CSE 151: Introduction to Machine Learning Spring 2015

Problem Set 2

Instructor: Kamalika Chaudhuri **Due on:** April 24

Instructions

- This is a 40 point homework.
- Homeworks will graded based on content and clarity. Please show your work *clearly* for full credit.
- For Problem 3, you are free to use any programming language that you wish. Please email a copy of your code to cse151homeworks@gmail.com.

Problem 1 (14 points)

In class, we mentioned that for some applications, it may make sense to do k-nearest neighbors with respect to a distance other than the usual Euclidean distance. In this problem, we will look at the k-nearest neighbor problem when the distance between the points is the following modified form of the Euclidean distance. Given two vectors $x = (x_1, x_2)$ and $z = (z_1, z_2)$, the modified distance measure $d_M(x, z)$ is defined as:

$$d_M(x, z) = \sqrt{\frac{1}{2}(x_1 - z_1)^2 + (x_2 - z_2)^2}$$

1. Consider the following labelled training dataset:

   $$((0, 0), 1), ((2, 2), 2), ((4, 0), 3)$$

   First, consider the 1-nearest neighbor classifier on these points with respect to the usual Euclidean distance, and draw the decision boundary for this classifier. Write down the equations for the different sections (or segments) of the decision boundary. Clearly mark each region in your drawing with the label assigned by the classifier to a test example in this region.

2. Now, consider the 1-nearest neighbor classifier on labelled datasets in part (1) with respect to the modified Euclidean distance d_M. In a separate figure, draw the decision boundary for this classifier. Again, write down the equations for the different segments of the decision boundary, and clearly mark each region in your drawing with the label assigned by the classifier to a test example in this region.

3. Repeat parts (1) and (2) (namely, drawing the decision boundary with respect to the Euclidean distance and the modified Euclidean distance d_M) for the following labelled training dataset:

   $$((0, 0), 1), ((1, 1), 1), ((-1, 1), 2)$$

Problem 2 (6 points)

In class, we talked about how k-nearest neighbor classifiers are robust to errors or noise in the data when $k > 1$. In this problem, we will look more closely at how exactly this error correction happens.

Suppose that we have two labels 0 and 1. Suppose we are given any k, and any test point x; let $z_1, \ldots, z_k$ be the k closest neighbors of x in the training data. For the rest of the question, we make the assumption that for all $i = 1, \ldots, k$, the probability that the label of z_i is not equal to the label of x is $p = 0.1$; moreover, for $i \neq j$, the events that the label of z_i is not equal to the label of x and the label of z_j is not equal to the label of x are independent. In reality of course, this assumption will not hold for very large k, but for small k, this is a fairly reasonable assumption to make.

1. What is the probability that the 1-nearest neighbor classifier makes a mistake on x?

2. Now calculate the probability that 3-nearest neighbor classifier and the 5-nearest neighbor classifier make a mistake on x. What can you conclude from these calculations about the robustness of these classifiers?

Problem 3 (20 points)

In this problem, we look at the task of classifying images of digits using k-nearest neighbor classification. Download the files `hw2train.txt`, `hw2validate.txt` and `hw2test.txt` from the class website. These files contain your training, validation and test data sets respectively.

For your benefit, we have already converted the images into vectors of pixel colors. The data files are in ASCII text format, and each line of the files contains a feature vector of size 784, followed by its label. The coordinates of the feature vector are separated by spaces.

1. For $k = 1$, 3, 5, 11, 16, and 21, build k-nearest neighbor classifiers from the training data. For each of these values of k, write down a table of training errors (error on the training data) and the validation errors (error on the validation data). Which of these classifiers performs the best on validation data? What is the test error of this classifier?

2. For $k = 3$, construct a 3-nearest neighbor classifier based on the data in `hw2train.txt`. Compute the confusion matrix of the classifier based on the data in `hw2test.txt`. The confusion matrix is a 10×10 matrix, where each row is labelled $0, \ldots, 9$ and each column is labelled $0, \ldots, 9$. The entry of the matrix at row i and column j is C_{ij}/N_j where C_{ij} is the number of test examples that have label j but are classified as label i by the classifier, and N_j is the number of test examples that have label j.

   Based on your calculation of the confusion matrix, what are i and j in the following statements:

   (a) The 3-NN classifier has the highest accuracy for examples that belong to class i.

   (b) The 3-NN classifier has the least accuracy for examples that belong to class i.

   (c) The 3-NN classifier most often mistakenly classifies an example in class j as belonging to class i.

   Based on your answers, which digits do you think are the easiest and the hardest to classify?

3. (Not for credit) To understand which examples are hard to classify, look at the examples with 2 that the 3-nearest neighbor algorithm classified as 8. How many such test examples are there? What are the labels of the 3 points in the training data that are closest to these test examples?

   Convert the vectors for these test examples back to 28×28 images. Looking at these images, how difficult do you think are they to classify?