DESIGNING A MINIMUN DISTANCE TO CLASS MEAN CLASSIFIER

Rahat Bin Osman

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
160204083@aust.edu

ABSTRACT

The main objective or goal for this assignment is to design a minimum distance to class mean classifier

1. INTRODUCTION

Class Mean Classifier is a basic classifier that classifies a feature data vector by calculating the minimum Euclidian distance between each class mean and the data.

2. TASK

We have been given a dataset of two-class vector prototypes.

a. Plot all samples from both classes, and keep the same color and marker for each class.
b. Classify test data using the Linear Discriminant Function below.
$g_i(X) = X^T{}_i\overline{Y} - \frac{1}{2}\,{}_i\overline{Y}^T{}_i\overline{Y}$
c. Draw the decision boundary between two classes.
d. Finding Accuracy.

3. DATASET

The two dataset used to train and test the classifier are train.txt and test.txt respectively. Our train dataset has 12 labeled data in total. And test dataset has 7 labeled data.
For each row the first and second column denotes the x and y coordinates for the vectors and the last column is the class label.

METHODOLOGY

4. a. Plot the train data

The Python Matplotlib framework is being used to plot all the train data. For test *class 1* we used the marker 'o' in red color. And for class 2 we used the marker '+' in black.

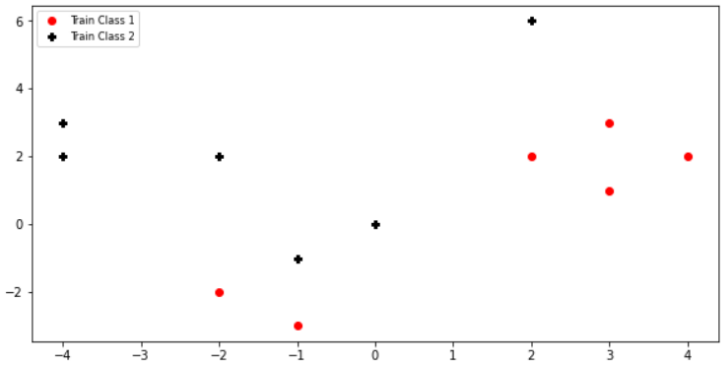

Figure 1: Plot all the train data

4. b. Classify test data

As we have two classes, we will have two mean vectors, one for each class. Running the Linear Discriminant function for every class, feature that scores better than other classifier will be put in the class that scores well.
Meaning if $g_1(x) > g_2(x)$ then x belongs to class 1. It is to note that here g(x) function is negated when derived to reduce mathematical complexity.
We used a red squared marker for class 1 and black vertically flipped triangle for class 2 as a marker while plotting.

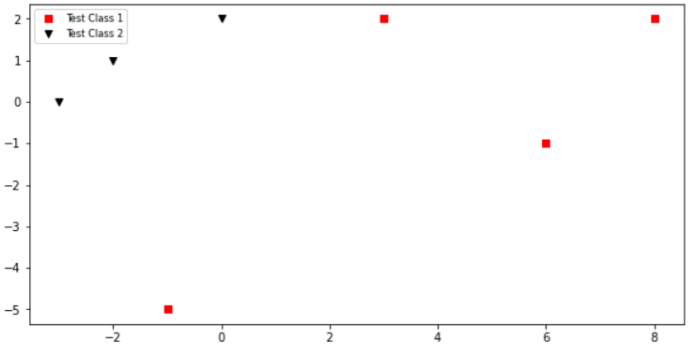

Figure 2: Classify test data

4. c. Decision Boundary

The decision boundary is the line which creates the border line between linearly separable classes.

For our mean classifier, the approximate boundary between the two classes will be:

$g_1(X) = g_2(X)$

$$=> g_i(X) - g_i(X) = 0$$

$$=> X^T{}_1\overline{Y} - \frac{1}{2}\,{}_1\overline{Y}^T{}_1\overline{Y} - X^T{}_2\overline{Y} + \frac{1}{2}\,{}_2\overline{Y}^T{}_2\overline{Y} = 0$$

$$=> X^T{}_1\overline{Y} - X^T{}_2\overline{Y} - \frac{1}{2}\,{}_1\overline{Y}^T{}_1\overline{Y} + \frac{1}{2}\,{}_2\overline{Y}^T{}_2\overline{Y} = 0$$

$$=> X^T({}_1\overline{Y} - {}_2\overline{Y}) - \frac{1}{2}\,({}_1\overline{Y}^T{}_1\overline{Y} + {}_2\overline{Y}^T{}_2\overline{Y}) = 0 \ldots(1)$$

In the above equation (1), the X^T is a feature vector of those coordinates for which the linear discriminant function for class 1 and class 2 both returns zero. Meaning, if plotted, they both are on the decision boundary line.

Hence,

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}(COEF_1 \quad COEF_2) - CONSTANT = 0$$

$$;[\,CONSTANT = \frac{1}{2}\,({}_1\overline{Y}^T{}_1\overline{Y} + {}_2\overline{Y}^T{}_2\overline{Y})]$$

$$=> X_2 = \frac{X1 \times COEF_1 + CONSTANT}{-COEF_2}$$

Plotting X_1 and X_2 for all value in the above equation for training and testing data we find the decision boundary. The range was [-4 to +8] We used a "--" in red for drawing the boundary in Matplotlib.

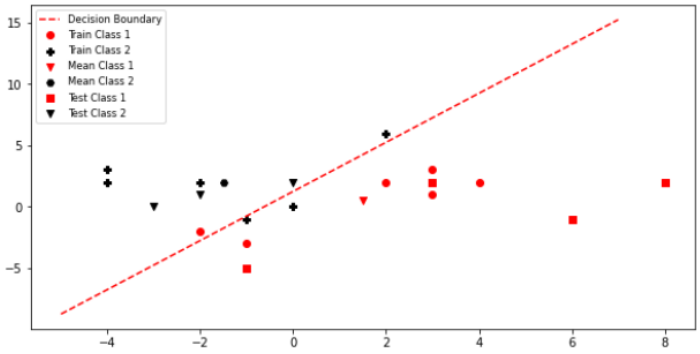

Figure 3: Decision Boundary

4. d. Accuracy

The formula for accuracy is,

$$\text{Accuracy, } \aleph = \frac{Correctly\ Classified\ Cases}{Total\ Test\ Cases} \times 100\%$$

Using the formula, we got 85.71% of accuracy in classifying the data.

CONCLUSION

The classifier had very high accuracy in decision making. Yet its misclassification rate is also unneglectable because of the boundary separates the two classes linearly.