Homework 5, ECE 6254, Spring 2014

Due: Wednesday March 26, at the beginning of class.

This assignment is optional. Should you choose to complete the assignment it will replace your lowest homework score.

Problem:

1. **Nearest Neighbor Classifiers**

   In this problem we will use the same file as in the last assignment (`mnist_49_3000.mat`), but now we will try classifying this data using various versions of the nearest neighbor classifier.

   As with last time, you will use the first 2000 data points to design the classifier, reserving the last 1000 to compute the test error. Rather than the tuning parameter C, you will now need to set the number k of nearest neighbors to use. You can again do this via a holdout set: treat the first 1000 data points as your training data, and estimate the error (called the "holdout error") of a k-nearest neighbor classifier using the other 1000 data points. Select the k having the lowest holdout error. *Note:* As before, once you have selected k, you should essentially ignore the holdout set when evaluating the performance of the resulting algorithm on the final 1000 points (the testing set).

   (a) Implement your own version of the k-nearest neighbor classifier. Calculate the nearest neighbors using the Euclidean distance. The MATLAB command `pdist2` is extremely helpful. Report the best performing value of k and the resulting error on the test set. Turn in your code.

   (b) Repeat the above experiment, but rather than using the Euclidean distance, use the ℓ_1 norm, i.e., $\|\mathbf{x} - \mathbf{x}_i\|_1 = \sum_j |x(j) - x_i(j)|$. Again report the value of k that you select and the resulting error on the test set.

   (c) Another variance of nearest neighbor classifiers is to consider a *weighted* vote among the nearest neighbors. For example, in binary classification ($y_i = \pm 1$) the resulting label could be determined via

   $$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{i \in I_k(\mathbf{x})} w_i(\mathbf{x}) y_i \right),$$

   where $I_k(\mathbf{x})$ denotes the k-nearest neighbors of $\mathbf{x}$ and $w_i(\mathbf{x})$ denote a set of weights. An example of the weights would be something like $w_i(\mathbf{x}) = 1/\|\mathbf{x} - \mathbf{x}_i\|$.

   Implement a weighted version of the k-nearest neighbor classifier. Experiment with different weighting schemes and report on your results.