# CSCI 5521: Homework 2
# Introduction to Machine Learning

Names: Abhiraj Mohan, Saksham Goel
Emails: mohan056@umn.edu, goelx029@umn.edu

## Problem 1: (script used: `q1.m`)

For the solution to this problem, we build a MATLAB script that implements PCA on the MNIST data set. We implement the following functions to plot features from class 8 or 9 in a 2-D space using PCA. For obtaining this plot, we use `q1.m` which utilizes another file that we have submitted – `mypca.` We have implemented our own PCA algorithm in this file where we use the in-built `svd` function provided by MATLAB to calculate two principal components for data with class labels 0, 8 and 9 from our training set.

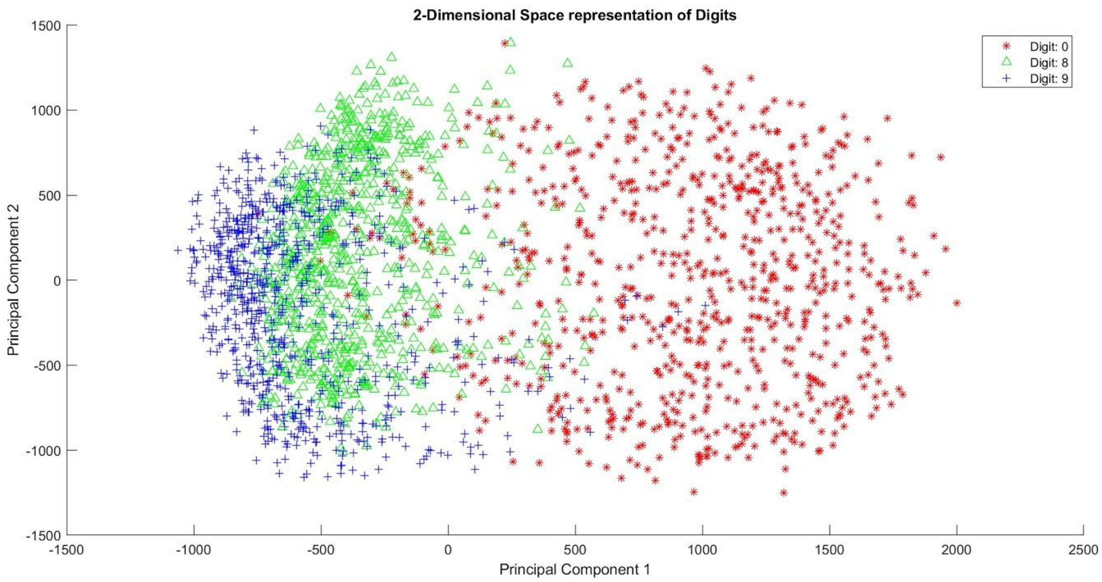


Fig 1: PCA projections for the 0, 8 and 9 classes in a 2-dimensional space

## Problem 2: (script used: `q2.m`)

For our solution to problem 2, we implement our own LDA algorithm to compute LDA projections from the PCA projections we obtain. We use mypca.m to compute principal components for our data samples with class labels the number 8 and the number 9. After projecting our data samples on the components, we find an LDA vector that maximizes the variance between the two classes and minimizes the variance within the classes itself. To complete this task, we use another file that we built - `LDA_twoclass.m`. In this file, we center the data on the origin as global mean first. After that, we calculate LDA vector from the scatter between the two classes and within the two classes.

Presented below, we have the most misclassified eight labeled data sample and the most misclassified nine labeled data sample based on the LDA projection. We know that our LDA vector lies in the $\mu 8$ - $\mu 9$ direction. Thus, we can obtain the most misclassified digit eight data sample by finding the data sample that has the minimum projection on the LDA out of all digit 8 projections and similarly, the most misclassified digit nine would be the projection of digit 9 sample which has the maximum magnitude.

Along with the figures, we also provide figures to illustrate the two-dimensional projection of our data, the LDA vectors and the classification vector. Our LDA classification vector is obtained by finding the threshold value that gives us the lowest error value for our classification. We find the vector orthogonal vector to our LDA vector and move it along the LDAprojection vector to find where we get the minimum error rate. For doing this, we iterate through all the values from the minimum to maximum values of the projection magnitude on the LDA vector and find for which threshold is the classification error on the training data the least. For the experiment performed, we find that the value for threshold we get is: **-194**

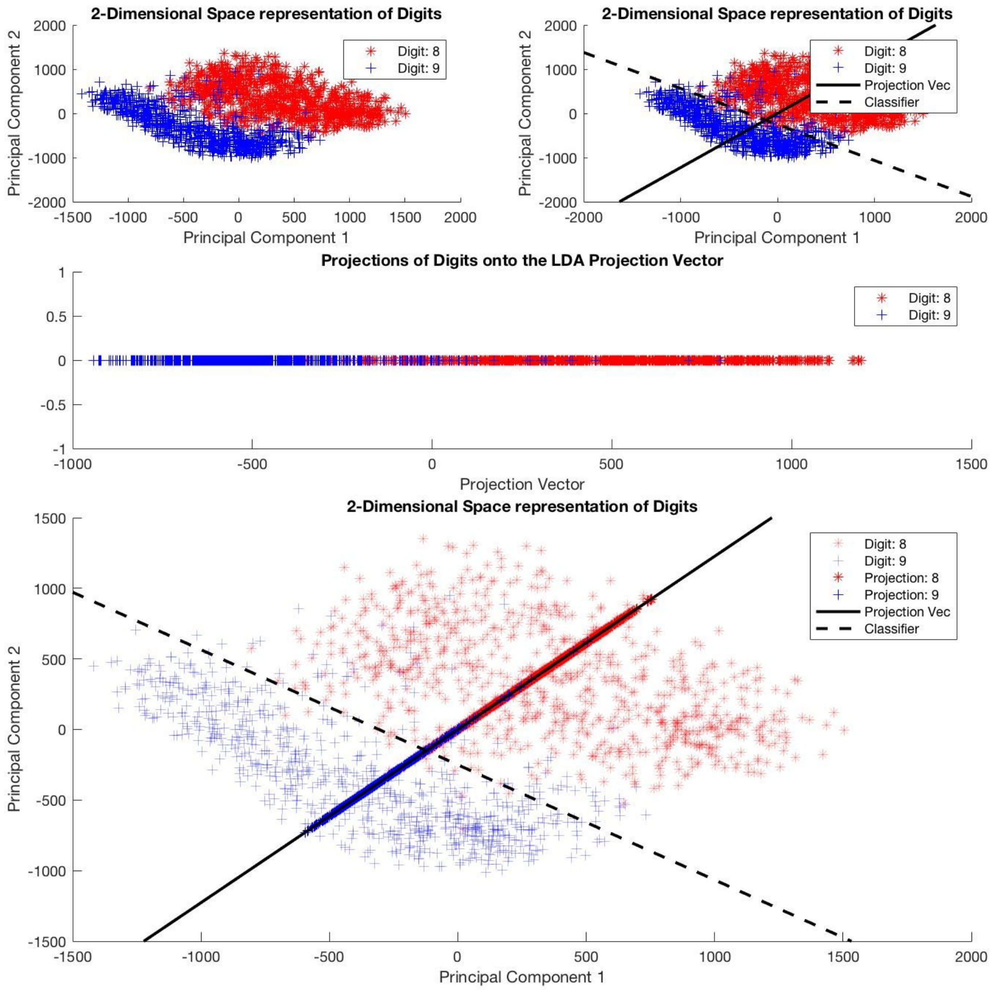


*Fig 2: LDA vector, LDA classifier and PCA projections of 8 and 9 digits in a 2-dimensional space*

Error Rate on the Training Data - **0.0512**

| **Confusion Matrix for Training Data (2 Dimension PCA Projection)** | Predicted Class - 8 | Predicted Class - 9 |
|---|---|---|
| Actual Class - 8 | 788 | 12 |
| Actual Class - 9 | 70 | 730 |

Error Rate on Test Data - **0.50**

| **Confusion Matrix for Test Data (2 Dimension PCA Projection)** | Predicted Class - 8 | Predicted Class - 9 |
|---|---|---|
| Actual Class - 8 | 113 | 87 |

| | | |
|---|---|---|
| Actual Class - 9 | 113 | 87 |

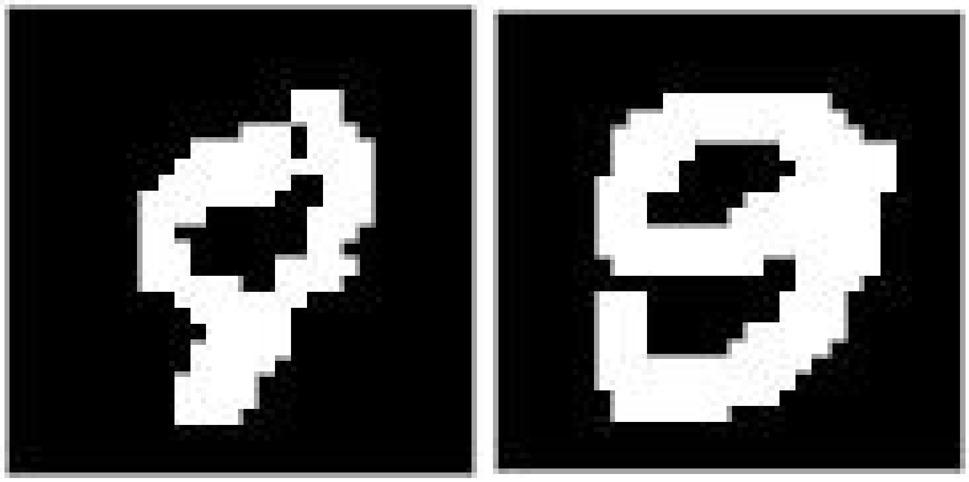

*Fig 3: Most misclassified 8 digit and 9 digit when projected on an LDA Classifier built from a 2-dimensional PCA subspace of the original MNIST Dataset*

## Problem 3: (script used: `q3.m`)

For this problem, we run our PCA algorithm iteratively with increasing number of principal components and calculate the explained variance until we reach a value that gives us a value of 0.9 for the percentage of the total variance explained by our principal components. This value we found was equal to **76**. So then we project our training data onto the 76 dimension subspace and use LDA on top of the subspace to project all the data on a 1 Dimensional Vector. Then we learn the best classifier using the same technique before, which checks at which threshold for the projection value does the training set gives the least error and use that as the threshold for computing the class labels for test data. The confusion matrix and error rate for training and test data is as follows:

| **Confusion Matrix for Training Data (76 Dimension PCA Projection)** | Predicted Class - 8 | Predicted Class - 9 |
|---|---|---|
| Actual Class - 8 | 797 | 3 |
| Actual Class - 9 | 21 | 779 |

Error Rate on Training Data - **0.0150**

| **Confusion Matrix for Test Data (76 Dimension PCA Projection)** | Predicted Class - 8 | Predicted Class - 9 |
|---|---|---|
| Actual Class - 8 | 106 | 94 |

| | | |
|---|---|---|
| Actual Class - 9 | 102 | 98 |

Error Rate on Test Data - **0.49**

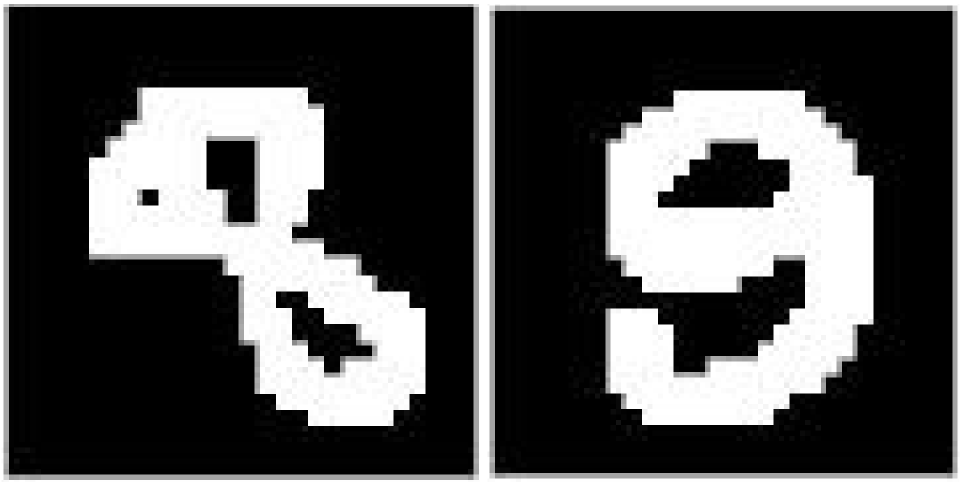

*Fig 4: Most misclassified 8 digit and 9 digit figure from an LDA Projection Classifier from a 76-dimensional PCA subspace of the original MNIST Subspace*

## Problem 4: (script used: `q4.m`)

For this part of the Homework, we are using k-NN algorithm for classification. The k-NN algorithm also known as K Nearest Neighbors algorithm. For this part, we first projected the input training data on a lower dimension such that at least 90% variance is at least preserved. For preserving the 90% variance we calculated it and found out that the number of dimensions required are 76. So for this part we first project the data for the digit 8 and digit 9 onto the 76 dimensional space. After that, we use the training data to find the best k for the k-NN by evaluating it on the training data itself. The possible values for k we considered are 1, 3, 5, 7, 9. Because we are evaluating over the training data, k = 1 gives 0 error rate, because k-NN uses training data itself to find the nearest neighbors and predict the class label. After than we use the test data and evaluate the accuracy of the k-NN classifier with k = 1. The resulting confusion matrices and error rate are as follows:

| **Confusion Matrix for Training Data (76 Dimension PCA Projection)** | Predicted Class - 8 | Predicted Class - 9 |
|---|---|---|
| Actual Class - 8 | 800 | 0 |
| Actual Class - 9 | 0 | 800 |

Error Rate on Training Data - **0.00**

| **Confusion Matrix for Test Data (76 Dimension PCA Projection)** | Predicted Class - 8 | Predicted Class - 9 |
|---|---|---|

| | | |
|---|---|---|
| Actual Class - 8 | 72 | 128 |
| Actual Class - 9 | 90 | 110 |

Error Rate on Test Data - **0.5450**

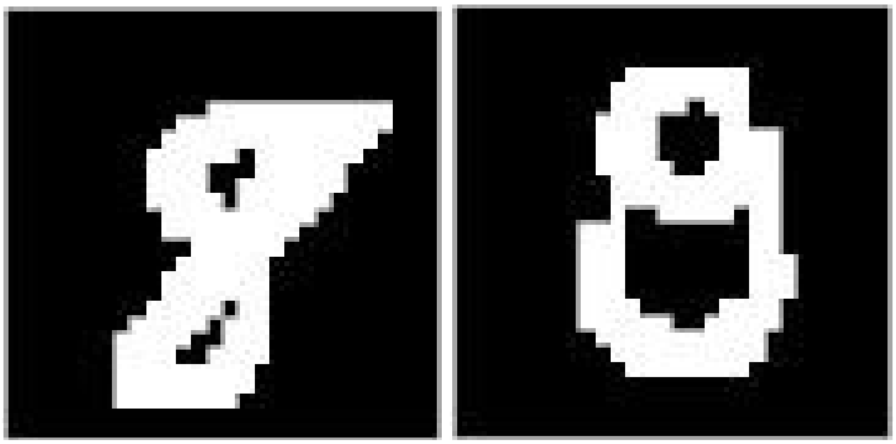

*Fig 5: Randomly chosen misclassified 8 digit and 9 digit figure from an LDA Classifier from a 76-dimensional PCA subspace of the MNIST Digit Dataset*