REPORT: SMAI Assignment 1

By: Rashika Kheria
201101101

Problem Statement:

Code a complete digit recognizer and test it on the MNIST digit dataset. Recognizer should read the image data, extract features from it and use a k-nearest neighbor classifier to recognize any test image.

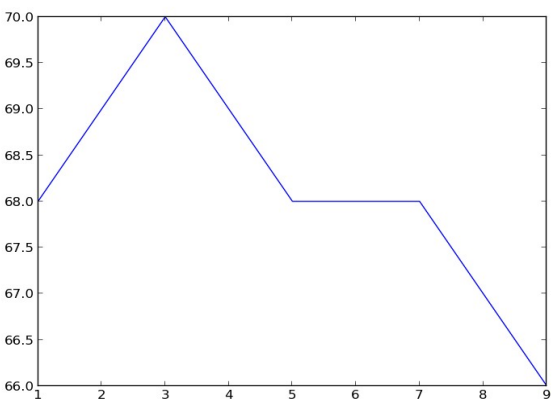
Solution:

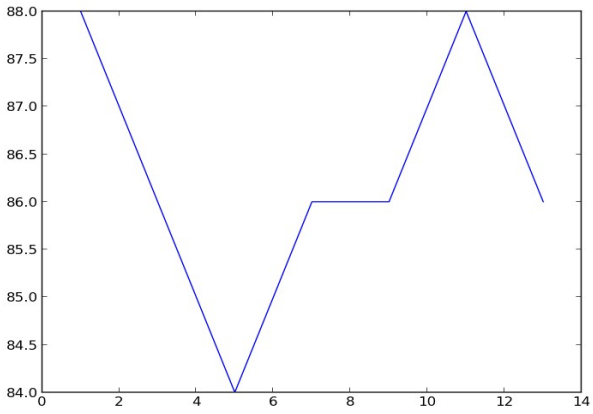
Features were extracted from each training image and a classifier model using k-NN algorithm was constructed to check every test image and label was assigned to that particular image. I coded k-NN algorithm on the following features:

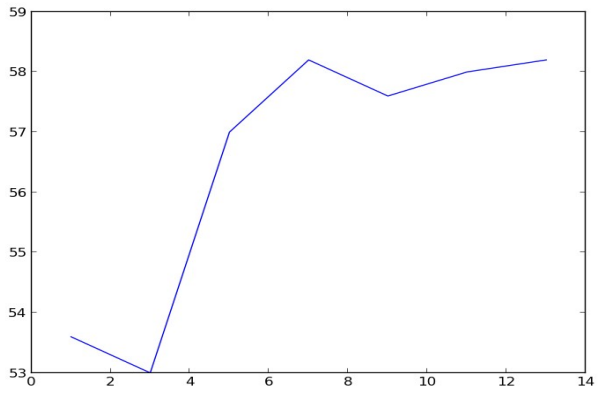
- *Counting number of 1's in each row(Feature 1):* I constructed a feature vector of 1 X 28 dimensions storing the total count of one's in each row.
- *Pixel Difference(Feature 2):* I constructed a feature vector of 1 X 784 dimensions storing the pixel value of the given image.
- *Concavities Measurement(Feature 3):* I divided the image into 4 zones each one obtaining its own feature vector of 5 dimension. This feature calculated the number of white pixels which encountered 0, 1, 2, 3 or 4 back pixels when moved in 4 directions. Thus finally a 20 X 1 dimension vector was created.
- *Zoning(Feature 4):* I divided the image into 7 X 7 matrices and calculated ((the total number of ones in each zone)/area of matrix)*100. Therefore a matrix of 1 X 49 dimension was created.
- *Orientation(Feature 5):* I calculated number of ones in every direction (seperated by angle 45 degree) and it resulted into a vector of 1 X 8 vector.
- *Multi Zoning(Feature 6):* I divided the given 28 X 28 matrix into 2 X 2, 4 X 4, 7 X 7 and 14 X 14 matrices and applied zoning method on eack subpart. It resulted into the formation of 1 X 265 dimension vector.

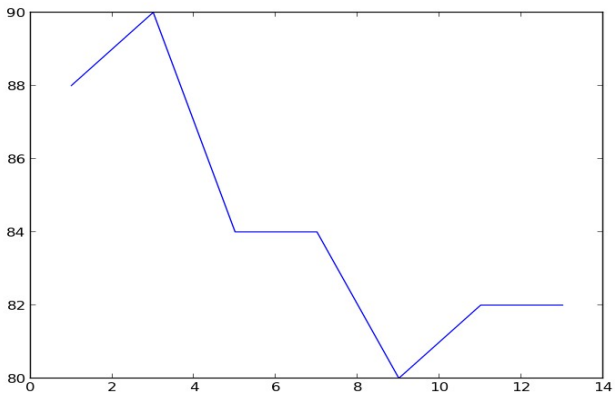
Result:

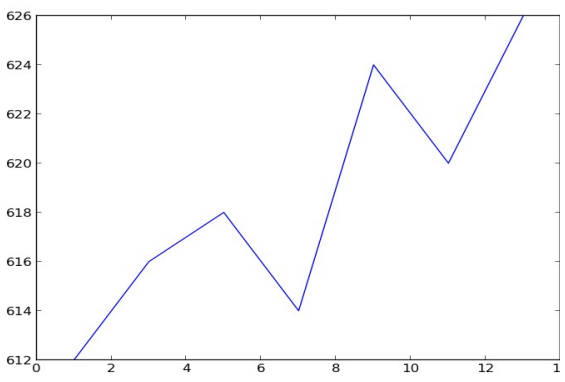
Variation of Accuracy with value of k (The first graph is for Feature 1, second for Feature 2 and so on):

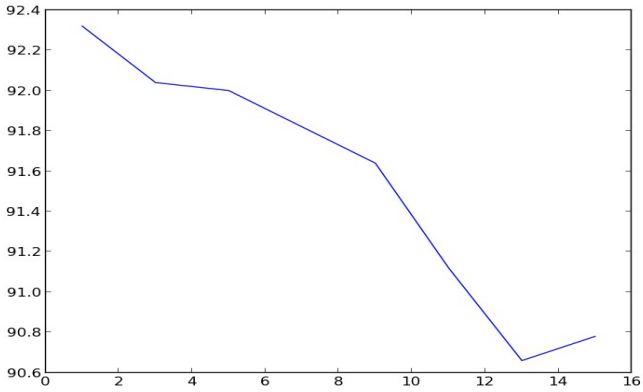












Variation of Performance with Training Data Size (Testing Data Size = 5000 and k = 5):

Training Data Size	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6
5,000	63.06%	90.67%	65.58%	88.01%	70.32%	91.52%
10,000	65.58%	92.54%	68.00%	89.93%	71.5%	92.0%
15,000	66.36%	Takes long time	68.35%	90.50%	72.36%	92.74%
20,000	66.54%	Takes long time	68.98%	90.55%	72.55%	93.44%
25,000	70.74%	Takes long time	69.55%	91.32%	73.12%	93.82%

Variation of Error Rate with Different Features and Training Sizes with mean value and standard deviation value in Error Rate (Testing Data Size = 5000 and k = 5):

Training Data Size	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6
5,000	36.94%	9.33%	34.42%	11.99%	29.68%	8.48%
10,000	34.42%	7.46%	32.00%	10.07%	28.5%	8%
20,000	33.64%	Takes long time	31.65%	9.50%	27.64%	7.26%
30,000	33.46%	Takes long time	31.02%	9.45%	27.45%	6.56%
40,000	29.26%	Takes long time	30.45%	8.68%	26.88%	6.18%
Mean in Error Rate	33.54%		31.91%	9.94%	28.03%	7.29%
SD in Error Rate	2.47%		1.36%	1.17%	0.98%	0.86%

Thus, the best feature is ***Multi-zoning*** with k = 1. The pixel difference feature is also providing good accuracy but since the code takes too much time to execute thus with respect to time Multi Zoning is the best feature.

The ***confusion matrix*** for ***Multi-zoning feature*** is (Train Size: 2500 and Test Size: 500):

	0	1	2	3	4	5	6	7	8	9
0	42	0	0	0	0	0	0	0	0	0
1	0	67	0	0	0	0	0	0	0	0
2	1	6	39	1	1	0	1	4	2	0
3	0	1	1	39	0	1	1	1	0	1
4	0	2	0	0	49	0	1	0	0	3
5	1	1	0	1	2	43	1	0	0	1
6	3	0	0	0	1	1	38	0	0	0
7	0	2	0	0	1	0	0	44	0	2
8	1	0	0	2	1	1	1	2	32	0
9	0	1	0	0	3	0	0	3	2	45

Failure Example with Reasoning: (Concavity feature): In this feature we check for the portion of the image encircled on all 4 sides, 3 sides, 2 sides, 1 sides or 0 side. Therefore, in this feature 4 and 9 get more or less similar feature vector. Hence 9 is classified as 4 in many cases.

Time required to run codes are:
a) Time Complexity : O(mn) where m = training data size and n = testing data size
b) Space Complexity : O(m|v|) where m = training data size and |v| is the size of the feature vector.

Assumptions for the Executable: The input matrix image is given as space seperated 784 numbers ending with \n.