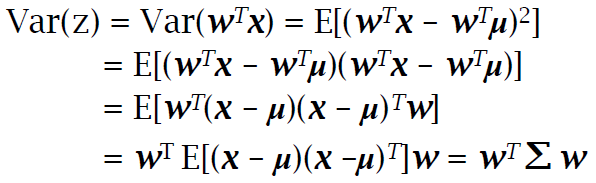
**Report for Homework 4**

**1.Introduction**

## In this homework, we deal with the problems about Dimensionality Reduction using [Dimensionality Reduction Toolbox.](http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html) We install and do some experiments on some methods in this toolbox for machine learning. Mainly we work on Principle Component Analysis, Linear Discriminant Analysis, Multi-Dimensional Scaling . First, we study pca.m function and use this function to implement dimensionality reduction on a specific dataset Optdigits. Then we reconstruct the dataset and compute the reconstruction error based on different principle components. Also we use Dimensionality Reduction toolbox to reduce the dimension of this dataset to plot the 2D graph of the first eigenvector and second eigenvector. Moreover, we select out a nonlinear method and apply this method to do dimensional reduction. And we compare the graphical results with these methods above. Finally we do dimensional reduction on a unlabeled dataset mystery\_data and compare the results using PCA and the chosen method Laplacian.

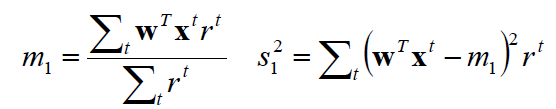
**2.Methods**

For the first part, we study pca.m function and work on this function for dimensional reduction for the dataset Optdigits. PCA is the abbreviation for Principal Component Analysis. This method tries to find a low-dimensional space such that when the data **X** is projected there, information loss is minimized. The projection of **X** on the direction of **W** is: **Z** = **WTX** . In order to maximize Var(**Z**), we need to find a proper directional **W** based on the formula below. Using Lagrange formulation , we can solve this optimization problem subject to **||W||** = 1.We find that the solution is the eigenvector of variance **Σ** , so we choose the eigenvector **W1** with the largest eigenvaluefor var(Z) to be maximum. Then we can take another eigenvector orthogonal to **W1** for the second principal component. In this way ,we can reduce the dimension to k by choosing k eigenvectors related to k largest eigenvalues λ. The distribution for the data points can be plotted based on these components. Also we can reconstruct the data and compute the reconstruction error using the formula in pca.m which is just the opposite process of reduction.

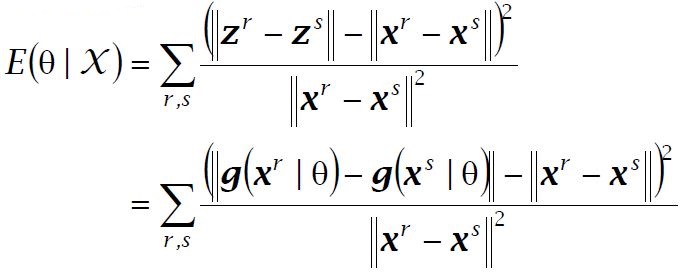




For LDA, its full name is linear discriminant analysis which is a supervised method dimensional reduction in classification problem. The basic idea is to find a low-dimensional space such that when **X** is projected , the classes are well separated. Therefore, we will be looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible. This method uses a linear discriminant to characterize the classification level based on the following formula. By optimizing the discriminant, the best direction **W** can be decidedto reduce the dimension. Then we can compute the projection with **W**.



Another method for dimensional reduction is multidimensional scaling. A n x n matrix D is called a distance or affinity matrix if it is symmetric, dii = 0 , and dij > 0 , i ≠ j. Given a distance matrix D(X), MDS attempts to find n data points y1, y2, ......yn in a low dimensions d , such that if dij(Y) denotes the Euclidean distance between yi and yj ,then DY is similar to D(X). Then, the pairwise distances between N points is transformed onto a low dimension and at the same time the rank order of the distances are preserved. The projection function is **Z** = **g**(**X**|**ϴ** ) . Based on the following formula ,we can find a **ϴ** which minimizes Sammon stress for point r and s.



Also we select out a Laplacian Eigenmaps-based mapping method for dimensionality reduction. Laplacian eigenmap is an unsupervised, explicit, and layered feature transform, so it is also suffered from the new sample problem. Given k points X1,X2,...XK  in Rl , we construct a weighted graph with k nodes, one for each point, and a set of edges connecting neighboring points. The embedding map is now provided by computing the eigenvectors of the graph Laplacian. Let G(V,E) be a undirected graph without graph loops. The Laplacian of the graph is

dij if i=j (degree of node i)

Lij = -1 if i≠j and (i,j) belongs to E

0 otherwise

If node i and j are connected , put weight matrix



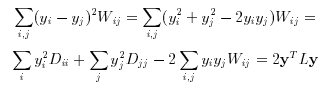
Consider the problem of mapping the weighted graph G to a line so that connected points stay as close together as possible. Let **y** = (y1,y2,......,ym) T be such a map. A reasonable criterion for choosing a ''good" map is to minimize the following objective function



under appropriate constraints. It turns out that for any **y** , we have



where L = D -W. And notice that Wij is symmetric and . Thus



This calculation also shows that L is positive semidefinite. Therefore, the minimization problem reduces to finding



Because L is a positive semidefinite , so by Lagrange formulation ,the vector **y** that minimize the objective function is given by the minimum eigenvalue solution to thegeneralized eigenvector problem



So we can leave out the eigenvector y0 corresponding to eigenvalue 0 and use the next m eigenvectors for embedding in m-dimensional Euclidean space. In this way , we can implement dimensional reduction using Laplacian Eigenmap-based method.

**3. Results**

For the first problem, we copy pca.m function to MATLAB and study the formula in it. We find that this function can output the principal components, the reconstruct dataset and the mean-square reconstruction error based on input dataset and the dimension of projection space.

For the second problem, based on dataset Optdigits, we investigate that how the number of principal components influence the reconstruction error using pca.m function. The figure.1 below shows the relationship between the reconstruct error and number of principal components, we can find that as the number of dimension increases, the reconstruction error decreases significantly.

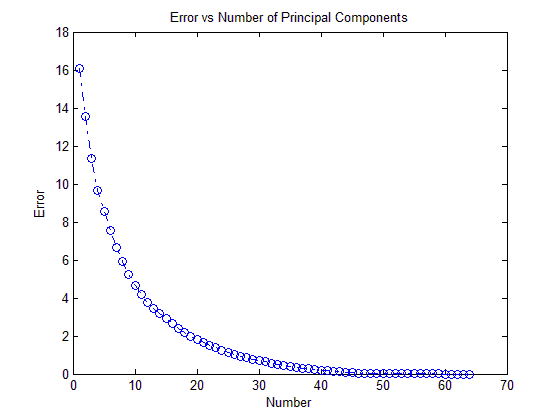


Figure.1 Reconstruction Error VS Number of Principal Components

For the third problem, using pca.m ,we can obtain the output eigenvectors for 2 dimension. Then we plot a graph shown below using these two eigenvectors and the digits represented in the last column of matrix dig\_tra. Every distinct color represent a digit it represents. We find that some class of data points are separated such as data points for digits 3,4,6 and 7. But some data points in the middle are not poorly separated such as that for digits 1,5 and 8.

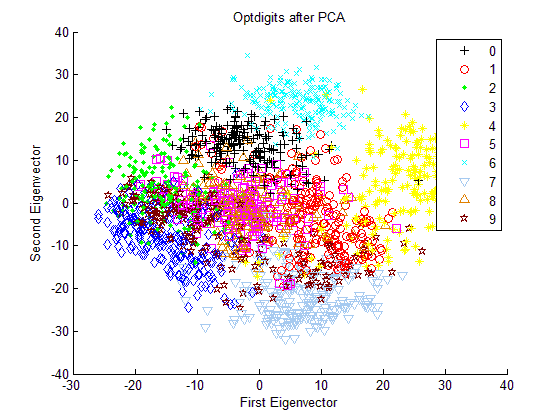


Figure.2 Optdigits distribution for PCA

For problem 4, we download and install Dimensional Reduction Toolbox. Also we read the documentation and do some practice on how to use this toolbox. For problem 5, we use this toolbox for PCA, LDA and MDS and plot the graphs for distribution of projected data points shown below. We find that for PCA, the result is almost the same as that based on pca.m above. Some class of data points in the middle are not well separated. For LDA, we see that the data points are distributed around a 2D line. And only a few data points are well separated such as points for digit 0. Most of the data points cannot be separated using this method for two dimensions. And for another method MDS, the result is almost the same as that of PCA. Part of data points are well separated and others in the middle are not.

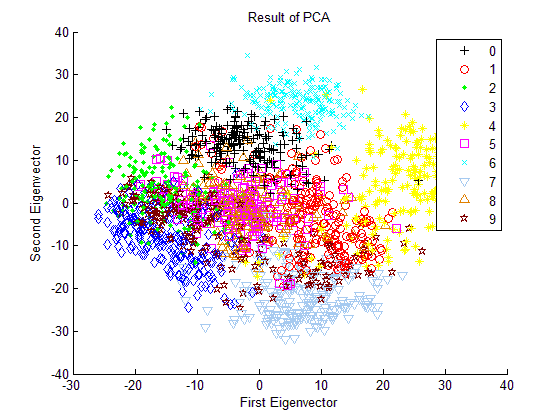


Figure. 3 Distribution of the Principal Components for PCA

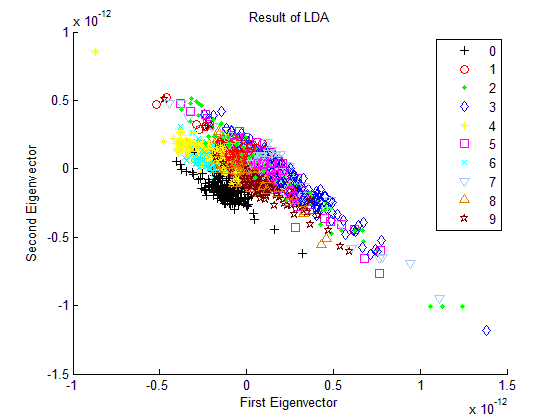


Figure. 4 Distribution of the Principal Components for LDA

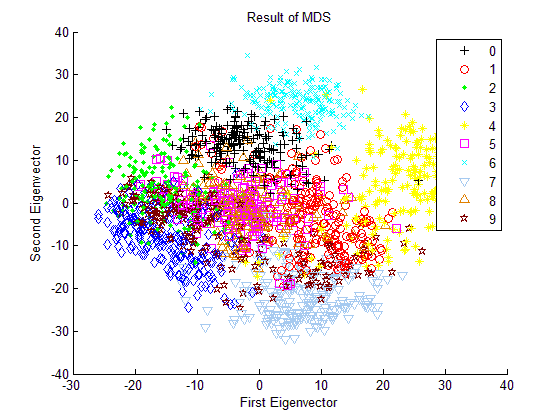


Figure. 5 Distribution of the Principal Components for MDS

For problem 6, we pick and study Laplacian eigenmap. There are mainly two parameters in this method including k and sigma. Considering sigma has minimal impact in the performance of our method, we do simulations under different parameter k to get better performance for this method and the graphs for different parameter k are shown below.

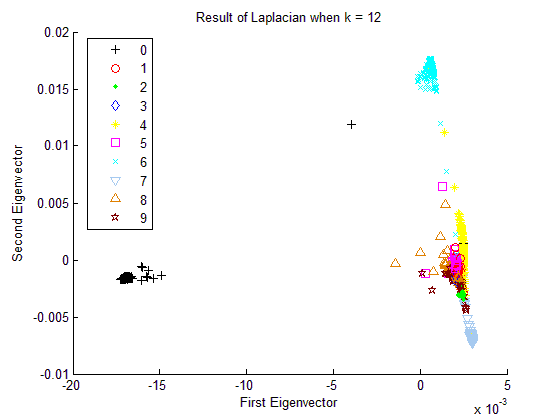


Figure. 6 Distribution for Laplacian when k =12

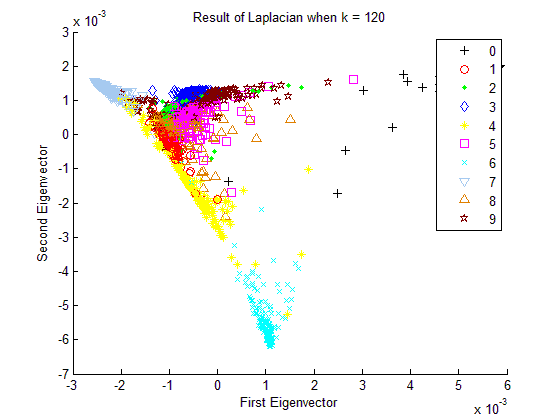


Figure. 7 Distribution for Laplacian when k =120

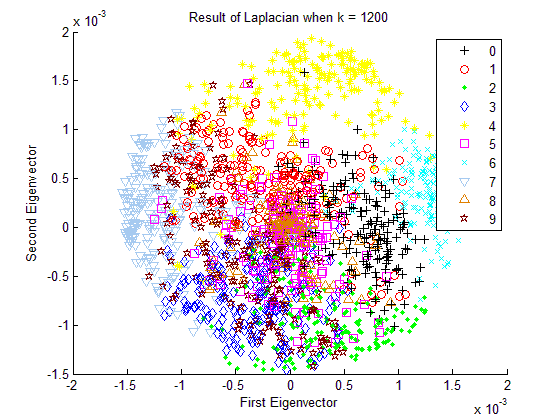


Figure. 8 Distribution for Laplacian when k =1200

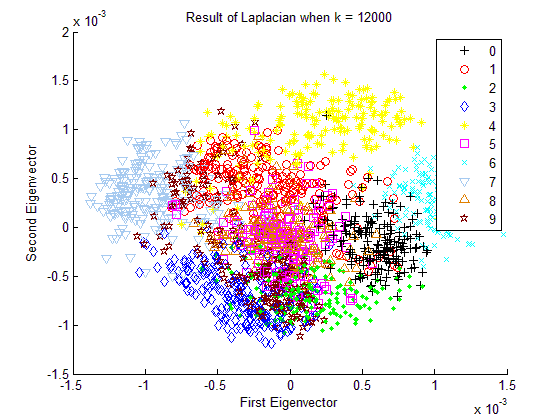


Figure. 9 Distribution for Laplacian when k =12000

We find that as the value of k increases, the data points seems to get closer to each other. At first , when k =12 , the data points within different classes are far away from each other .For example points for digits 0, 6 and 7 are well separated. But when k = 12000 , the data points for different digits all gather into a small scope of area just from -1.5 to 1.5 for both X and Y axis. And most of the data points cannot be separated in this scenario.

For problem 7 , because the dataset mystery\_data is unlabeled. Thus we cannot separate the dataset using PCA and Laplacian. The results are shown below. We can see that for PCA , there are some clusters of data points, but because of no labels , we cannot decide which class these data points belong to. For Laplacian, the data points are more scattered , and there are also some clusters of points on the peripheral.

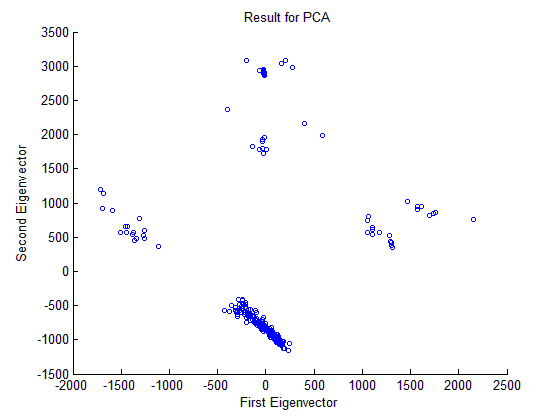


Figure. 10 PCA for Unlabeled Dataset

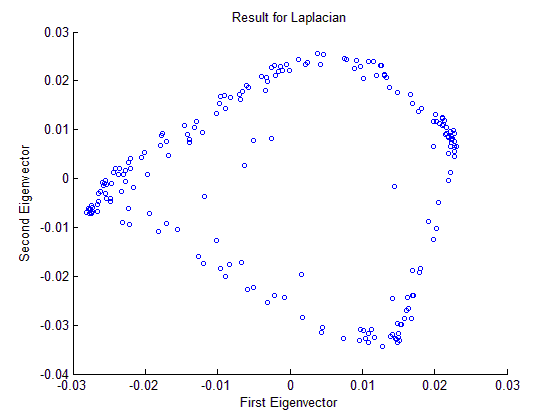


Figure. 11 Laplacian for Unlabeled Dataset

**4. Discussion**

Based on the results in the section above, we can make a discussion on the methods we have used for dimensional reduction. For problem 2 and 3, we use pca.m function to study the performance of PCA. We can find that, as the number of principal components increases, the reconstruction error decreases significantly. This is because the higher dimension projection keeps more information about the original dataset and has higher probability to reconstruct the dataset correctly. We also study the graph that shows the result of dimensional reduction for PCA. We can see that the data points of some digits are well separated such as 3, 4, 6 and 7. But the data points for other digits are mixed together.

For problem 5 and 6, we use Dimensional Reduction toolbox to implement different method including PCA, LDA ,MDS and Lapalacian. Based on the results in the last section, we find that LDA has the worst performance of these four methods. Because when reduced to 2D , most of data points are not separated with LDA. We can see that only the data points within digit 0 are well separated. This is because that LDA is a linear method which uses a linear discriminant. Thus for 2D, the data points are likely projected into a plane. For some dataset, it's difficult to apply such a linear method to separate the data points with two dimensions. We may need to reduce the dataset to a higher dimension for better performance. And for PCA and MDS, we obtain almost the same result for this dataset. This implies they both can function well with two dimensions. And for Laplacian , the parameters in the function compute\_mapping can influence the distribution of the projected data points. As the parameter k increases, the data points for different digits are getting closer. And it's getting harder to separate these data points. This is because when the value of k is small, the eigenvalues can be very small, thus the objective function for Laplacian can also get smaller. Thus the performance for dimensional reduction will become better than that when the value of k is large. Therefore for Laplacian , when k =12 , this method can achieve better performance for dimensional reduction.

For problem 7, because we use the dataset mystery\_data with no labels. Thus for PCA and Laplacian, the dimensional reduction is very different and we can just obtain some clusters of data points. And we cannot separate the data points for digits efficiently because of no labels for them. We can see that the data points for PCA are mainly separated into four clusters. However, the data points for Laplacian are more diverged. So for unsupervised learning, PCA may achieve more performance than LDA for dimensional reduction. This is because PCA can separate the data points into a larger range of area in the 2D coordinate plane.

**5. Software listing and executable software**

This program uses MATLAB including three files, p2.m, p5.m and p7.m.

For problem 2 and 3, please click and run p2.m .For problem 5 and 6, please click and run p5.m. And for problem 7, please run p7.m to get the results and graphs.