Manifold Learning and Pattern Classification

Mikhail Podvalkov

Due to May 6, 2022

1. Introduction

The data for learning was a part of the ORL face database. The training data consisted of 20 individuals; each had ten images for each size of 92x112 pixels. The computer assignments aim to design a Laplacian Eigenmaps to reduce the complexity of images and extract features. The computer assignments created a LE in MATLAB code [1] with a different number sub-manifold dimension. The best model was used to train the KNN classifier and test it.

2. Theory

Laplacian Eigenmaps is a nonlinear dimensionality reduction algorithm that assumes the data lies on a low-dimensional manifold. (Figure 1) [2].

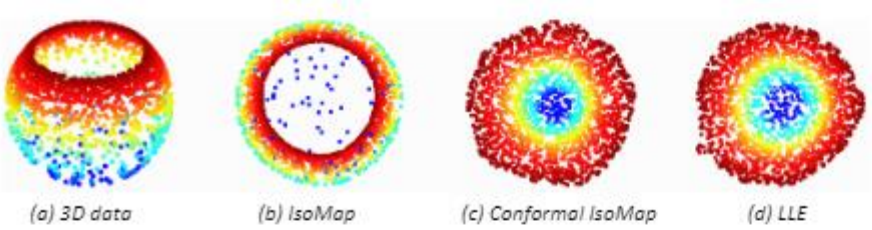

Figure 1 The example of the representation of the data.

The first step of the method is to find the close image by using the K-nearest neighbor because the algorithm works with the local data structure. This close image was used to create a local similarity symmetric matrix W. The neighboring nodes had weights equal to the kernel distance; otherwise, weights equal zero (1).

$$w_{ij} = k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2*\sigma^2}} \; otherwise \; w_{ij} = 0 \; (1)$$

The weights matrix was used to find the diagonal matrix D (2), which was used in flinging Laplacian matrix L (3).

$$D_{ii} = \sum_j w_{ij} \; (2)$$

$$L = D - W \; (3)$$

The $D^{-1}L$ matrix was used to extract the eigenvalues and eigenvectors (4), and the M lowest eigenvalue is the sub-manifold low-dimensional space.

$$Ly = \lambda Dy \quad (4)$$

This sub-space was used as training data for the KNN classifier. The method to add test images in this space is the incrementally sample embedded method (5)

$$y = \sum_{i=1}^{K} \frac{k(x, x_i)}{\sum_{j=1}^{K} k(x, x_j)} y_i \quad (5)$$

where x is a new image and y is an eigenvector or low-dimension space

3. Results and discussion

The first step is to find the weights matrix of the closed images. After that, the Laplacian matrix and sub-manifold low-dimensional space were found. The test data was randomly picked from one image for each individual. The experiments were provided ten times to get all the images to become a test and training data. The Matlab code provides two types of sub-manifold low-dimension space for M=50 and M=100. The visualization sub-manifold low-dimension space is presented in Figure 2.

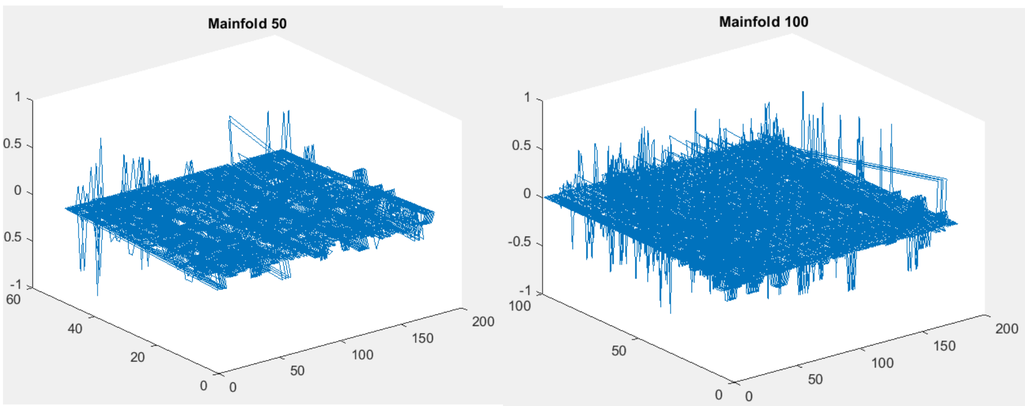

Figure 2 The visualization sub-manifold low-dimension space for M=50 and M=100

The test images were incrementally by sample embedded method to test data for all KNN classifiers. All ten LE model was used in the KNN classifier with k=7. The result of KNN with LE method M=50 and M=100 was presented as a confusion matrix in Figure 3.

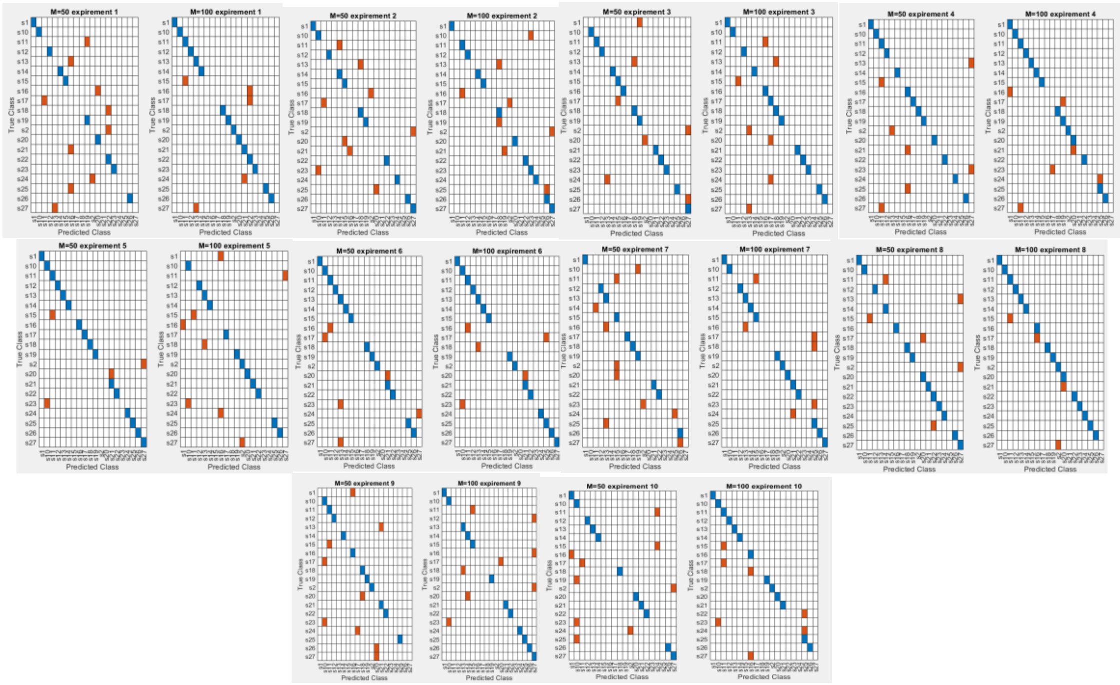

Figure 3 The confusion matrix for each experiment with M=50 and M=100 LE.

The average accuracy of all experiments for M=50 is 0.625 and for M=100 is 0.68. The best accuracy result for M=50 and M=100 is 0.8.

4. Conclusion.

The LE method is similar to the PCA extraction method. However, unlike PCA, the Laplacian Eigenmaps algorithm does not try to preserve exact pairwise distances or even relative pairwise distances between faraway points. The algorithm focuses on mapping the embedding corresponding to nearest neighbors as close together as possible. The LE techniques used the eigenvectors corresponding to the smallest eigenvalues in contrast to PCA did. Moreover, the LE is based on the nxn matrix smaller than in PCA.

The LE reduces the complexity (dimensionality) and size of training data without losing the critical information, allowing a more straightforward classification method. Furthermore, the LE could be used in nonlinear embedding, which increases the number of problems to which this method could be applied. Also, the eigenvector directly gives the embedding of the point. Therefore, the Laplacian Eigenmaps could be used in image classification because the dimension of one image is its resolution, which is significant. The LE could effectively use for pre-processing of image classification. However, the LE could work if we define the neighbor data point, which could be a computational task.

Reference

1. The official MATLAB site: https://www.mathworks.com/
2. The professor M.R Azimi lecture 23-24