Manifold Learning Comparison for Face Ordering

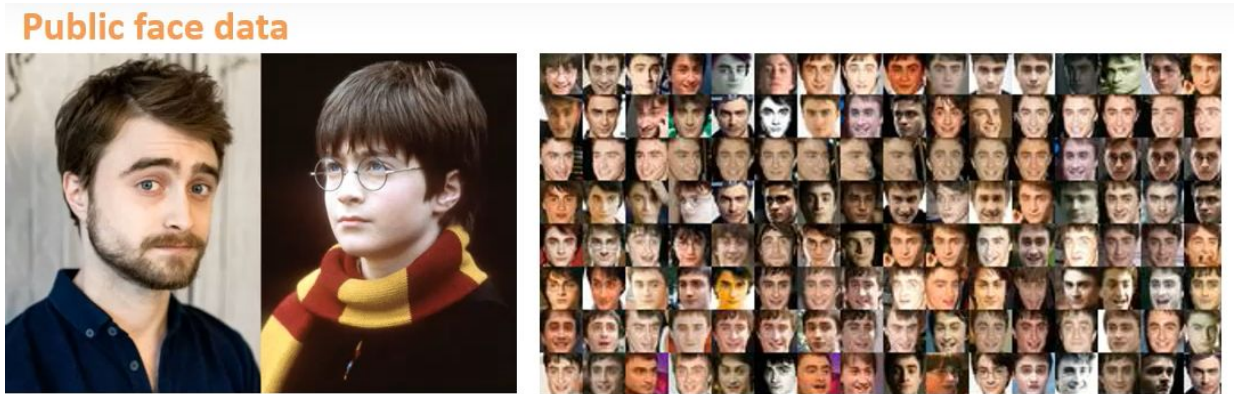
Ng Yui Hong(20434594)
CSIC5011 pj2 present

Outline

- Introduction
- Dataset
- Methods
- Performance Evaluation
- Conclusion and Further work

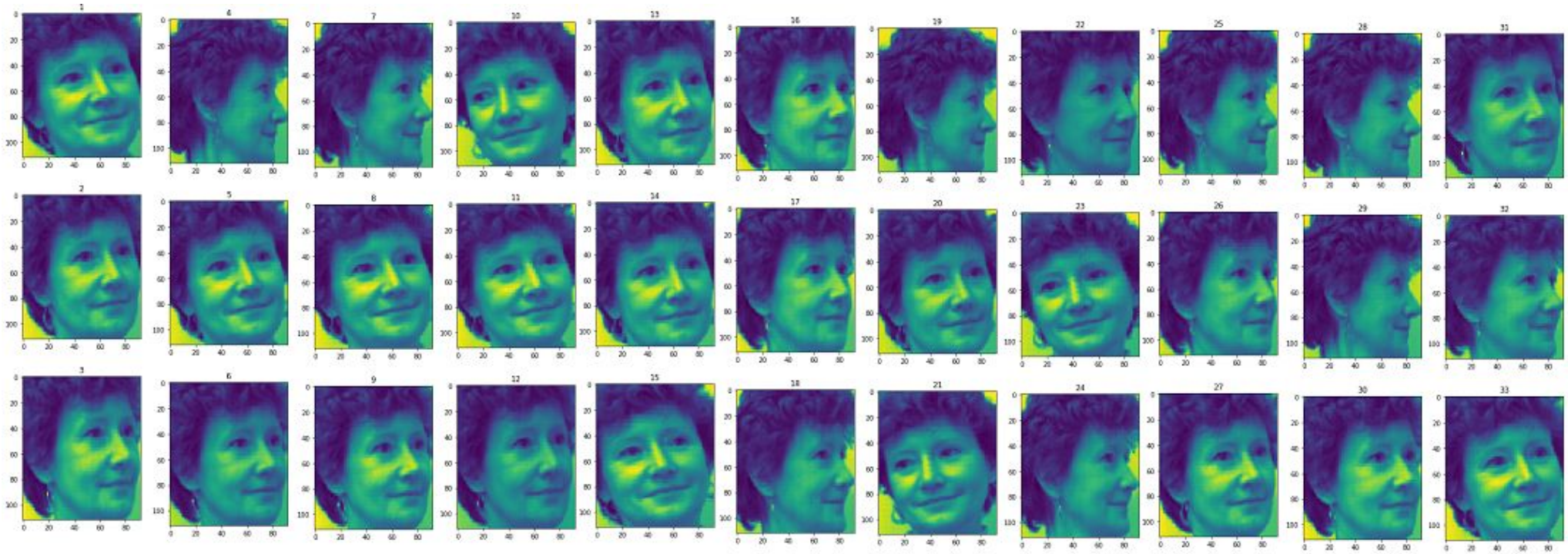
Introduction

- In this project, we want to explore how manifold learning methods contribute to appearance-based methods so as to order head directions
- Face pose determination is an important area of research in human computer interaction (HCI). An important problem in HCI is to determine one's focus of attention. This can be inferred from the person's head orientation
- The images of a person's face observed under different pose and lighting conditions can be considered as points in a high-dimensional vector space. In order to judge the similarities and detect the differences of these images, the dimensionality reduction is required.



Dataset

- The dataset contains 33 faces of the same person (112×92×33) in different angles. The pictures of the 33 faces are shown below.



Methods

- Methods Comparison
  - Multidimensional Scaling (MDS)
  - ISOMAP
  - Locally Linear Embedding (LLE)
  - Local Tangent Space Alignment (LTSA)
  - Modified LLE (MLLE)
  - Hessian LLE (HLLE)
  - Diffusion Map
  - t- distributed Stochastic Neighbor Embedding (t-SNE)

Multidimensional Scaling

- MDS aims to recover Euclidean coordinate in given pairwise distance metric.
- Given a set of data points $x_1, x_2, ..., x_n \in \mathbb{R}^p$, let

$$X = [x_1, x_2, ..., x_n]^{p \times n}.$$

The distance between different samples is denoted as $d_{ij}^2 = ||x_i - x_j||^2$ and then a squared distance matrix $D_{ij} = d_{ij}^2$ is created.

Afterwards, we compute $-\frac{1}{2}HDH^T$ as B where H is the Householder centering matrix.

By using eigenvalue decomposition $B = U\Lambda U^T$ with $\Lambda = diag(\lambda_1, ..., \lambda_n)$

we can choose the top k eigenvalues and corresponding eigenvectors to form the embedding data point $\tilde{T}_k = U_k \Lambda_k^{\frac{1}{2}}$ where

$$U_k = [u_1, ..., u_k], u_i \in \mathbb{R}^n$$
$$\Lambda_k = diag(\lambda_1, ..., \lambda_k)$$

with $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k > 0$.

ISOMAP

- ISOMAP is a extended method of MDS.
- The main difference is that ISOMAP uses pairwise geodesic distances between data points and graph shortest path distances to reconstruct the data.

The first step is to construct a neighborhood graph $G = (V, E, d_{ij})$ where $V = \{x_i : i = 1, ..., n\}$, $E = \{(i, j) : \text{if } j \text{ is a neighbor of } i \}$ and $d_{ij} = d(x_i, x_j)$.

Next, we compute the graph shortest path distance

$$d_{ij} = min_{P=(x_i,...x_j)}(||x_i - x_{t_1}|| + ... + ||x_{t_{k-1}} - x_j||)$$

which connects i and j.

- Compute $K = -\frac{1}{2}HDH^T$ thus the embedding coordinates

locally Linear Embedding (LLE)

- The central point of LLE is that any data point in a high dimensional space can be a linear combination of data points in its neighborhood.

Given a graph $G = (V, E)$, one can first do linear fitting. Namely, for each x_i and its neighbors $\mathcal{N}_i$, solve

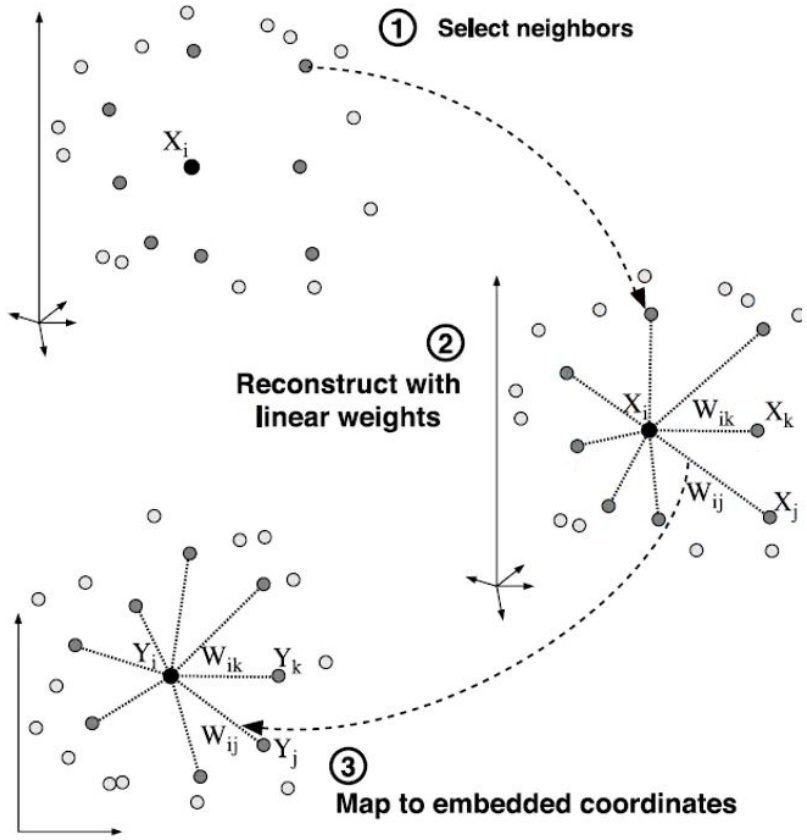
$$\min_{\sum_{j \in \mathbb{N}_i} w_{ij} = 1} \left\| x_i - \sum_{j \in \mathcal{N}_i} w_{ij} x_j \right\|^2$$

The next step is to do global alignment by computing $K = (I - W)^T (I - W)$ where

$$W_{ij} = \begin{cases} w_{ij}, & j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases}$$

- Eigen decompose $K = U \Lambda U^T$ with $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$

  thus embedding coordinates



locally Linear Embedding (LLE) improvement

- Modified LLE
- MLLE is another method to improve the performance of LLE. It uses multiple weight vectors projected from orthogonal complement of local PCA in each neighborhood. It replaces the weight vector above by a weight matrix W

- Hessian LLE
- HLLE is also a method to solve the regularization problem of LLE. It revolves around a hessian-based quadratic form at each neighborhood.

Local Tangent Space Alignment(LTSA)

LTSA is a modified version of LLE. Its first step is to do local PCA, i.e. computing local SVD on neighborhood of x_i, $x_{i_j} \in \mathcal{N}(x_i)$,

$$\tilde{X}^{(i)} = [x_{i_1} - \mu_i, \ldots, x_{i_k} - \mu_i]^{p \times k} = \tilde{U}^{(i)} \tilde{\Sigma} \left(\tilde{V}^{(i)} \right)^T$$

where $\mu_i = \sum_{j=1}^{k} x_{i_j}$.

The second step is to do tangent space alighment,

$$K^{n \times n} = \sum_{i=1}^{n} S_i W_i W_i^T S_i^T, \quad W_i^{k \times k} = I - G_i G_i^T$$

wehere $S_i^{n \times k} : [x_{i_1}, \ldots, x_{i_k}] = [x_1, \ldots, x_n] \, S_i^{n \times k}$ and $G_i = \left[1/\sqrt{k}, \tilde{V}_1^{(i)}, \ldots, \tilde{V}_d^{(i)} \right]^{k \times (d+1)}$.

Finally, we do eigenvalue decomposition $K = U \Lambda U^T$ and find the smallest $d+1$ eigenvectors of K with dropping the smallest one. The remaining d eigenvectors will give rise to d-mebedding coordinates.

Diffusion Map

Give a data set $x_i \in \mathbb{R}^d$, one can define a undirected weighted graph $G = (V, E, W)$. V and E are the same as before but W is a symmetric matrix,

$$W_{ij} = W_{ji} = exp(-\frac{d(x_i, x_j)^2}{t})$$

for $i \in \mathcal{N}_i$, otherwise 0.

Then, let $d_i = \sum_{j=1}^{n} W_{ij}$ and $D = diag(d_i)$. A random walk on graph G can be defined through Markov matrix,

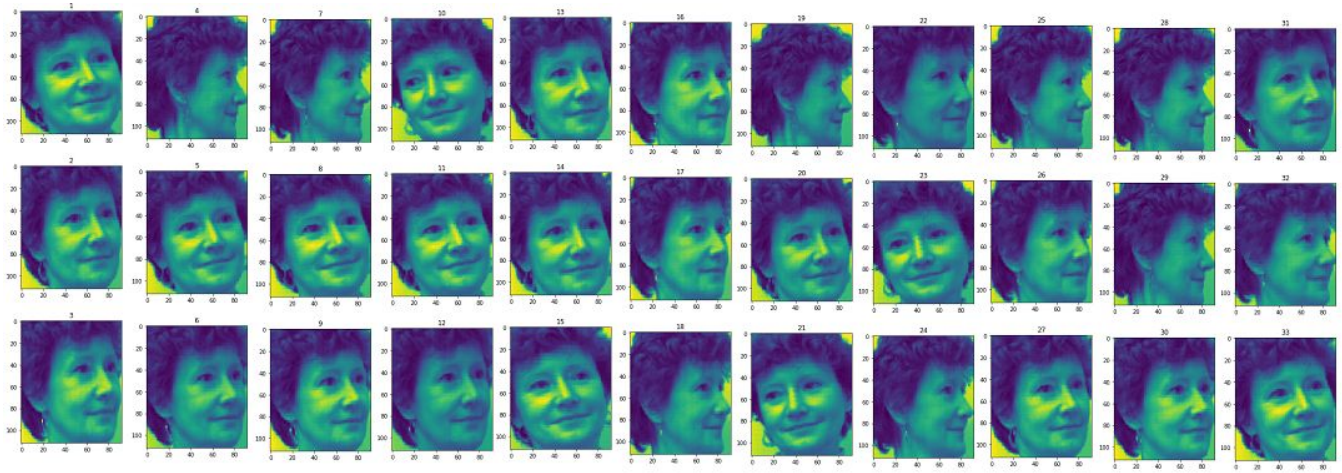
$$L = D^{-1}W - I$$

.

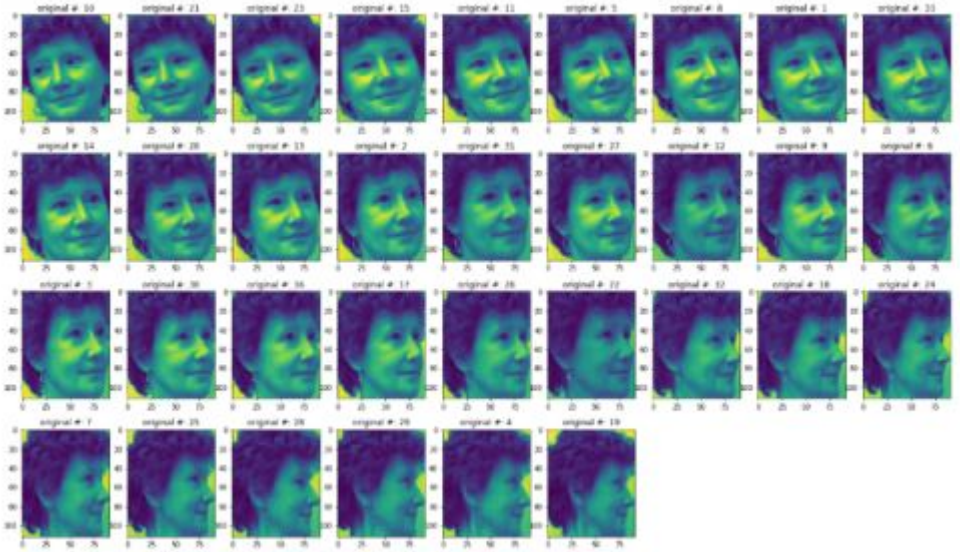
Finally, we do eigenvalue-decomposition of L and obtain the embedding coordinates.

Reorder the data according to face orientation

- Raw Data



- Ground Truth Data



Performance Evaluation

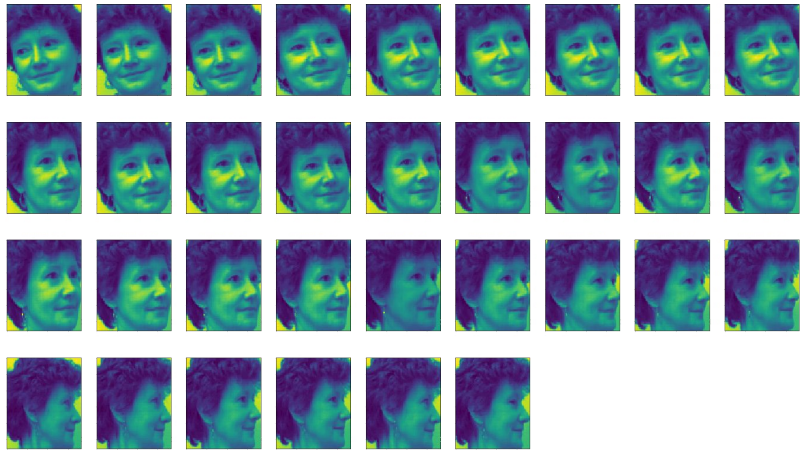
Original#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Ground truth	8	13	19	32	6	18	28	7	17	1	5	16	12	10	4	21
Diffusion Map	9	11	19	31	5	18	28	6	17	1	7	16	12	10	4	21
MDS	9	10	19	32	5	18	30	6	17	1	7	16	12	11	4	21
ISOMAP	7	13	19	32	5	18	28	10	17	1	6	16	12	9	4	21
LLE	8	13	19	32	5	18	28	7	17	1	6	16	12	10	4	21
MLLE	8	12	19	32	6	18	28	7	17	1	5	16	11	10	4	21
HLLE	8	13	19	32	6	18	28	7	17	2	5	16	12	10	4	21
Spectral Embedding	6	13	18	32	8	19	28	7	17	3	5	16	12	10	4	21
LTSA	8	13	19	32	6	18	28	7	17	2	5	16	12	10	4	21
t-SNE	8	12	20	30	14	22	23	10	17	1	18	16	5	2	6	9

Original#	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Ground truth	22	26	33	11	2	24	3	27	29	23	15	30	31	20	14	25	9
Diffusion Map	22	26	33	13	2	23	3	27	30	24	14	32	29	20	15	25	8
MDS	22	26	28	13	2	23	3	27	31	24	14	33	29	20	15	25	8
ISOMAP	22	26	33	11	2	24	3	27	29	23	14	30	31	20	15	25	8
LLE	22	26	33	11	2	23	3	27	29	24	14	30	31	20	15	25	9
MLLE	22	26	33	13	2	23	3	27	29	24	14	31	30	20	15	25	9
HLLE	22	26	33	11	1	23	3	27	29	24	14	31	30	20	15	25	9
Spectral Embedding	22	26	33	11	2	24	1	27	29	23	14	30	31	20	15	25	9
LTSA	22	26	33	11	1	23	3	27	29	24	14	31	30	20	15	25	9
t-SNE	33	25	24	27	15	11	4	32	29	21	19	28	31	13	7	26	3

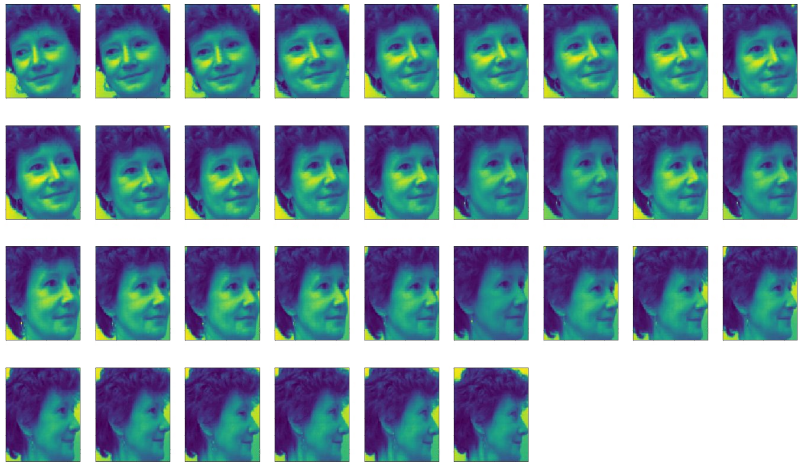
Methods	Diffusion Map	ISOMAP	LLE	MDS	HLLE	Spectral Emebedding	LTSA	MLLE	t-SNE
TAE	20	10	6	30	8	12	8	10	164

Visualization of Results

MDS

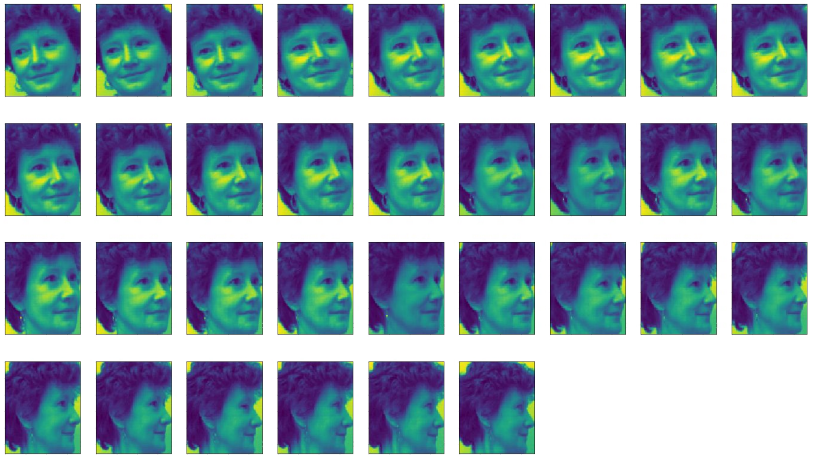


ISOMAP

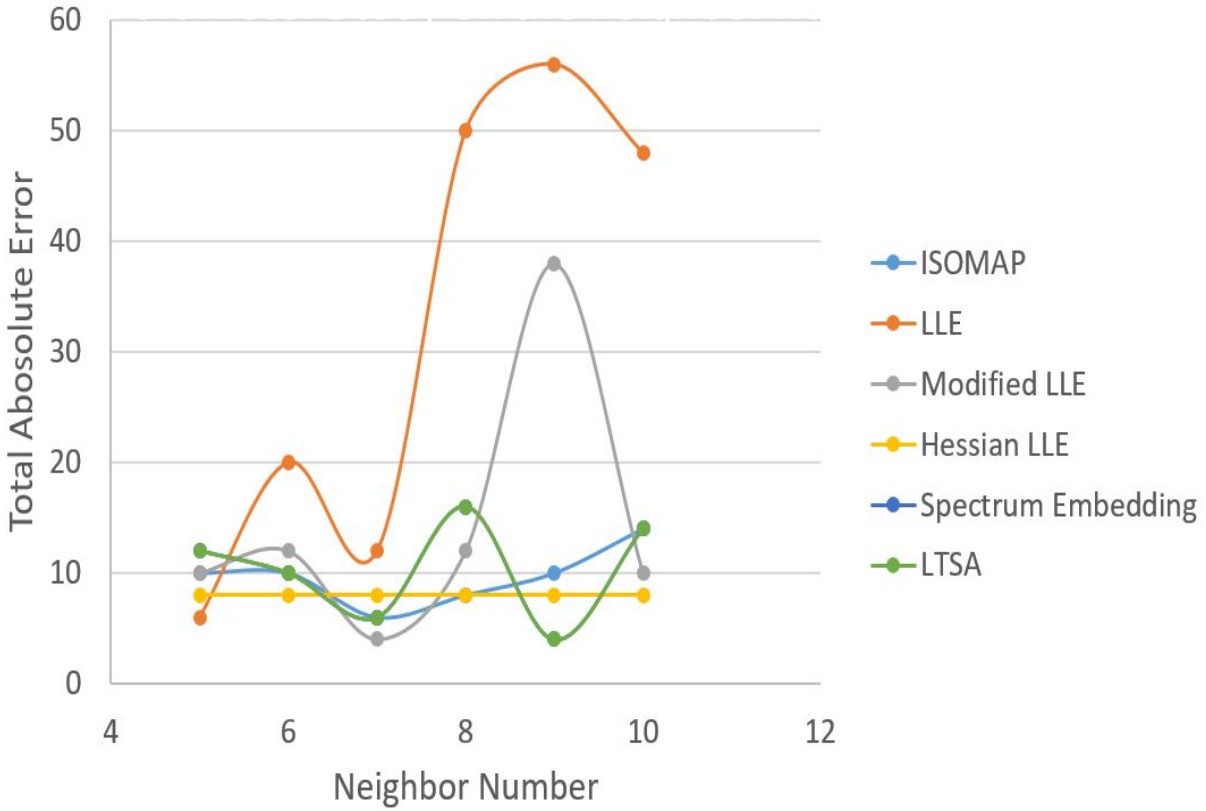


Diffusion Map

LLE



Hyper Parameter Study

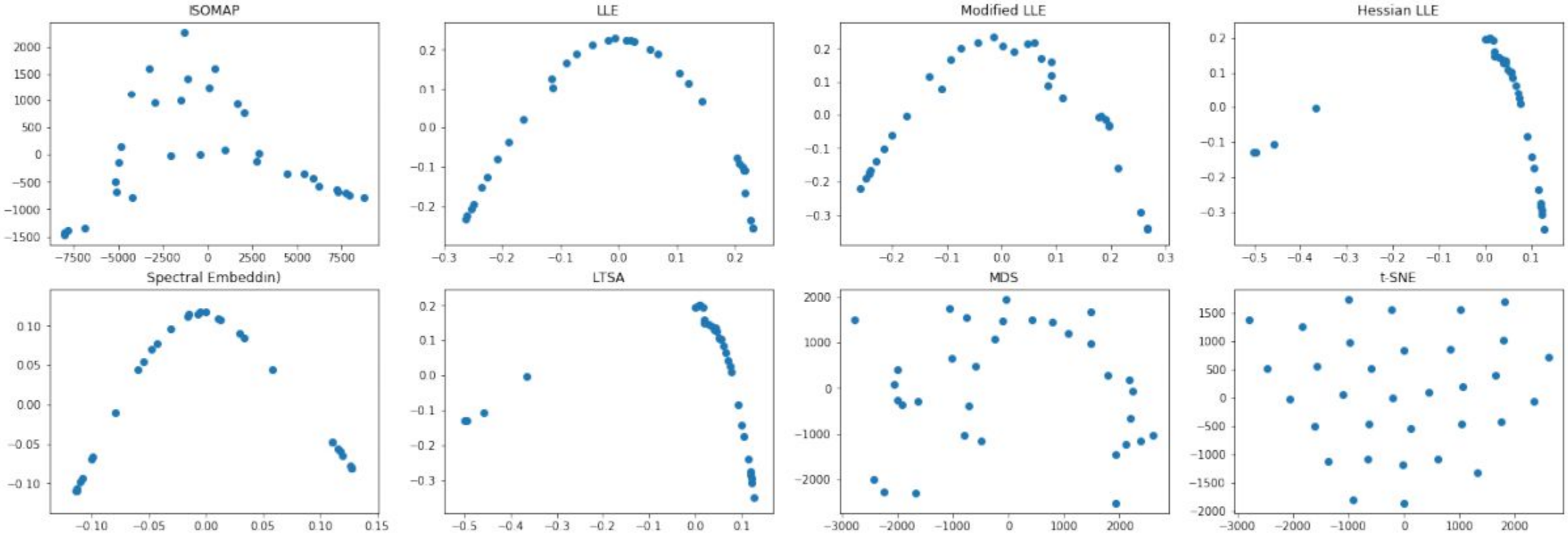


- We explored the effect of number of neighbors (k) on the sorting performance for each method.
- We tested for k =5 to 10, the methods were all applied with sklearn.manifold library.
- Compared with the other methods, LLE tends to be affected by k more. In terms of performance versus number of neighbors, we can find that 5 tends to be the best for most methods.

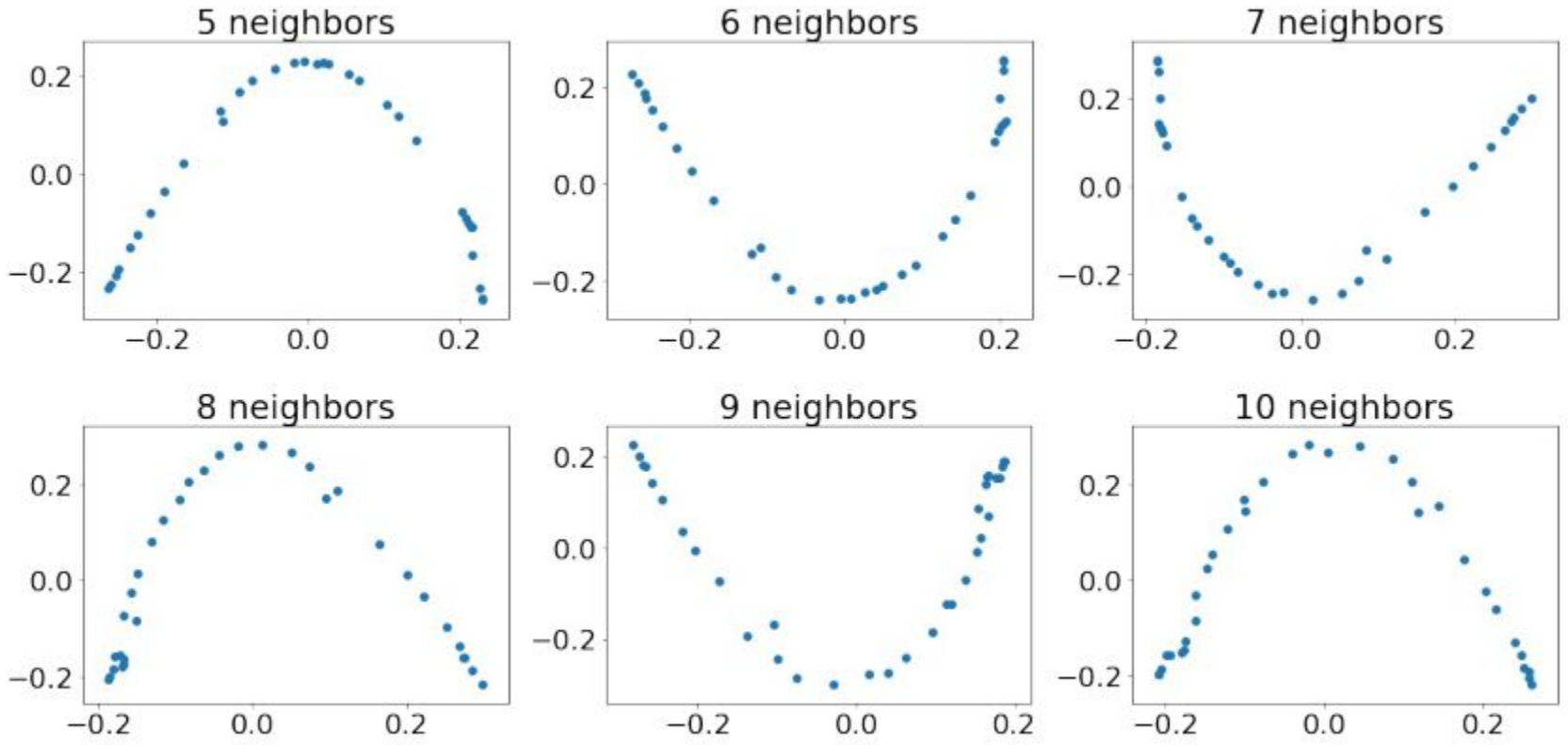
2D Embedding comparison

- We compared the 2D embedding graph for different methods based on sklearn.manifold library.
- We assigned the number of neighbors to be 5.
- Obviously, LLE and Spectral embedding seem to better than others in terms of x-axis which represents the first component. With the results before taken into consideration (i.e. TAE of different methods), we know that LLE should be the best.

Manifold Learning with 2 components, 5 neighbors



2D Embedding comparison



- Inspired by the results in previous slide we also studied the effect of number of neighbors on the 2D embedding graph.
- Here, we only take LLE for an example . One interesting phenomenon is that the shape of the graph can be both ’V’ and ’Λ’.
- However, this phenomenon makes no difference to the conclusion. The focus here is the discrepancy between different points in the figure.
- As the number of neighbors increases, more and more points tend to be indistinguishable. This provides us with another way to understand that the number of neighbors of 5 should behave the best.

Conclusion and Further Work

Conclusion

- All the methods we explored show reasonable sorting order except t-SNE.
- LLE exhibits the lowest TAE performs best.
- Parameter study was performed on the number of nearest neighbour k and 5 proved to be the best, when number of component is 2.

Further Work

- Apply these methods into a more complicated dataset. (e.g. head images with both turning and nodding motion).
- By adding one degree of freedom, more eigenvectors may be involved in deciding the order.
- More techniques can be combined with the methods in this report to achieve state-of-art results.