Implementing K-Nearest Neighbors (KNN)

Rahat Bin Osman
160204083
dept. of CSE
Ahsanullah university of Science and Technology
Dhaka, Bangladesh

***Abstract*—In KNN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.**

***Index Terms*—Parameter Selection, Euclidean Distance, Neighbours, Proximity**

I. Introduction

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. similar data points are close to each other. The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematical calculation; the distance between points on a graph. The straight-line distance (also called the Euclidean distance) is a popular and familiar choice for the calculation.

II. Experimental Design / Methodology

We used a labeled text data set ”train.txt” to train the data, And ”test.txt” to test the data on and write the desired output on ”prediction.txt” file

A. Plotting the data

We take all sample points (train data) and test data from both classes, in different color and marker for each classes.

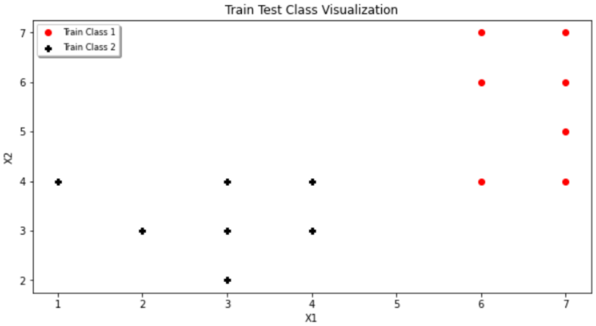

Fig. 1. Plotting the sample data

B. Run the Algorithm

Firstly, we will determine the K values (can be user input and where K is Odd) for all test files. We’ve calculated the Euclidean distance for every test points with the train data. The euclidean distance formula for two points is given below:

$$Euclidean\ distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

After calculating the distance, we sorted and took the k-th minimum distance for measuring the neighbours. And the most counted class in the determined class.

III. Result Analysis

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.

IV. Conclusion

As this algorithm is basically a majority vote among labels classification, we can use this algorithm as a recommend system in real life application. Such as, Amazon product suggestion or Netflix/YouTube movies recommendation.