UNIT-4

CLASSIFICATION

Distance based Learning Algorithms:

Distance-based algorithms are machine learning algorithms that classify queries by computing distances between these queries and a number of internally stored exemplars. Exemplars that are closest to the query have the largest influence on the classification assigned to the query. Two specific distance-based algorithms, the nearest neighbor algorithm and the nearest-hyperrectangle algorithm, are studied in detail.

It is shown that the k-nearest neighbor algorithm (kNN) outperforms the first nearest neighbor algorithm only under certain conditions. Data sets must contain moderate amounts of noise. Training examples from the different classes must belong to clusters that allow an increase in the value of k without reaching into clusters of other classes. Methods for choosing the value of k for kNN are investigated. It shown that one-fold cross-validation on a restricted number of values for k suffices for best performance. It is also shown that for best performance the votes of the k-nearest neighbors of a query should be weighted in inverse proportion to their distances from the query.

Principal component analysis is shown to reduce the number of relevant dimensions substantially in several domains. Two methods for learning feature weights for a weighted Euclidean distance metric are proposed. These methods improve the performance of kNN and NN in a variety of domains.

The nearest-hyperrectangle algorithm (NGE) is found to give predictions that are substantially inferior to those given by kNN in a variety of domains. Experiments performed to understand this inferior performance led to the discovery of several improvements to NGE. Foremost of these is BNGE, a batch algorithm that avoids construction of overlapping hyperrectangles from different classes. Although it is generally superior to NGE, BNGE is still significantly inferior to kNN in a variety of domains. Hence, a hybrid algorithm (KBNGE), that uses BNGE in parts of the input space that can be represented by a single hyperrectangle and kNN otherwise, is introduced.

The primary contributions of this dissertation are:

(a) Several improvements to existing distance-based algorithms,

(b) Several new distance-based algorithms,

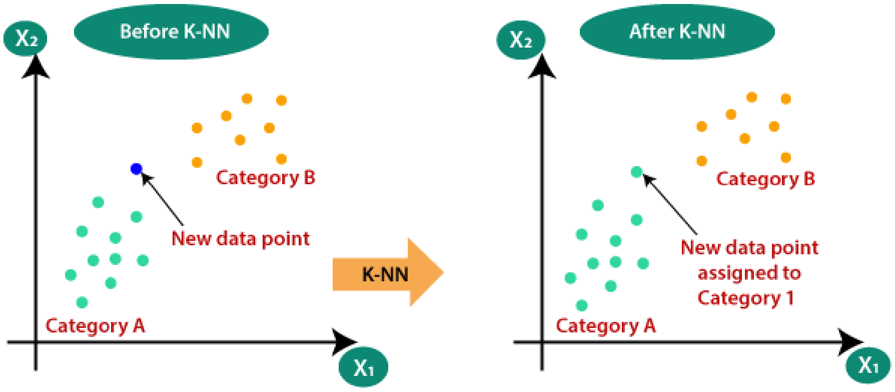
(c) An experimentally supported understanding of the conditions under which various distance-based algorithms are likely to give good performance.

K-NEAREST NEIGHBOR(KNN) ALGORITHM:

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

NEED FOR K-NN ALGORITHM:

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

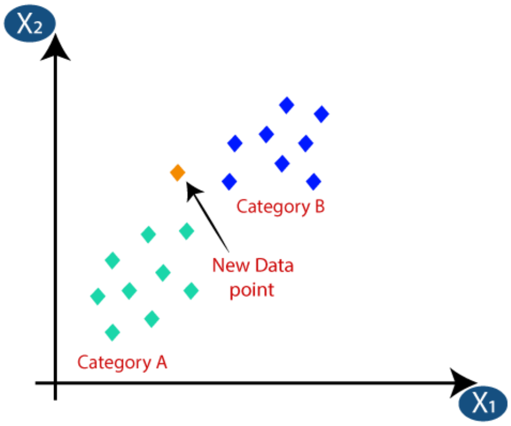


WORKING:

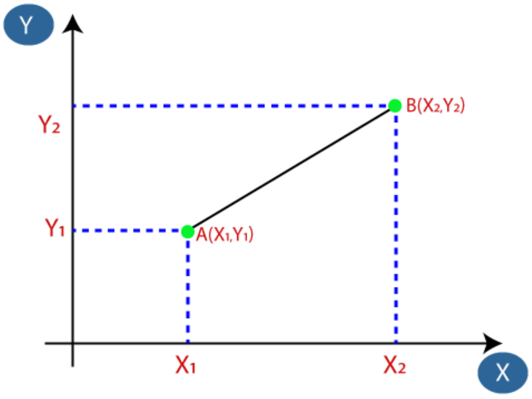
The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

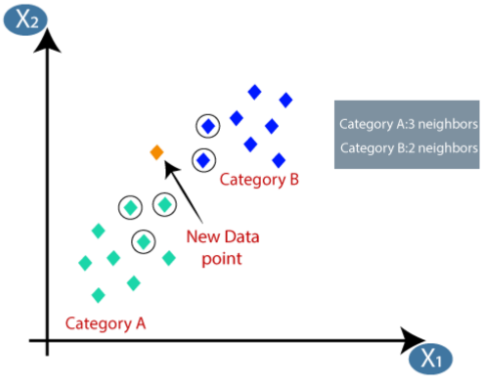


- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

SELECTION OF THE VALUE OF K IN THE K-NN ALGORITHM:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

ADVANTAGES OF KNN ALGORITHM:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

DISADVANTAGES OF KNN ALGORITHM:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.