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 submit a printout of your code along with your homework.

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 the points in part (1) with respect to the modified Euclidean distance d_M, and 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:

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

Solutions

1. Finding the decision boundaries is essentially figuring out which test points have each training point as their nearest neighbor. Taking the training point $(0, 0)$ as an example. The test points with $(1, 1)$ as the nearest neighbor are exactly the intersection of solutions for the inequalities: $d(x, (0, 0)) \leq d(x, (2, 2))$ and $d(x, (0, 0)) \leq d(x, (4, 0))$.

   Using the regular Euclidean distance, these inequalities can be simplified to linear inequalities. $d(x, (0, 0)) \leq d(x, (2, 2))$ gives

   $$\begin{aligned}(x_1 - 0)^2 + (x_2 - 0)^2 &\leq (x_1 - 2)^2 + (x_2 - 2)^2 \\ 0.5x_1 + 0.5x_2 - 1 &\leq 0\end{aligned}$$

The solutions to this linear inequality form a half-plane, with the boundary line given by $0.5x_1 + 0.5x_2 - 1 = 0$. This is the equidistance line between $(0, 0)$ and $(2, 2)$.

Similarly, $d(x, (0, 0)) \leq d(x, (4, 0))$ gives the half-plane

$$(x_1 - 0)^2 + (x_2 - 0)^2 \leq (x_1 - 4)^2 + (x_2 - 0)^2$$
$$0.5x_1 - 1 \leq 0.$$

The intersection of these two half-planes is the region for which $(0, 0)$ is the nearest neighbor.

Similarly, $d(x, (2, 2)) \leq d(x, (4, 0))$ gives the half-plane

$$(x_1 - 2)^2 + (x_2 - 2)^2 \leq (x_1 - 4)^2 + (x_2 - 0)^2$$
$$0.5x_1 - 0.5x_2 - 1 \leq 0.$$

The intersection of these two half-planes is the region for which $(2, 2)$ is the nearest neighbor.

To do this efficiently for all training points, we simply solve each of $d(x, (0, 0)) = d(x, (2, 2))$, $d(x, (0, 0)) = d(x, (4, 0))$ and $d(x, (2, 2)) = d(x, (4, 0))$. This gives the boundary lines, and thus the half-planes. The intersection of the proper half-planes gives us the "cell" corresponding to each training point, and also the decision boundaries.

The decision boundaries are shown in Figure 1.

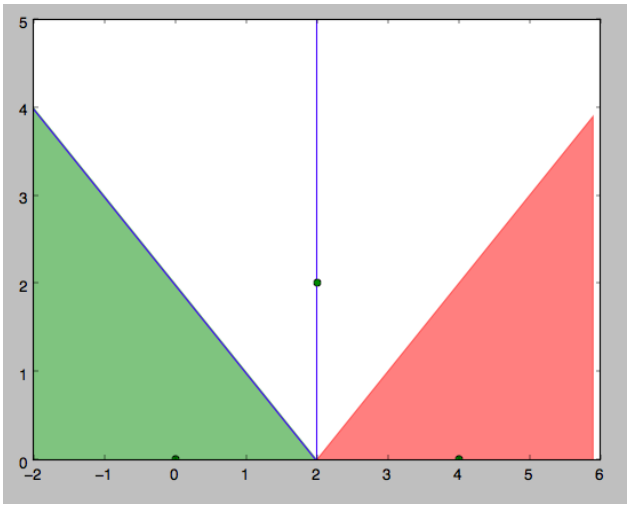

Figure 1: Decision boundary using the regular Euclidean distance. Green, white and red spaces show the area for which (0,0), (2,2) and (4,0) are respectively the nearest neighbours.

2. Repeat the procedure above with the modified Euclidean distance. For example, $d_M(x, (0, 0)) = d_M(x, (2, 2))$ gives the boundary line,

$$\frac{1}{2}(x_1 - 0)^2 + (x_2 - 0)^2 = \frac{1}{2}(x_1 - 2)^2 + (x_2 - 2)^2$$
$$x_1 + 2x_2 - 3 = 0$$

$d_M(x, (0, 0)) = d_M(x, (4, 0))$ gives the boundary line,

$$\frac{1}{2}(x_1 - 0)^2 + (x_2 - 0)^2 = \frac{1}{2}(x_1 - 4)^2 + (x_2 - 0)^2$$
$$x_1 - 2 = 0$$

$d_M(x,(2,2)) = d_M(x,(4,0))$ gives the boundary line,

$$\frac{1}{2}(x_1-2)^2+(x_2-2)^2 = \frac{1}{2}(x_1-4)^2+(x_2-0)^2$$
$$x_1 - 2x_2 - 1 = 0$$

After all boundary lines are computed, the region associated with each training point and the decision boundaries can be obtained.

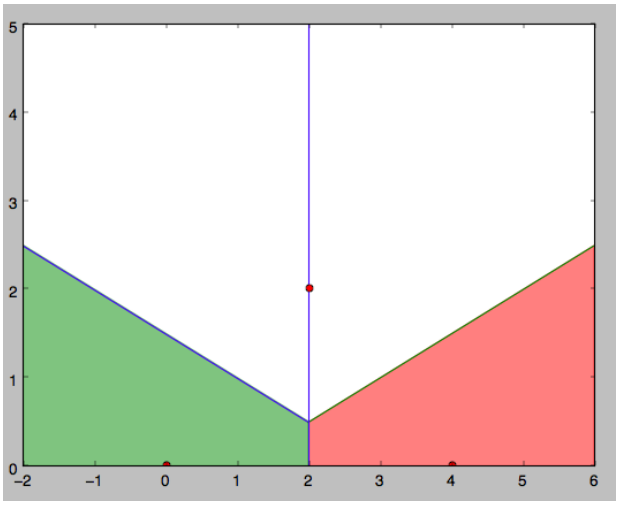

Figure 2: Decision boundary using the modified Euclidean distance. Green, white and red spaces show the area for which (0,0), (2,2) and (4,0) are respectively the nearest neighbours.

An alternative method can be found by writing the modified Euclidean distance as,

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

This is the usual Euclidean distance on a space with the x-axis scaled by $1/\sqrt{2}$. Therefore a procedure to find the decision boundaries using this distance function is:

Step 1: converting a point $x = (x_1, x_2)$ to $x' = \left(\frac{x_1}{\sqrt{2}}, x_2\right)$, i.e. scaling the x-axis by $1/\sqrt{2}$,

Step 2: drawing the decision boundaries on this x-scaled space, using the usual Euclidean distance,

Step 3: scaling the x-axis back to the original scale.

3. **Note that there are only two labels in this question! (not three, like previous problem)**
The decision boundaries using the regular Euclidean distance are shown in Figure 3.

$d_M(x,(0,0)) = d_M(x,(1,1))$ gives the boundary line,

$$(x_1-0)^2+(x_2-0)^2 = (x_1-1)^2+(x_2-1)^2$$
$$x_1 + x_2 - 1 = 0$$

$d_M(x,(0,0)) = d_M(x,(-1,1))$ gives the boundary line,

$$(x_1-0)^2+(x_2-0)^2 = (x_1+1)^2+(x_2-1)^2$$
$$x_1 - x_2 + 1 = 0$$

$d_M(x,(1,1)) = d_M(x,(-1,1))$ gives the boundary line,

$$(x_1-1)^2+(x_2-1)^2 = (x_1+1)^2+(x_2-1)^2$$
$$x_1 = 0$$

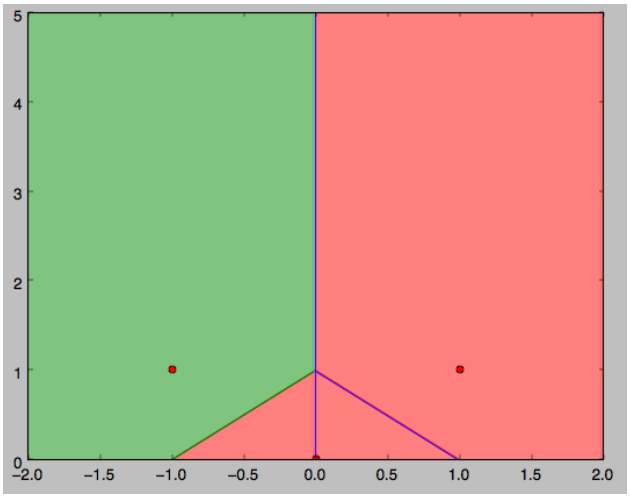

Figure 3: Decision boundary using the regular Euclidean distance. Red and green spaces show the area for which the labels are 1 and 2 respectively. The lines show the NN-boundary between the three points.

The decision boundaries using the modified Euclidean distance are shown in Figure 4.
$d_M(x,(0,0)) = d_M(x,(1,1))$ gives the boundary line,

$$\frac{1}{2}(x_1-0)^2+(x_2-0)^2 = \frac{1}{2}(x_1-1)^2+(x_2-1)^2$$
$$x_1+2x_2-1.5 = 0$$

$d_M(x,(0,0)) = d_M(x,(-1,1))$ gives the boundary line,

$$\frac{1}{2}(x_1-0)^2+(x_2-0)^2 = \frac{1}{2}(x_1+1)^2+(x_2-1)^2$$
$$x_1-2x_2+1.5 = 0$$

$d_M(x,(1,1)) = d_M(x,(-1,1))$ gives the boundary line,

$$\frac{1}{2}(x_1-1)^2+(x_2-1)^2 = \frac{1}{2}(x_1+1)^2+(x_2-1)^2$$
$$x_1 = 0$$

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 $p = 0.1$. (In reality, this assumption will not hold for large k, but for small k, this is not a bad assumption to make.)

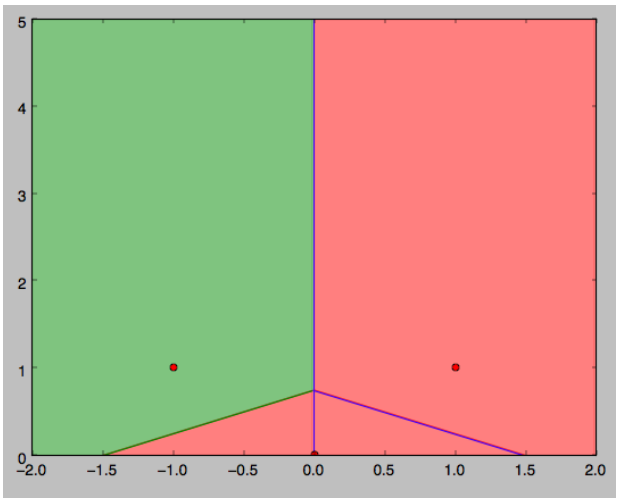

Figure 4: Decision boundary using the modified Euclidean distance. Red and green spaces show the area for which the labels are 1 and 2 respectively. The lines show the NN-boundary between the three points.

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?

Solutions

For a test point x, a k-nearest neighbor classifier will give the incorrect label if and only if the majority (formally, at least $\lceil k/2 \rceil$, where $\lceil k/2 \rceil$ means the smallest integer larger than $k/2$) of x's k closest neighbors, $z_1, \cdots, z_k$, have the incorrect label. Since each of these k neighbors has the same probability p = 0.1 of having an incorrect label, the probability that exactly n of the k neighbors have the incorrect label, $P\{\text{exactly } n \text{ neighbors incorrect}\} = \binom{k}{n} p^n (1-p)^{k-n}$.

The probability that at least $\lceil k/2 \rceil$ neighbors have the incorrect label, or in other words, the probability that a k-NN classifier makes a mistake is,

$$P\{k\text{-NN mistake}\} = \sum_{n=\lceil k/2 \rceil}^{k} P\{\text{exactly } n \text{ neighbors incorrect}\} = \sum_{n=\lceil k/2 \rceil}^{k} \binom{k}{n} p^n (1-p)^{k-n}.$$

1. $P\{1\text{−NN mistake}\} = P\{\text{exactly 1 neighbors incorrect}\} = 0.1$
2. $P\{3\text{-NN mistake}\}$
   $= P\{\text{exactly 3 neighbors incorrect}\} + P\{\text{exactly 2 neighbors incorrect}\}$
   $= 0.1^3 + \binom{3}{2} 0.1^2 \times 0.9 = 0.028$

   $P\{5\text{-NN mistake}\}$
   $= P\{\text{exactly 5 neighbors incorrect}\} + P\{\text{exactly 4 neighbors incorrect}\} + P\{\text{exactly 3 neighbors incorrect}\}$
   $= 0.1^5 + \binom{5}{4} 0.1^4 \times 0.9 + \binom{5}{3} 0.1^3 \times 0.9^2 = 0.00856$

   Therefore, 5-NN classfier is more robust than both 1-NN and 3-NN 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, 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?

Solutions

1. The error values are:

k	Training Error	Validation Error
1	0	0.1267
3	0.0630	0.1367
5	0.0830	0.1467
11	0.1120	0.1767
16	0.1390	0.1900
21	0.1570	0.2033

   Based on the validation error, the best classifier is the 1-NN classifier. Its error computed on the test data is 0.0967.

2. For the 3-NN classifier, the confusion matrix is:

Label	0	1	2	3	4	5	6	7	8	9
0	0.89	0	0	0.033	0	0	0	0	0	0
1	0	1	0.053	0.1	0	0	0.027	0	0	0
2	0	0	0.89	0	0	0	0	0	0	0.037
3	0	0	0	0.67	0	0.077	0	0	0	0
4	0	0	0	0	0.93	0	0	0	0	0.037
5	0	0	0	0.033	0	0.81	0	0	0.036	0
6	0.11	0	0.033	0	0	0	0.95	0	0.036	0
7	0	0	0	0.033	0	0.038	0.027	0.93	0	0
8	0	0	0.026	0.067	0	0	0	0	0.93	0
9	0	0	0	0.067	0.071	0.077	0	0.067	0	0.93

(a) The (i, i)-th entry of the confusion matrix is the fraction of examples belonging to class i which get (correctly) classified as class i. Thus, for this problem, (i, i) is the index of the maximum diagonal entry. From the matrix, $i = 1$.

(b) For this problem, (i, i) is the index of the minimum diagonal entry. From the matrix, this is for $i = 3$.

(c) For $i \neq j$, the (i, j)-th entry of the confusion matrix is the fraction of examples belonging to class j which get mistakenly classified as class i. Thus, for this problem, (i, j) is the index of the maximum off-diagonal entry. From the matrix, $i = 6$, $j = 0$.

From the confusion matrix, 1 is the easiest class to learn, and 3 is the hardest class. The highest amount of confusion happens between images of 0 and 6.