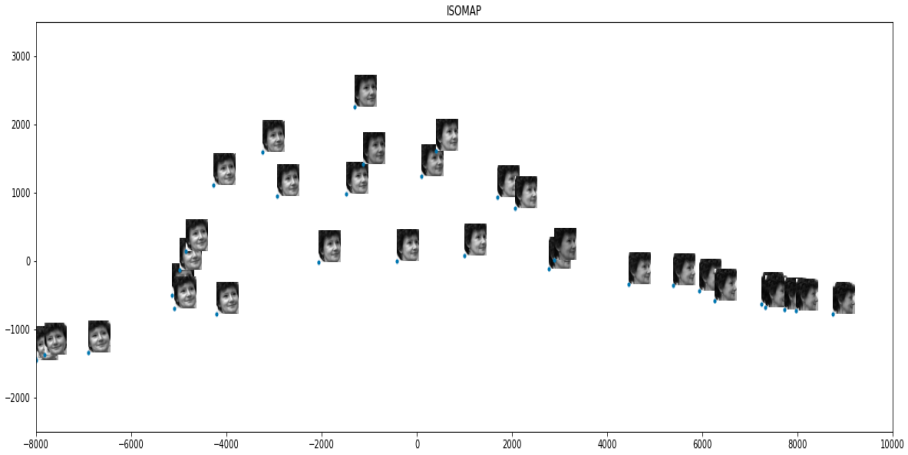
Compare Figure 6 with Figure 3(a), the 2D embedding graphs of MDS and t-SNE show the similar situation. They both sort 33 face images from left to right in general, but with no capability to extract first eigenvector with great effect, which is totally different from LTSA and DM. Thus, it reveal one significant conclusion again, obtaining the first eigenvector with perfect performance can lead to a smaller error of algorithm and better results in images order.

4.2.5 Hyper-parameter study

In order to further evaluate the influence of parameters on algorithm performance, we change the number of nearest neighbour(K) from 2 to 32, in a broadband. The one-hot error results show in Figure 7. It reveals that for all these three nonlinear methods, when the number of nearest neighbour changing from 2 to 32, the errors are first reduce to the lowest level and then continue to increase to the end. If the K is equal to 5, these three algorithms all obtain the best performance in this image-order task with 33 face images.

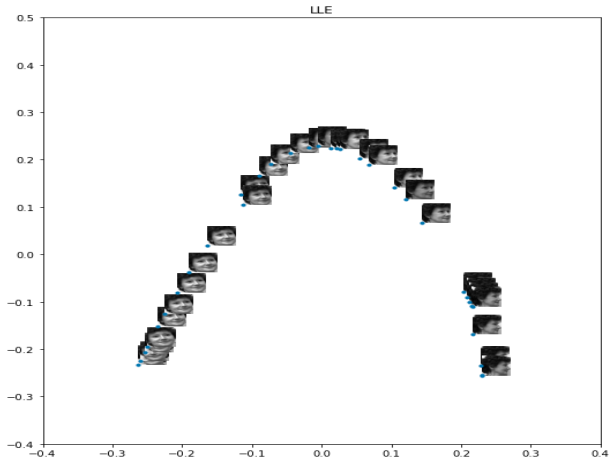


(a) MDS

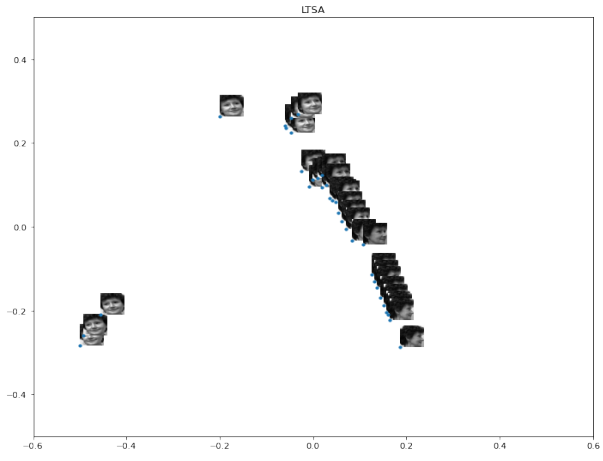


(b) ISOMAP

Figure 3: 2D embedding graph by MDS and ISOMAP



(a) LLE



(b) LTSA

Figure 4: 2D embedding graph by LLE and LTSA

(a) LLE (b) LTSA

Figure 5: Ordered face images with LLE and LTSA

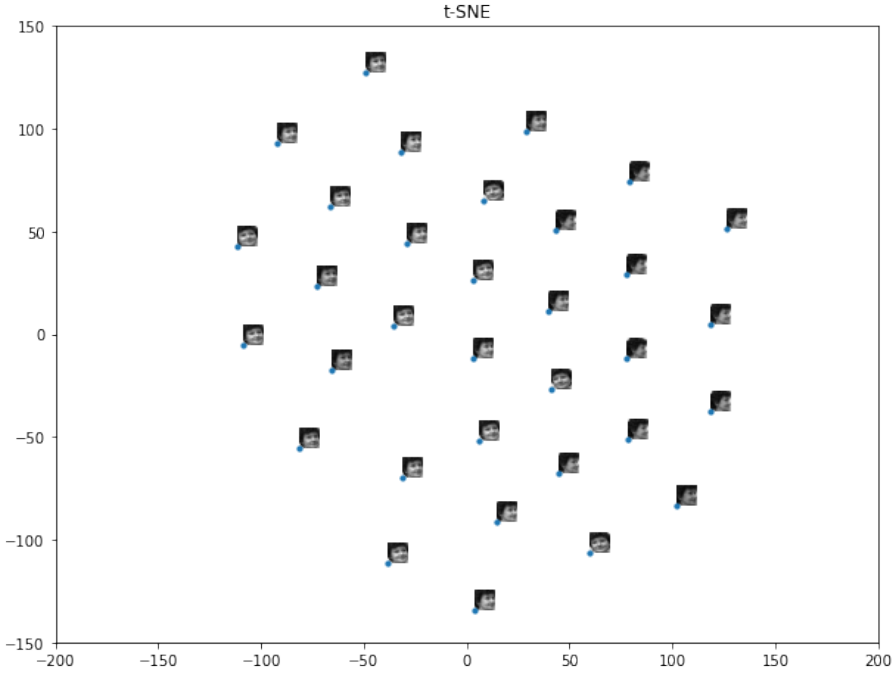


Figure 6: 2D embedding graph by t-SNE

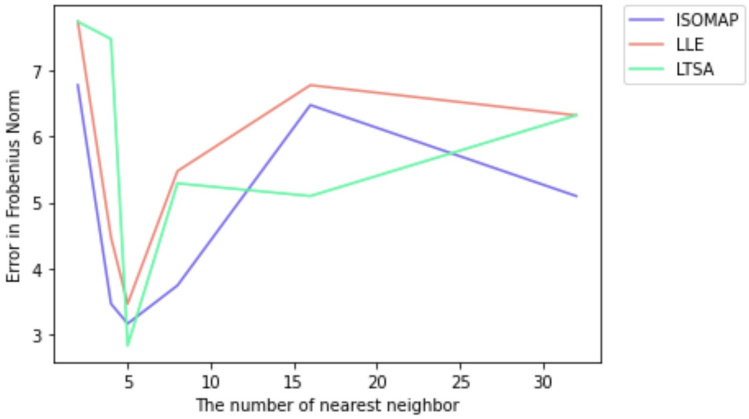


Figure 7: One-hot error changes with various nearest neighbour numbers

4.3 Conclusion

In summary, from this image-order project, we obtained several main conclusions as following: First, except t-SNE and MDS methods, the other four nonlinear embedding methods enable to perform greatly in 33 face image order. Second, the successful algorithms all show the same characteristics – they possess the capacity to extract perfect fist eigenvector. The better results of first eigenvector, the smaller the error is, and then lead to more perfect image-order performance. Last, for the specific task, the number of nearest neighbour has best value which can result in smallest error. In this 33 images order task, when the number of nearest neighbour equals to 5, the nonlinear algorithms get best performance, which is totally true for all the methods (ISOMAP, LLE and LTSA).

From this project, we obtained several main factors that show great influence on the performance of nonlinear embedding methods. Therefore, in the future, the most interesting challenge for us is to design a novel method which can achieve broad tasks with good performance, such as move object detection, emotion detection and so on.

References

[1] Joshua B. Tenenbaum, Vin de Silva and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. In *Science 22 Dec 2000: Vol. 290, Issue 5500*, pages 2319–2323.

[2] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. In *Science 22 Dec 2000: Vol. 290, Issue 5500*, pages 2323–2326.

[3] Manifold learning. Retrieved from: https://scikit-learn.org/stable/modules/manifold.html

[4] PyDiffMap. Retrieved from: https://pydiffmap.readthedocs.io/en/master/readme.html

[5] zhenyue Zhang, Hongyuan Zha. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. In *SIAM Journal on Scientific Computing 2004: Vol.26*, pages 313-338.

[6] Van der Maaten, L.J.P., Hinton.G.E. Visualizing Data Using t-SNE. In *Journal of Machine Learning Research Nov 2008: Vol.9*, pages 2579-2608.