Dimensional reduction methods for face dataset (CSIC 5011 Final project)

ZHONG Guangzheng
Hong Kong Baptist University
22482032@life.hkbu.edu.hk

Abstract

In this final project, we explore the dimensional reduction methods on face dataset. The dataset contain 33 faces of the same person in different angles. We perform diffusion map, MDS, isomap, LLE, LTSA and t-SNE on this dataset, and visualize the results.

The code can be found in : https://github.com/zhong-gz/CSIC-5011-Final-project.

The presentation video can be found in : https://www.youtube.com/watch?v=DrR2wcYPccc.

I. Introduction

Dimensional reduction are proposed for high dimensional data. One of the problems in high dimensional datasets is that, not all features is important for particular task. Although some of the methods with high computational complexity shows good performance in prediction or classification problems [1], there is still meaningful to reduce the dimension of data, which may obtain models with better explain-ability.

There are two main applications of dimensional reduction. The first is feature extraction [2]. Due to the increasing dimension of data, an effective technique is required for all data mining and machine learning tasks. To improve the performance of machine learning methods and reduce the training time, dimensional reduction is considered as a pre-processing. The second is visualization [3]. Good visualizations help to get better output. It also explains the distribution of data, and the working principle of machine learning methods.

In this project, we investigate face dataset with 6 dimensional reduction methods. The dataset contain 33 faces of the same person in different angles. The resulting mapped data are visualized and analyzed.

This project report is organized as follows. In Section II, III, IV, V, VI and VII, we investigate diffusion map, MDS, isomap, LLE, LTSA and t-SNE. The conclusion and future work are presented in section VIII.

II. Diffusion map

We explore the diffusion map [4] on the face dataset. Particularly, we construct the weight of map W_{ij} as follows

$$W_{ij} = e^{\frac{-\|x_i - x_j\|^2}{t}}. \tag{1}$$

Then we define the degree D as

$$D = diag(\sum_j W_{ij}). \tag{2}$$

And we obtain the diffusion matrix L by

$$L = D^{-1}W - I. \tag{3}$$

The largest eigenvalue of the diffusion matrix L is 0, i.e., $\lambda_0 = 0$. We consider the second-largest eigenvalue λ_1, and its corresponding eigenvector v_1. We sort the faces by the values of $v_1(i), i = 1, \cdots, n$. The results with different value of parameter t are shown in fig. 1.

(a) $t = 1$

(b) $t = 10$

(c) $t = 100$

Fig. 1: Diffusion Map on Face dataset with different parameter value t.

It can be found that, the result of $t = 1$ is not satisfactory. The fourth image from right is not similar to its neighboring image. It may due to that, the dimension of data x is large ($d = 10304$). Although two image is similar, the value of $\|x_i - x_j\|^2$ is still large. As a result, when t is small, the weight $\mathbf{W}$ is small between two similar data. As the parameter t increases, $e^{\frac{-\|x_i - x_j\|^2}{t}}$ decreases. Therefore, the weight $\mathbf{W}$ can describe the difference between two data appropriately for large t on high dimensional dataset.

III. Multidimensional scaling (MDS)

Visualizing the degree of similarity between specific dataset cases is done through multidimensional scaling (MDS) [5]. Using MDS, the pairwise distances among a set of n data can be mapped into a low dimensional space.

In this project, we consider the metric multidimensional scaling (mMDS). The optimization process is generalized to a number of loss functions and input matrices of known distances with weights and other factors. The loss function, which is a residual sum of squares. Specifically, we optimize the following optimization problem.

$$\min_{x_1, x_2, \cdots, x_n} \sqrt{\sum_{i \neq j = 1, \ldots, N} \left(d_{ij} - \|x_i - x_j\|\right)^2} \tag{4}$$

where d_{ij} is the distance between the i^{th} data and the j^{th} data in original space. In this project, we apply the Euclidean distance, i.e., $d_{ij} = \|z_i - z_j\|^2$, where z_i is the original data.

The MDS mapped data $x_i, i = 1, \cdots, n$ are visualized in fig. 2.

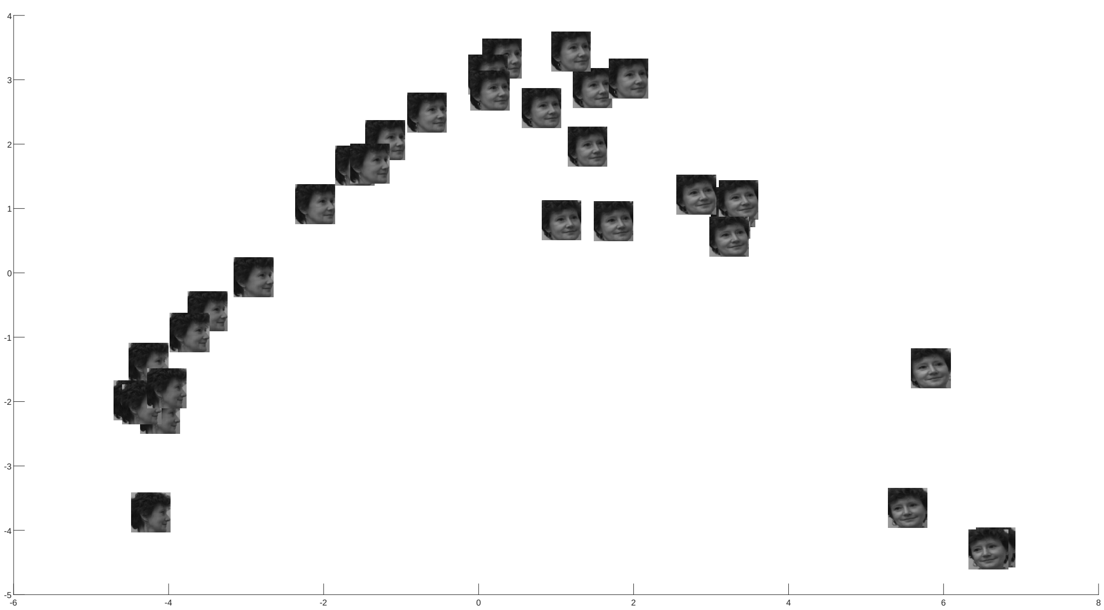


Fig. 2: MDS on Face dataset.

It is clear that, images with similar angle are located near to each other. The mapped data preserves the distance in original space. The images that the person looking to the right is far from that of looking ahead.

IV. Isomap

Isomap [6] is one example of an isometric mapping method. It extends metric multidimensional scaling (MDS) by taking into account the geodesic distances imposed by a weighted graph. Isomap differs from other scaling methods because it incorporates the geodesic distance generated by a neighborhood graph. This is done so that the resulting embedding will include manifold structure. According to isomap, the geodesic distance is the total of the edge weights along the shortest path between any two nodes, e.g., by Dijkstra's algorithm [7].

The isomap is composed with 4 steps.

1) The neighbors of each point are determined.
2) A neighborhood graph is constructed by considering K nearest neighbor.
3) The shortest path between any two data is calculated.
4) MDS method is performed, where the distance d_{ij} in (4) is the shortest path between the i^{th} data and the j^{th} data.

The isomap mapped data with $K = 5$ is visulized in fig. 3.

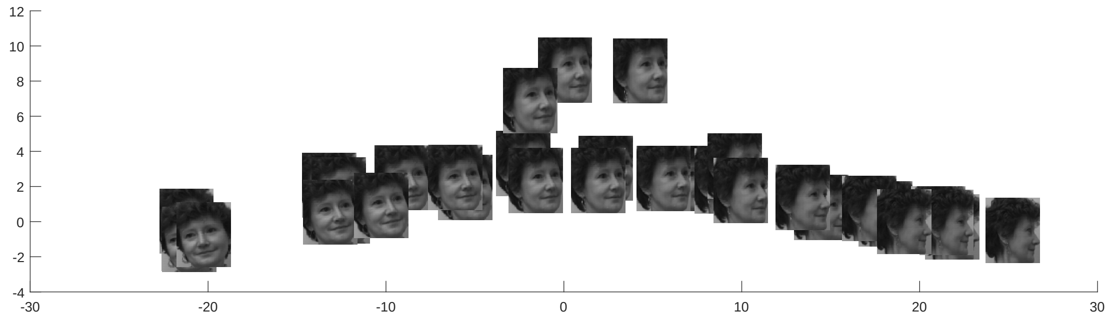


Fig. 3: Isomap on Face dataset with $K = 5$.

Similar with MDS (fig. 2), the images with similar angle is near to each other, as the shortest neighbor path distance is relatively small. The images on the left-hand side and that on the right-hand side are the angles that are totally diffidence, i.e., face ahead and face to right.

Comparing with MDS (fig. 2), the isomap method results in a denser mapped data. It may due to that the distance it considered is the shortest neighbor graph path instead of the Euclidean distance. The shortest neighbor graph path can be considered as a kernel matrix. The isomap method is proposed for manifold data. The resulting data shows that the face dataset could be a dataset with manifold structure.

V. Locally-linear Embedding (LLE)

In comparison to isomap, locally-linear embedding (LLE) [8] offers a number of benefits, such as faster optimization when using sparse matrix algorithms and superior outcomes for a variety of issues. Finding each point's closest neighbors is the main idea of LLE, which is the same as isomap. Then, for each point, a set of weights that best describes the point as a linear combination of its neighbors is computed. The low-dimensional embedding of the points is then discovered using an eigenvector-based optimization technique, ensuring that each point is still described by the same linear combination of its neighbors. LLE typically does not handle non-uniform sample densities well because there is no fixed unit to prevent the weights from drifting as different regions have different sample densities.

The LLE mapped data with $K = 5$ is visulized in fig. 4.

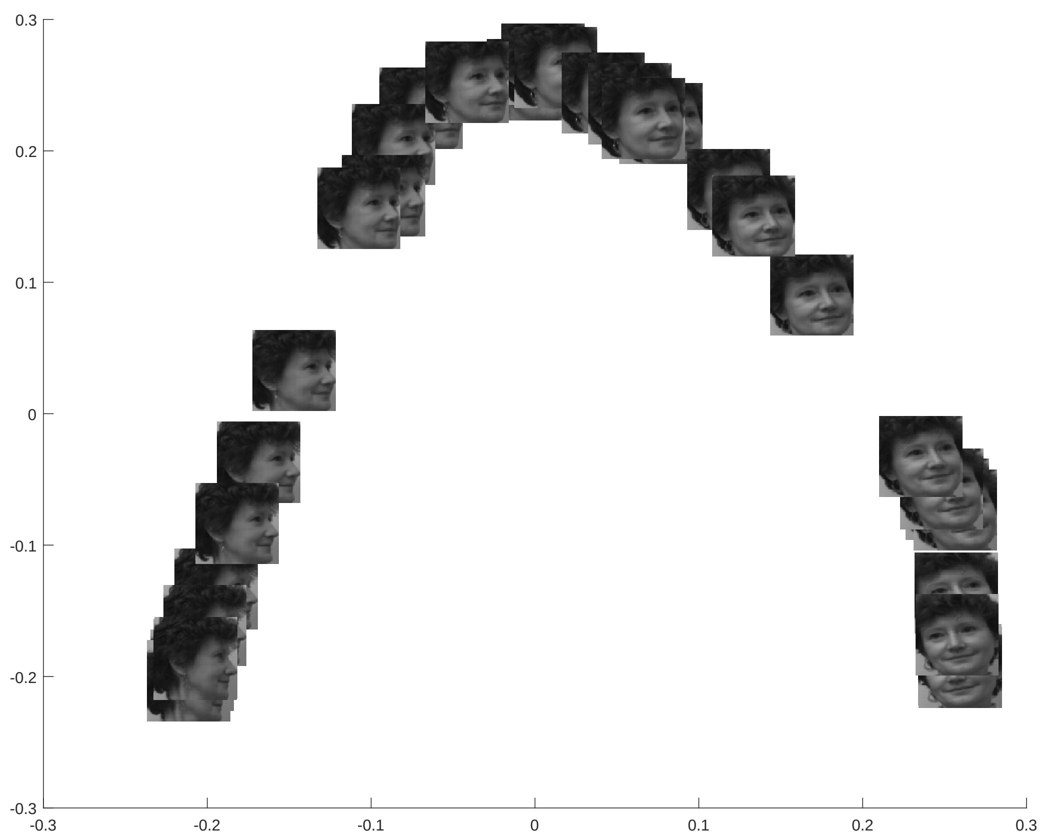


Fig. 4: LLE on Face dataset with $K = 5$.

It can be seen that, the similar images are close to each other in LTSA (fig. 4). However, compared with isomap (fig. 3), two different images are not far from each other enough (e.g., images with an ordinate about -0.2), while these images are located on the left and right ends of isomap.

The main difference between isomap and LLE is that isomap uses geodesic distances to preserve the intrinsic geometry of the data, while LLE uses linear mappings to preserve the local relationships between the data points. In general, isomap is better suited for preserving global structure, while LLE is better suited for preserving local structure.

VI. Local tangent space alignment (LTSA)

Local tangent space alignment (LTSA) [9] is a manifold learning method that can efficiently learn a nonlinear embedding into low-dimensional coordinates from high-dimensional data. It is based on the idea that when a manifold is correctly unfolded, all the tangent hyperplanes to the manifold will become aligned. The k-nearest neighbors of every point is considered and the tangent space at each point is computed by the d-first principal components in each local neighborhood. Furthermore, it then optimizes to find an embedding that aligns the tangent spaces.

LTSA method can be broken down into the following steps:

1) Compute the k-nearest neighbors for each point in the high-dimensional data.
2) For each point, compute the local tangent space by computing the d-first principal components of its k-nearest neighbors.
3) Compute the weight matrix W, where W_{ij} is the weight assigned to the edge between points i and j. This weight is proportional to the squared distance between the points in the tangent space.
4) Compute the graph Laplacian $L = D - W$, where D is the diagonal matrix of row sums of W.
5) Compute the d-first eigenvectors of L, excluding the first eigenvector (which corresponds to the trivial constant solution). These eigenvectors form the new low-dimensional representation of the data.

The optimization objective is to find the low-dimensional embedding that aligns the tangent spaces, which is achieved by minimizing the cost function:

$$\min_{x_i} \sum_{ij} W_{ij} \|x_i - x_j\|^2 \tag{5}$$

where x_i is the mapped data in low-dimensional space. This cost function encourages nearby points in the high-dimensional space to be nearby in the low-dimensional space, while also ensuring that the tangent spaces are aligned.

The LTSA mapped data with $K = 5$ is visulized in fig. 5.

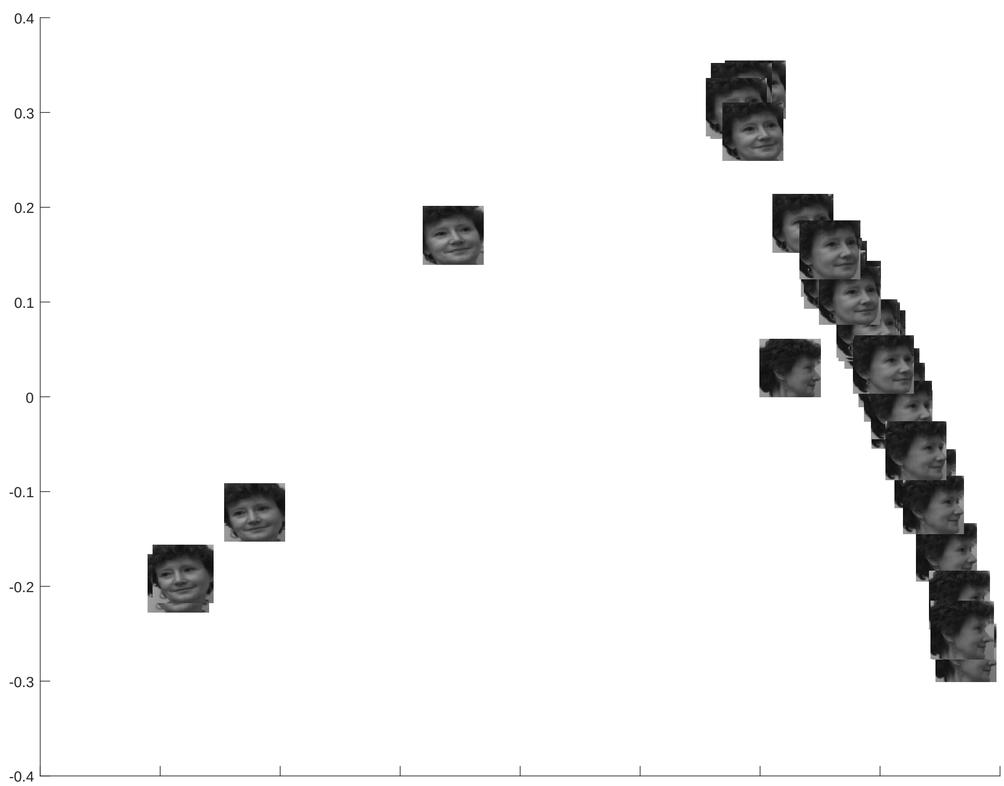


Fig. 5: LTSA on Face dataset with $K = 5$.

It can be seen that, the similar images are close to each other in LTSA (fig. 5). However, compared with isomap (fig. 3), 3 images that facing to the right are far from the other images, which is reasonable for human instinct that facing to right differs to those facing to left.

LTSA and isomap are both nonlinear dimensionality reduction techniques that aim to preserve the underlying geometric structure of high-dimensional data in a lower-dimensional space. However, there are some differences between the two methods.

One of the main differences is the definition of neighborhood relationships. Isomap considers a fixed radius to define the neighborhood of each data point, while LTSA applies a variable radius based on the local density of the data. This can lead to LTSA being more robust to variations in the density of the data.

Another difference is the approach to preserve the geometry of the data. Isomap tries to preserve the pairwise geodesic distances between all pairs of data points, while LTSA tries to preserve the distances between nearest neighbors. This can lead to LTSA being better suited for preserving the local structure of the data.

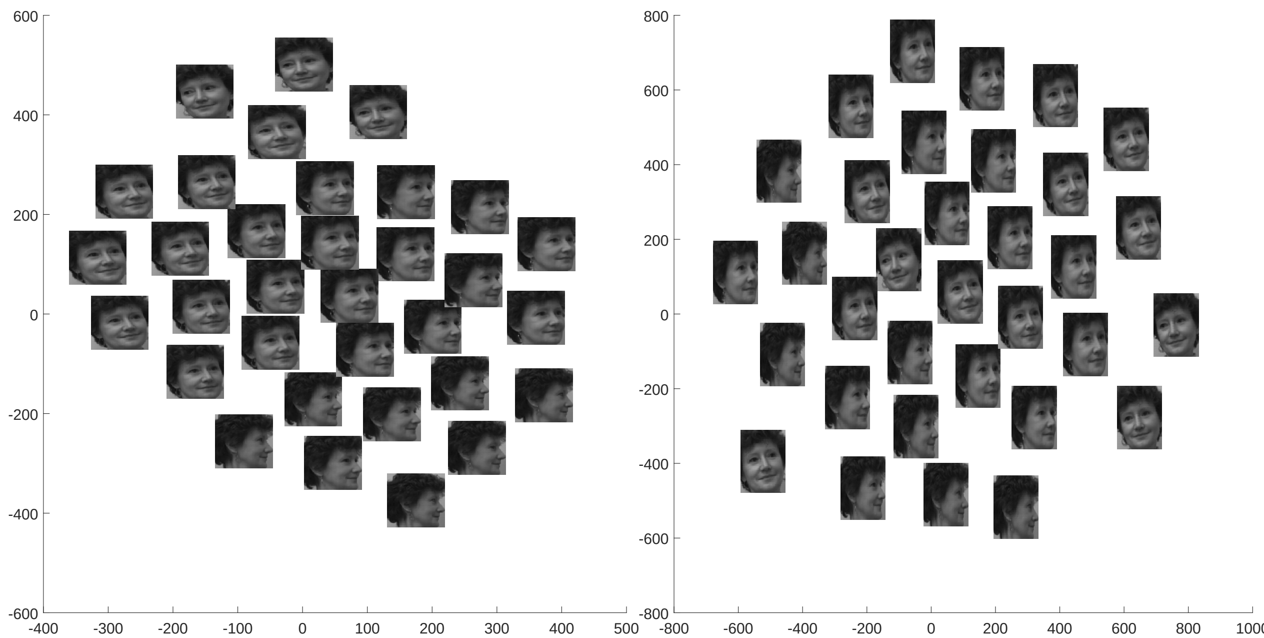
Finally, the loss function of LTSA is different from that of isomap. LTSA minimizes the squared difference between the distances of nearest neighbors in the high-dimensional space and the corresponding distances in the low-dimensional space. Isomap tries to minimize the sum of the squared differences between the pairwise geodesic distances in the high-dimensional space and the corresponding distances in the low-dimensional space.

VII. 2-D t-SNE

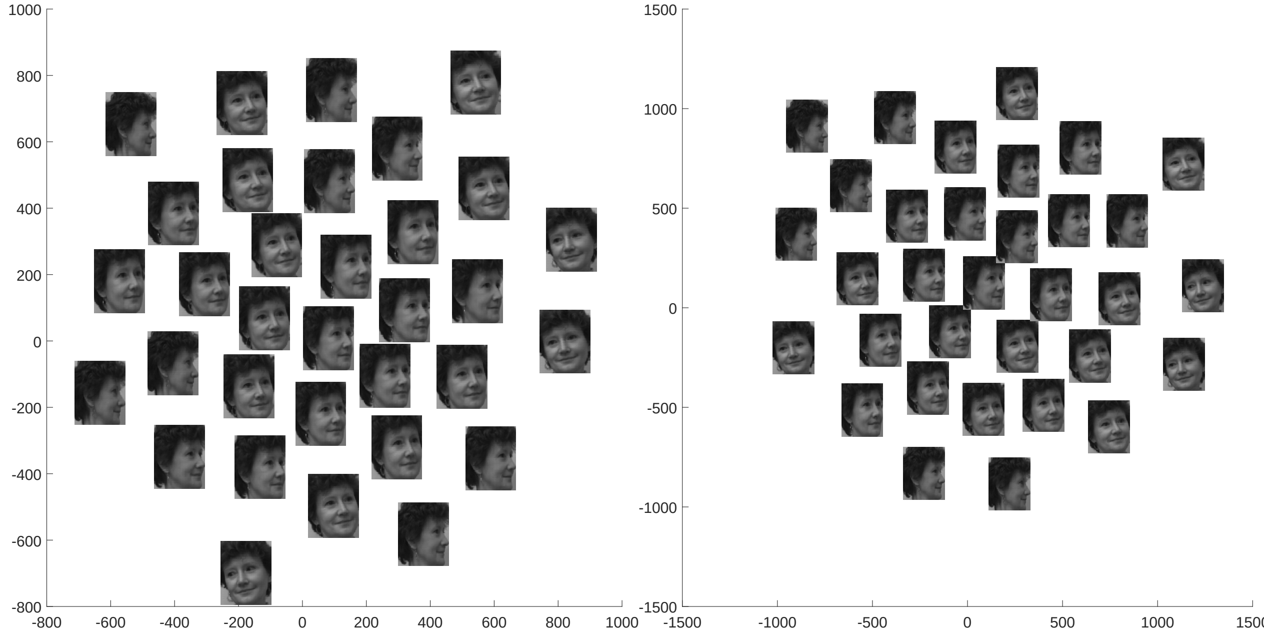
The t-distributed stochastic neighbor embedding (t-SNE) [10] is a non-linear dimensionality reduction technique that maps high-dimensional data to low-dimensional space for visualization. The main idea of t-SNE is to preserve the proximity relationships between data points in high-dimensional space and to try to preserve those relationships as much as possible in the low-dimensional space. It uses a probability distribution to represent the position of each data point in both high-dimensional and low-dimensional space, and then minimizes the KL divergence between these two distributions. t-SNE can be used for clustering, classification, and visualizing high-dimensional datasets.

In t-SNE, the high-dimensional data is represented by a Gaussian distribution, and the low-dimensional data is represented by a Student's t-distribution. The Gaussian distribution is used to calculate the similarity between data points in the high-dimensional space, while the student's t-distribution is used to calculate the similarity between data points in the low-dimensional space. The application of the student's t-distribution in the low-dimensional space helps to prevent the crowding problem, which is a common issue in other dimensionality reduction techniques such as PCA.

The 4 sets of t-SNE mapped data with $perplexity = 30$ are visualized in fig. 6.



(a) (b)



(c) (d)

Fig. 6: t-SNE on Face dataset.

It can be seen that, the results with same setting differ from each other. This may due to that the t-SNE algorithm is a stochastic algorithm, which means that the results can vary each time it is run on the same data with the same parameter settings. The t-SNE uses a random initialization of the embedding, and the optimization process involves a random walk through the high-dimensional space. As a result, the t-SNE can converge to different local minima depending on the starting point and the path taken during optimization. Therefore, running the algorithm multiple times with different random seeds may get a better understanding of the underlying structure of the data.

The main differences between t-SNE and other dimensionality reduction methods, such as PCA and isomap, is that t-SNE uses a probabilistic approach to modeling the similarity between points, whereas other methods typically use a deterministic approach. This allows t-SNE to better handle complex, nonlinear relationships between points in the data.

Finally, t-SNE is often used for visualization purposes, whereas other methods may be used for a wider range of tasks, such as feature extraction or data compression.

VIII. Conclusion

In this project, we investigate face dataset with dimensional reduction methods, including diffusion map, MDS, isomap, LLE, LTSA and t-SNE. The resulting mapped data are visualized and compared with each other. The analysis of each method is also presented.

The dimensional reduction methods that we investigate can be only conducted on existing data. In the future, a mapping function for image dimensional reduction could be considered. With a mapping function, the new data (or testing data) can be mapped into low dimension space and be classified by appropriate machine learning methods.

References

[1] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[2] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[3] S. Kaski and J. Peltonen, "Dimensionality reduction for data visualization [applications corner]," *IEEE signal processing magazine*, vol. 28, no. 2, pp. 100–104, 2011.

[4] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the national academy of sciences*, vol. 102, no. 21, pp. 7426–7431, 2005.

[5] A. Mead, "Review of the development of multidimensional scaling methods," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 41, no. 1, pp. 27–39, 1992.

[6] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[7] E. DIJKSTRA, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[8] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[9] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM journal on scientific computing*, vol. 26, no. 1, pp. 313–338, 2004.

[10] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.