Vibhor Jain
3/3/13
CSE 176

# Programming Assignment 2: Report

The program implements the k-nearest-neighbor algorithm using weighted and un-weighted averages (with Euclidean and angular distances). It takes in training and a testing pattern set and makes output data of the testing pattern set (based on the training pattern set). This output data is the program's predictions of the test pattern set's output vectors, which is compared to the set's actual output vectors.

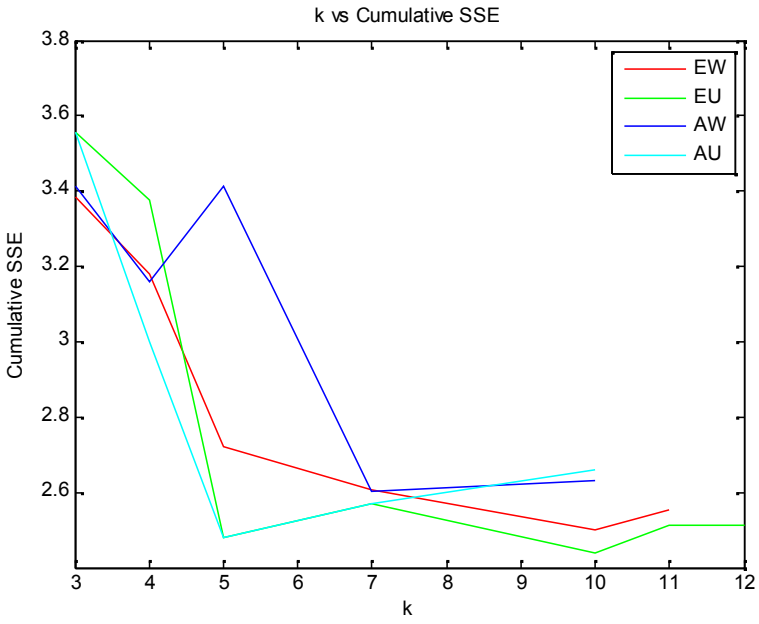
The program was implemented in MATLAB. The main file to be run is 'knn_driver.m', which makes the function calls to 'knn_algo.m', 'sortMethod.m', 'computeEuclidDist.m', 'computeAngularDist.m', and 'quickSort.m'. As with any MATLAB program, the user runs the main file on the MATLAB console window (type 'knn_driver') to compile and run the entire program. These files must be in the same directory (along with the configuration file (.cfg), training set file, and test set file).

In addition to the tools used in the first programming assignments, this program makes use of functions such as find() and sum(), which help to calculate averages and SSE. This data is written to a text file (.log). A trace of the program running is given below:

```
>> knn_driver
Enter configuration file name (in this directory)
glass-knn.cfg
>> diary off;
```

Appropriate modifications are made to the configurations file and the results are recorded in the output text file (.log file). Note: 'diary off;' was executed in order to stop recording the sample trace of the program.

Many experiments were run on this program by changing the k parameter (3, 4, 5, 7, 10, 11, 12), distance metric (Euclidean or Angular), and averaging scheme (weighted or un-weighted) in the configuration file. These parameter configurations were assessed based on the cumulative sum squared error of the test output vectors. The graph of the results of these experiments is given below:




The term 'EW' represents the experiments run using Euclidean distance and the weighted sum of output vectors, 'EU' represents the experiments using Euclidean distance with un-weighted sum of the output vectors, 'AW' represents the experiments using the Angular distance with the weighted sum of the output vectors, and 'AU' represents the experiments using the Angular distance with un-weighted sum of the output vectors.

According to the graph, we pick the optimal k for which we get the least cumulative SSE.
The best distance measure seems to be Euclidean distance, as it seems to produce less noise in the cumulative SSE (there are less local minima). We see the error for Euclidean distance decaying more consistently than the error for angular distance. This method seems to generate good results (low cumulative SSE) using higher k values than the alternative. Based on the best distance metric, we pick our optimal k to be 10.

I learned through this exercise that the k-nearest-neighbor algorithm is as good as the quality of the training data set. Noise from the training dataset can have an adverse impact on the algorithm, although the weights for each vector can nullify some of these effects. The performance of this algorithm would be largely based on how effectively we can store and work with our feature vectors in our datasets.

Acknowledgements:

Mitchell, Tom M. *Machine Learning*. New York: Mc-Graw Hill, 1997. Print.

"Sorting Algorithms/Quicksort." *Rosetta Code*. N.p., n.d. Web. 13 Feb. 2013.