Stony Brook University
Department of Computer Science
CSE 353: Machine Learning
(Spring 2014)
Instructor: Luis E. Ortiz

Programming Homework 1: Nearest Neighbors

Out: *Thu Apr 3,* **Due**: *Sat Apr 12 at midnight (11:59pm)*

Recognizing Handwritten Digits Automatically

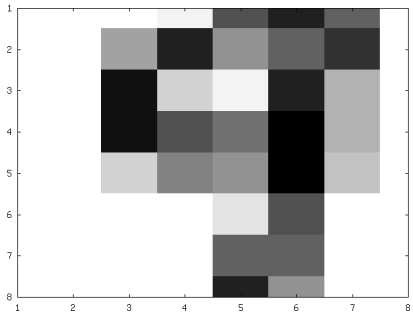
In this homework you will apply the nearest-neighbor machine-learning technique to a stylized version of the problem of optical recognition of handwritten digits.

The data set of examples was created by pre-processing 32x32 bitmap images of hanwritten digits (i.e., each image was originally represented as a 32x32 matrix of pixels, each pixel taking value 1 or 0 corresponding to a black or white pixel, respectively). The result of the pre-processing is an 8x8 "grayscale" image where each pixel takes an integer value from 0 to 16. [1]

The following is an example for the digit 9:

Original Image

Transformed Image



For each preprocessed image, each pixel corresponds to an attribute taking one of 16 values. Thus, the data has 64 attributes. Although each attribute is an integer from 0 to 16, treat each attribute as real-valued. There are 10 classes corresponding to each of digit.

You are provided five (5) data files.

- `optdigits_train.dat` contains (a permuted version of) the original training data.
- `optdigits_train_trans.dat` contains the transformed image version of the examples in the file `optdigits_train.dat`.
- `optdigits_test_trans.dat` contains the transformed image version of the test examples.
- `optdigits_trial.dat` contains an example of each digit from the original validation set.
- `optdigits_trial_trans.dat` contains the transformed image version of the examples from `optdigits_trial.dat`.

Each file is composed of one data example per line. Each line contains 65 integers separated by a single empty space. The first 64 integers correspond to the values of each of the attributes (i.e.,

[1] Pre-processing is done to "normalize" the data to help correct for small distortions and reduce dimensionality. The resulting images provide a good approximation of the original images for classification purposes. Please visit `http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits` for more information about this data set.

a number from 0 to 16) and the last integer is the example class label (i.e., the corresponding digit $0, 1, \ldots, 9$). The training and test data sets have 3823 and 1797 examples, respectively.

You are asked to implement the technique of k-nearest neighbors classification using Euclidean distance as the distance metric and apply it, using $k = 1$ and $k = 3$ to the data set of optical handwritten digits described above. *As a tie-breaking rule during classification, select the lowest digit as the class label (i.e., if there is a tie between* 3 *and* 7*, pick* 3 *as the label).*

Applying 1-Nearest Neighbors

Here, you are asked to evaluate the performance of a 1-nearest neighbor classifier by producing its *learning graph*, which is a *plot of the test error as a function of the number of training examples* (i.e., the size of the training set). In particular, compute and display/plot the learning graph for training sets consisting of the *first* $m = 10, 50, 100, 500, 1000, 3823$(all) examples in the `optdigits_train_trans.dat` file. Recall that the *test error* is the misclassification error obtained from evaluating the classifier on the *test data*. The file `optdigits_test_trans.dat` contains the test data, which in this case is the transformed version of the images.

Applying 3-Nearest Neighbors

For each image in the trial set, provide the indexes to the 3-nearest neighbor examples in the training dataset. The index to an example is the row, or line number, of the file with the training dataset.

1. For each transformed image example in the trial data set given in the `optdigits_trial_trans.dat` file, identify the *indexes to the* 3*-nearest neighbors* in the training dataset *in the transformed space*, using the training data file of transformed images given in `optdigits_train_trans.dat`.

2. List, *in increasing order of Euclidean distance*, the *indexes* to the 3-nearest neighbors you identified for each of the 10 exemplars trial dataset, *include their respective labels.* Also provide the *output labels of the corresponding* 3*-nearest-neighbors classifier* for each of the trial examples.*How many* of the trial examples are correctly classified by the 3-nearest-neighbors classifier?

3. *(Optional)* Display the corresponding *original 32x32 pixels, B&W images*, which you can find in the `optdigits_train.dat` file, of the 3-nearest neighbors of each of the 10 exemplars in the trial data set, along with their respective labels. Display the images as a row, starting with the exemplar itself (whose original image you get from the file `optdigits_trial.dat`), and followed by the original images for the 3-nearest neighbors, *in increasing order of Euclidean distance.*

   NOTE: *The first* 1024 *binary values of each line in the files* `optdigits_train.dat` *and* `optdigits_trial.dat` *encode the original bitmap images as bit vectors. You need to appropriately reshape that vector to obtain the original bitmap image in matrix form.*

What to Turn In

You need to submit the following (electronically via Blackboard):

1. A **written report** (*in PDF*) of all your results (including plots, list of indexes, ordered by Euclidean distance, classifier's output labels, answer to question, and, for the optional part, a grid of images, properly formatted per the layout instructions given above), along with a brief discussion.

2. All your **code and executable** (as a tared-and-gziped compressed file), with instructions on how to run your program. A platform-independent executable is preferred; otherwise, also provide instructions on how to compile your program. Please use standard tools/compilers/etc. generally available in most popular platforms.

Collaboration Policy: *It is OK to discuss the homework with your peers, but each student must write and turn in his/her own report, code, etc. based on his/her own work.*