Duttaabhinivesh Devathi
MATH 444
Assignment 4
April 7, 2017

Problem 1:

In this problem, we are asked to develop a **PCA Classifier** for the HandWrittenDigits data used in previous reports. A PCA classifier takes a projection of the data space for each cluster of the data, projects a test data point onto the data space, and then it classifies the test data point to the cluster whose projection minimizes the norm of the difference between the projection of the data point and the data point itself. These are defined as follows:

$$P_j = U_m^{(j)} * U_m^{(j)^T} \qquad (1.1)$$

$$f(x) = argmin\{d(x, P_j x) |\ 1 \leq j \leq k\} \in \{1, 2, \ldots, k\} \qquad (1.2)$$

x in equation 1.2 is the data point in question and m in equation 1.1 is the rank of the SVD of the data set X. The distance used in this classifier (and all subsequent classifiers) is the dissimilarity difference defined in **1.3**.

$$d(x, y) = \left| 1 - \frac{x^T y}{\|x\| \|y\|} \right| \qquad (1.3)$$

The PCA classifiers were done for ranks $m = 5, 10, \ldots 80$ and the accuracy of the classification was then plotted against the rank of the SVD in **Figure 1**. The test set has 2007 different test digits.

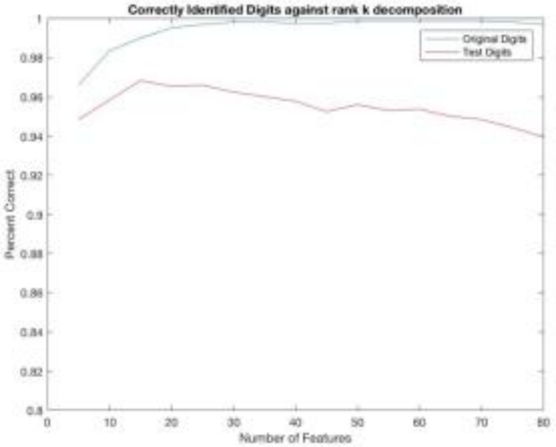

Figure 1: Line plot of the success rate of the PCA Classifier for Handwritten Digits versus rank of the SVD Decomposition.

As shown in **Figure 1**, the success rate of the original digits increases as the rank of the SVD increases up to approximately 80, which can signify that no new information is being added as the rank increases. However, the success rate of the Test Set peaks when the rank of the SVD is 15, and then slowly decreases. However, the success rate does not decrease by much as this graph is constrained by 0.8 and 1 success rate, and the decrease is very slight.

Problem 2:

This problem asked to implement the k-nearest neighbors algorithm on the same HandWrittenDigitsTestSet data against the HandWrittenDigits data. This algorithm finds the distance of each data point against the entire Training Set and finds the first k data points with the smallest distances. Then, these k data points classify the data point with respect to their cluster, but the actual classification is the majority classification of the k data points. For example, if k = 5, and 3 of them classify the test point as cluster 1, and 2 classify the test point as cluster 2, then the data point is classified as cluster 1 because a majority of the k points were classified as cluster 1. When the training data is classified against itself, and k = 1, then the success rate of the classification algorithm is 100% because the smallest distance is the point which gives a distance of 0. The distance calculation is given by **1.3**.

The question asked to implement the k-nearest neighbors algorithm for *k* ranging from 1 to 30 for both the training set and the test set. The success rates of each implementation is plotted in **Figure 2**.

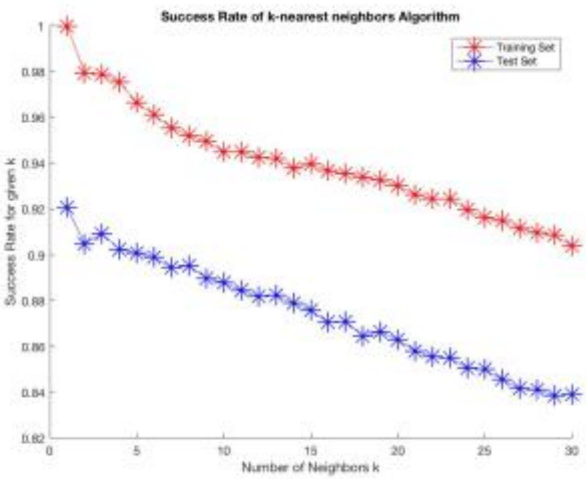

Figure 2: This is a plot of the success rate of the classification of the test set and training set of Handwritten Digits, using the k-nearest neighbors algorithm. The success rate ranges from 0.82 to 1 on this graph.

As shown in **Figure 2**, the success rate of the k-nearest neighbors algorithm for the Training Set on k=1 is 100%, which supports the hypothesis provided previously. However, it is clear that as *k* increases, the success rate of the algorithm decreases for both the Training Set and the Test Set, but not rapidly as the graph ranges from 0.82 to 1. The downfall of this algorithm however, is that it performs very slowly, as the total runtime on this machine (MacBook Pro 2.2GHz Intel Core i7) was approximately 15 minutes. Although this algorithm performs reasonably well, it does not compare to the success rates of the PCA classifier and the LDA Classifier (as will be shown ahead) given the computational time of the algorithm.

Problem 3:

This problem used a different data set, ForestSpectra (Training Set), and asked to classify ForestSpectraTest (Test Set), using an LDA Classifier. An LDA Classifier uses the orthogonal projection created by the eigen-space of the matrix $S_w^{-1}S_b$, projects the test data point onto the eigen-space and then classifies by choosing the cluster which gives the smallest distance from the centroid.

$$c_j = \frac{1}{p_j}\sum_{l=1}^{p_j} z^{(l)} \qquad (3.1)$$

$$f(x) = \ argmin\{d\left(Q^T x, c_j\right) \mid 1 \leq j \leq k\} \ \in \{1,2,\ldots,k\} \qquad (3.2)$$

First, however, we must show the clustering of the data to show whether the clustering is separated enough to classify the data well.

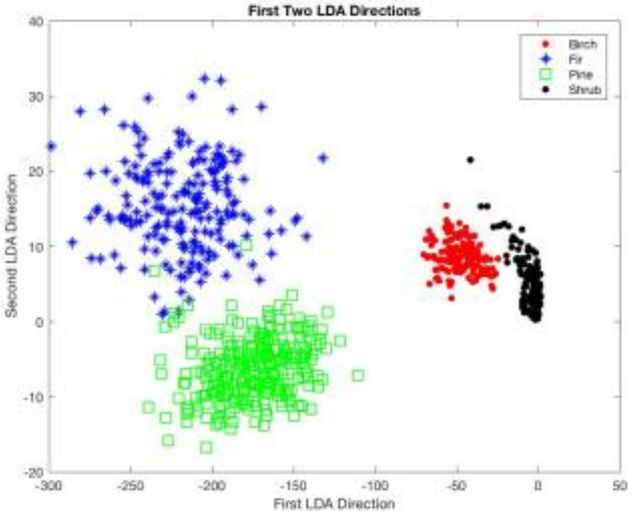

Figure 3: The first two LDA Directions of the Training Set, which shows the clusters in a vector space that maximizes the separation of the data and maximizes the coherence of each cluster.

As shown in **Figure 3**, there is a separation of clusters, but the separation is not ideal. This will cause the centroids of the clusters to be near each other, which causes some ambiguity in classification of the test set. This suggests that the clusters themselves may not be very distinct.

Using the LDA Classifier, the success rate of the classification of the Training Set is approximately 92%, while the classification of the Test Set is 88%. These values are stored in the variable **ratio_correct** in **problem3.m** of the code.