Manifold Learning and Classification methods for MNIST dataset

LIU Yunqin
Division of EMIA
The Hong Kong University of Science and Technology
yliuns@connect.ust.hk

Abstract

In this final project, we explore several manifold learning methods on MNIST dataset to find its lower-dimensional embedding. Based on the embedding results, we use various classification methods to categorize the data. Particularly, we perform MDS, ISOMAP, LLE, LTSA, Diffusion Maps,and t-SNE to reduce the dimension of data, and use SVM, KNN, Random Forest methods to classify the data.

1 Introduction

Manifold learning is a subfield of machine learning that deals with the problem of dimensionality reduction in high-dimensional data. The goal of manifold learning is to find a low-dimensional representation of the data that preserves the underlying structure of the data. In many real-world applications, data is often high-dimensional and complex, making it difficult to analyze and visualize[1]. Manifold learning provides a solution to this problem by identifying the intrinsic low-dimensional structure of the data and projecting it onto a lower-dimensional space. This allows for easier analysis and visualization of the data.

Some well-known manifold learning methods include: t-Distributed Stochastic Neighbor Embedding (t-SNE), ISOMAP, Locally Linear Embedding (LLE), Diffusion Maps, Local Tangent Space Alignment (LTSA), and Multidimensional Scaling (MDS)[2]. In this project, we implement these methods to find and visualize the lower-dimensional embedding of the high-dimensional MNIST dataset. Based on the embedding results, we use various classifiers to categorize the data and calculate the accuracy of combined methods.

2 Dataset

We select the MNIST dataset, which contains 60,000 training images and 10,000 test images, available on the website

http://yann.lecun.com/exdb/mnist/

some images in dataset, shown in Fig. 1. The images are in grayscale with the size 28-by-28 and are labeled for 10 distinct types. For this project, we selected 10,000 images for our experiments.

Submitted to 30th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

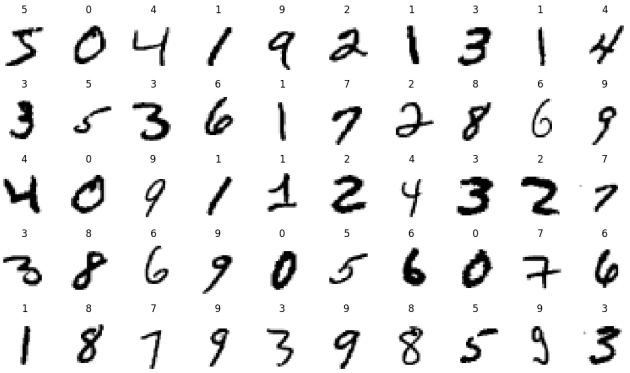

Figure 1: images in MMIST dataset.

3 Methodology

In this project, a variety of nonlinear manifold learning methods are implemented to embed the high dimensional MNIST data into the lower dimesional space. Details of each ways are shown below.

3.1 MDS

Multidimensional scaling (MDS) is a method that aims to preserve the pairwise distances between data points in a lower-dimensional space.The algorithm works by computing the eigenvectors of the distance matrix, which captures the underlying structure of the data. The distance matrix can be constructed using various measures such as Euclidean distance, correlation distance, or cosine distance.

Next, MDS uses eigenvalue decomposition to find the optimal low-dimensional representation that minimizes the stress function, a measure of the difference between the pairwise distances in the original space and the low-dimensional space.

3.2 ISOMAP

ISOMAP (Isometric Feature Mapping) is a nonlinear dimensionality reduction technique based on the concept of geodesic distances, which are distances measured along the shortest path on a curved surface, rather than the straight line distances used in Euclidean space.The steps of ISOMAP algorithm are as follows:

Firstly, construct a neighborhood graph G from the given distances dX(i,j) using the specified method, such as $\epsilon - ball$ approach or kNN approach.

Secondly, compute the shortest-path distances dG(i,j) between all vertices of G by using Dijkstra's algorithm.

Finally, Apply MDS method with dG(i,j) as input distances to find a k-dimensional representation Y of the original data.

3.3 LLE

LLE (Locally Linear Embedding) seeks to preserve the local structure of the data and constructs a low-dimensional representation by finding a linear combination of the nearest neighbors of each point that best approximates the point.

LLE works by constructing the neighborhood graph: For each data point, find its k-nearest neighbors and construct a graph where each point is connected to its neighbors.

Next, Compute the weight matrix: For each point, compute the weights that best reconstruct the point from its neighbors using linear weights. The weights can be computed by solving a system of linear equations:

$$\mathbf{W} = argmin \sum_{i=1}^{n} \left\| x_i - \sum_{j \in N_i} \omega_{ij} x_j \right\|^2 \tag{1}$$

subject to $\sum_{j \in N_i} \omega_{ij} = 1$. Here, $x_i$is the i-th data point, N_i is the set of indices of its neighbors, and W is the weight matrix.

Once the weight matrix is estimated, LLE compute the low-dimensional representation of the data by minimizing the reconstruction error of the weights using the MDS approach. The embedding can be computed by solving a system of linear equations:

$$\mathbf{Y} = argmin \sum_{i=1}^{n} \left\| \mathbf{w}_i^T \mathbf{Y} - \mathbf{y}_i \right\|^2 \tag{2}$$

subject to $\mathbf{Y}^T \mathbf{1} = 0$. Here, $\mathbf{Y}$is is the low-dimensional embedding, $\mathbf{w}_i$ is the i-th row of the weight matrix, and $\mathbf{y}_i$is the embedding of the i-th data point..

3.4 LTSA

Local Tangent Space Alignment (LTSA) works by first constructing a graph representation of the data, where each data point is considered as a node in the graph. The edges of the graph are constructed based on the pairwise distances between the data points.

Then, LTSA estimates the local tangent space for each data point by finding the k-nearest neighbors of each point and using principal component analysis (PCA) to estimate the local tangent space. The global optimization algorithm is used to find a lower-dimensional representation of the data that preserves the local geometry of the data. This is done by aligning the tangent spaces of neighboring data points in the high-dimensional space and then projecting the aligned tangent spaces onto a lower-dimensional space.

3.5 Diffusion Maps

To apply diffusion maps to a high-dimensional data set, the first step is to construct a similarity matrix that measures the pairwise similarity between the data points.This similarity matrix $\mathbf{A}$ is typically constructed using a Gaussian kernel function that takes into account the distance between the data points in the high-dimensional space:

$$a_{ij} = \exp\left\{ -\frac{\|x_i - x_j\|^2}{\epsilon} \right\} \tag{3}$$

Next, the similarity matrix is used to construct a Markov chain, where each data point is considered as a node in the chain. The transition probabilities matrix $\mathbf{P}$ between the nodes are calculated based on the similarity matrix:

$$p_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{ij}} \tag{4}$$

Once the Markov chain is constructed, the diffusion process is applied to the chain to obtain a low-dimensional representation of the data. The diffusion process spreads out the probability mass over the nodes in the chain, emphasizing the important features and relationships between the data points.

3.6 t-SNE

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a nonlinear dimensionality reduction technique based on the concept of preserving the similarity between data points in high-dimensional space.

t-SNE works by first Calculate the similarity between each pair of high-dimensional points using a Gaussian kernel function

$$P_{j|i} = \frac{\exp\Big(-\|x_i - x_j\|^2/2\sigma_i^2\Big)}{\sum_{k\neq i}\exp\Big(-\|x_i - x_k\|^2/2\sigma_i^2\Big)} \tag{5}$$

Next, t-SNE constructs a lower-dimensional space and a probability distribution over pairs of points in the lower-dimensional space. The probability distribution over the lower-dimensional space is constructed using a Student's t-distribution

$$q_{j|i} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k\neq i}(1 + \|y_i - y_k\|^2)^{-1}} \tag{6}$$

The goal of t-SNE is to find a mapping between the high-dimensional space and the lower-dimensional space that minimizes the difference between the two probability distributions. This is done using a gradient descent algorithm that adjusts the mapping between the two spaces to minimize the Kullback-Leibler divergence between the two probability distributions.

the Kullback-Leibler divergence between the two probability distributions:

$$KL(P||Q) = \sum_{i,j} p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{7}$$

the gradient of the cost function with respect to the low-dimensional data points:

$$\frac{\partial C}{\partial y_i} = 4\sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \tag{8}$$

4 Experiments

In this project, a variety of nonlinear manifold learning methods are implemented to embed the high dimensional MNIST data into the lower dimesional space. Details of each ways are shown below.

4.1 Results for embedding

4.1.1 MDS

In this experiment, We first test the performance of MDS embedding. The distribution of this set of figures based on the first two eigenvectors can be shown in Fig. 2. It is clear that, images with similar shape are located near to each other. The mapped data preserves the distance in original space. The separation is great for some numbers, such as 7 and 0, but mediocre for others.

4.1.2 ISOMAP

Then, We test the performance of ISOMAP embedding on k = 5 nearest neighbor graph and the distribution of images based on ISOMAP embedding is shown in Fig. 3.Compared to MDS, ISOMAP uses geodesic distances to preserve the intrinsic geometry of the data and their distributions are similar but inverse. In MDS, the number 0 are mainly clustered in the lower left corner, while they are clustered in the upper right corner in ISOMAP. The performance of ISOMAP in distinguishing those numbers is superior to MDS because the distances are larger between the various numbers.

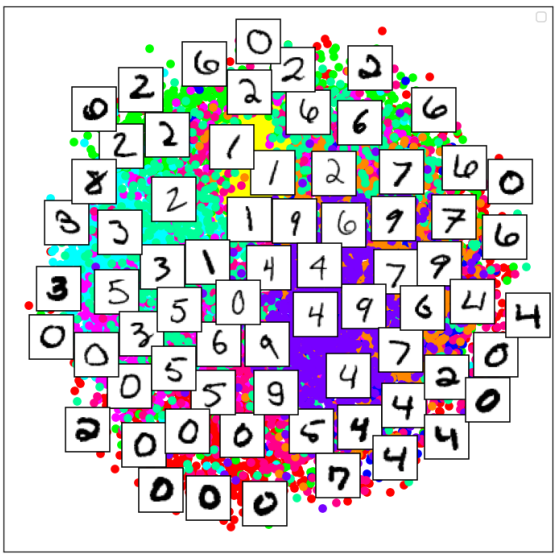

Figure 2: Distribution of the digital images based on MDS embedding.

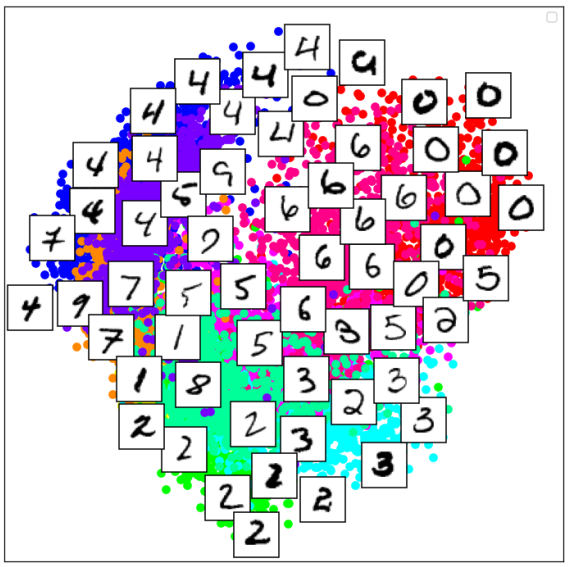

Figure 3: Distribution of the digital images based on ISOMAP embedding.

4.1.3 LLE

Next, We test the performance of LLE embedding on k = 5 nearest neighbor graph and the distribution of images based on LLE embedding is shown in Fig. 4. Compared to ISOMAP, LLE uses linear mappings to preserve the local relationships between the data points. LLE has achieved better performance than IOSMAP in distinguishing some numbers, such as number 0 and 7. These two numbers are located on opposite side of the first eigenvector from LLE and can easily be categorized into different types.

4.1.4 LSTA

After that, We test the performance of LSTA embedding on k = 5 nearest neighbor graph and the distribution of images based on LSTA embedding is shown in Fig. 5. Different from LLE which focus on maintaining neighborhood distances, LTSA describes the local geometry of each neighborhood

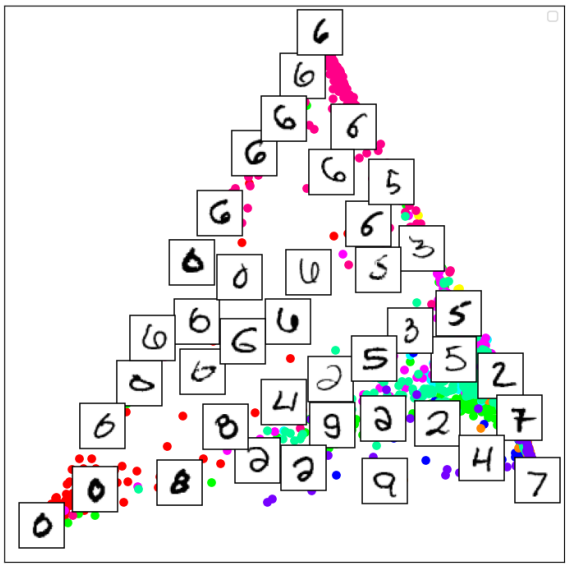

Figure 4: Distribution of the digital images based on LLE embedding.

through its cut spaces and performs global optimization to align these local cut spaces to obtain the corresponding embeddings. However, LSTA seems to perform terribly on the MNIST dataset because the distribution of numbers are disorder. For example, the number 2 and 5 appear in various parts of the distribution.

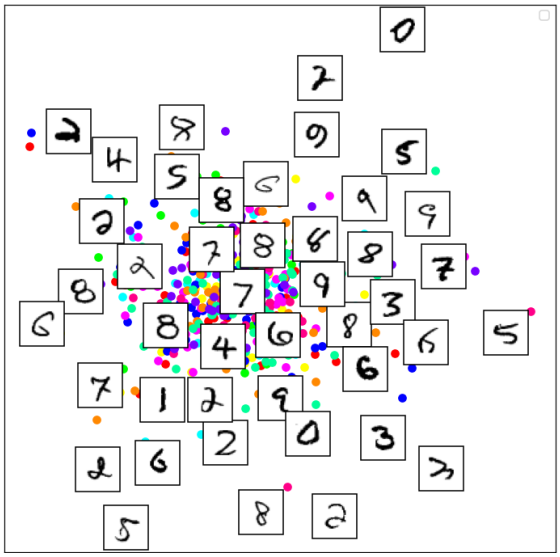

Figure 5: Distribution of the digital images based on LSTA embedding.

4.1.5 Diffusion Maps

Also, We test the performance of Diffusion Maps embedding and the distribution of images is shown in Fig. 6. Diffusion Maps embedding has achieved great performance in separating and clustering some digits, such as digital 0 and 7. They are far away from each other on the distribution map.

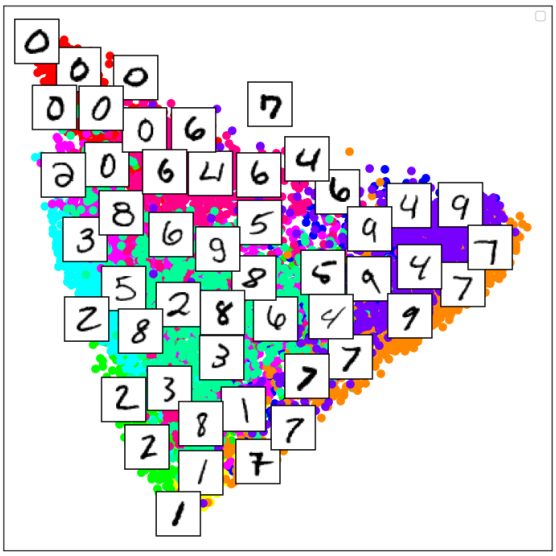

Figure 6: Distribution of the digital images based on Diffusion Maps embedding.

4.1.6 t-SNE

Finally, We test the performance of t-SNE embedding and the distribution of images based on t-SNE embedding is shown in Fig. 7. Compared to Diffusion Maps, t-SNE does a better job at separating and clustering the digits. t-SNE has discovered not only the clusters of digits, but also the angle of digits. For example, the cluster of digital 1 shift the angle to the right, with the increasing of the second eigenvector.Therefore, the t-SNE embedding can be considered as a good manifold learning method to capture the structure behind this dataset.

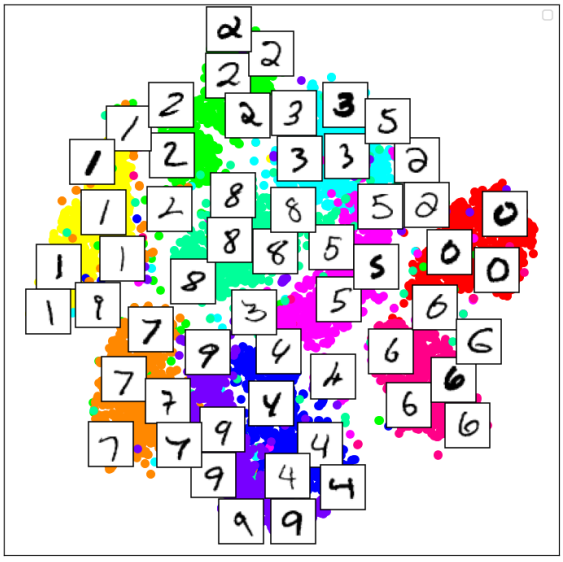

Figure 7: Distribution of the digital images based on t-SNE embedding.

4.2 Results for classification

Based on the 2D embedding results, we choose k-nearest neighbors(KNN), random forest and SVM as the classifiers to identify the types of the digital in the dataset. The accuracy for different combinations of embedding methods and classifiers are listed in Table (1).We can conclude that the

t-SNE+Random Forest method has achieved the best performance. Meanwhile, the accuracy of t-SNE is always the highest when using the same classifier compared to other manifold learning methods. It has been shown that t-SNE performs best at separating and clustering on the MINST dataset.

Table 1: The accuracy for different combinations of embedding methods and classifiers

Embedding	Accuracy via classifiers		
Methods	SVM	Random Forest	KNN
MDS	50.3%	44.4%	45.1%
ISOMAP	59.7%	54.6%	55.6%
LLE	80.9%	85.7%	85.8%
LTSA	11.6%	11.6%	30.2%
Diffusion Maps	56.2%	51.2%	52.2%
t-SNE	93.8%	95.3%	94.6%

5 Conclusion

In this project, we investigate MNIST dataset with dimensional reduction methods, including MDS, ISOMAP, LLE, LTSA, Diffusion Maps and t-SNE.The resulting are visualized and compared with each other. We also classify the types of the digits by using different combinations of embedding methods and classifiers and the t-SNE+Random Forest method has achieved the best performance.

References

[1] Farzana Anowar, Samira Sadaoui, and Bassant Selim. "Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE)". In: *COMPUTER SCIENCE REVIEW* 40 (May 2021). ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2021.100378.

[2] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *JOURNAL OF MACHINE LEARNING RESEARCH* 9 (Nov. 2008), pp. 2579–2605. ISSN: 1532-4435.