CHAPTER 4

K-NEAREST NEIGHBOR

The k-Nearest Neighbor (k-NN) algorithm measures the distance between a query scenario and a set of scenarios in the data set. The distance between two scenarios is determined by simply pass through the data set one scenario at a time to the established measure and then compares it to the query scenario. The given dataset is represented as a matrix, containing scenarios, where each scenario contains features. A vector with length of output values accompanies this matrix, listing the output value for each scenario. It should be noted that the vector can also be seen as a column matrix; if multiple output values are desired, the width of the matrix may be expanded. k-NN algorithm has been used, improvised, developed and adapted to various situations in pattern recognition and data mining fields for a long time.

4.1 k-NN RULE

In k-NN classification, each instance is defined by some attributes and all the instances inside the data are represented by the same number of attributes, although there may be some missing attribute values. One of these attributes is called the class attribute which contains the class value of the data, whose values are predicted for new, unseen instances.
1-NN (K=1) rule assumes the value of the immediate neighbor to be the class of the new instance. However usually k-NN rule is used which assigns an instance the class which is represented mostly in its K neighbors. K can be any number of its neighbors, \( K = 1, 2, 3, 4\ldots n \), where n is the number of cases as in figure 4.1. The closeness of a neighbor is defined on the basis of attributes defining the new instance and the training instances. Training instances whose attribute values are similar to that of the new instance are considered as the nearest, but many times the exact similar instance is not found, so the nearest instance is the one with least dissimilarity.

![Figure 4.1 k-NN classifications with K neighbors](image)

In k-NN a small volume of the space of the attributes is taken and the new case is taken as the center of this volume. The radius of this volume is the distance from the new case to the \( K^{th} \) nearest neighbor. The estimated probability that this new case belongs to a certain class depends upon the
relative frequencies of the classes of the training cases in this volume. The new case is assigned to the class that has the highest estimated probability (Hand et al. 2001).

The basic form of $K=1$ gives an unstable result because of its high variance and sensitiveness which makes it inconsistent to use (Hand et al 2001), therefore larger value of $K$ is used. The value of $K$ that is to be used depends entirely upon the data set which is found out using different distribution analysis on the data. It has been observed that if the training set is large enough k-NN yields good results. In the large training set, this simple rule as a probability of error which is less than twice the Bayes probability of error and hence is less than twice the probability of error of any other decision rule, nonparametric or otherwise, based on the infinite sample set (Cover & Hart 1967).

### 4.2 DISTANCES MEASURE

The distance between two scenarios is computed using the following distance functions.

- **Absolute distance measuring:**

\[
d_A(x, y) = \sum_{i=1}^{N} |x_i - y_i|
\]  

(4.1)
Euclidean distance measuring:

\[
d_{E}(x, y) = \sum_{i=1}^{N} \sqrt{(x_i - y_i)^2}
\]  

(4.2)

Because the distance between two scenarios is dependent on the intervals, it is recommended that resulting distances be scaled such that the arithmetic mean across the dataset is 0 and the standard deviation 1. This can be accomplished by replacing the scalars with according to the following function:

\[
x^i = \frac{x - \overline{x}}{\sigma(x)}
\]  

(4.3)

Where \(x\) is the un-scaled value, \(\overline{x}\) is the arithmetic mean of feature across the data set, \(\sigma(x)\) is its standard deviation and \(x^i\) is the resulting scaled value. The arithmetic mean is defined as:

\[
\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]  

(4.4)

The computation of standard deviation is as follows:

\[
\sigma(x) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})^2}
\]  

(4.5)

4.3 PROXIMITY MEASURES

As mentioned above, K-NN calculates the distance between the attributes of new instance and previous instances to find out the class. The
term ‘distance’ depends entirely on the data; different types of data have different ways of finding out the distance. There are many different categorizations of attribute data.

- Nominal attributes are categorically discrete data that consist of category names only and there is no distance between its values, i.e., they only differ in being similar or not (=, ≠). Binary attributes are a special case of nominal attributes.

- Ordinal attributes have a natural ordering to their values, but it is not possible to define the distance between them for example tall, medium and short are three attribute values that define a person’s height but their difference cannot be given in numbers, i.e., only the symbols >, <, =, ≠ can be applied to them.

- Interval attributes have some real number values and the difference between the values is meaningful. For example when measuring the temperature in Celsius there is a meaningful difference in its values, i.e., there exist a distance between its values and >, <, =, ≠, +, - operations can be performed on them.

- Ratio attributes are like interval attributes, but the difference is that in ratio attributes the value zero (0) is absolute zero, i.e., a characteristic disappears meaning the temperature in Kelvin scale or different kinds of counts and percentages are example of the ratio scale. Ratios of these values are meaningful. Another categorization of attribute-data is between qualitative and quantitative.
Qualitative attributes are categorical attributes usually expressed as category names using natural language. They can have order or no order between their values. Nominal and ordinal attribute-data are its two types depending on the ordering in its values.

Quantitative attributes are expressed as numerical values. They describe the value as a measurable quantity; this value can be exactly measured in terms of numbers. However all numbers are not measurable like the social security number. Therefore only measurable attributes are called quantitative attributes. Interval and ratio attribute-data are its two types. There is another categorization of data.

Homogeneous data: Data in which all the attributes are of same type. For example all of the attributes are nominal or interval type.

Heterogeneous data: Data in which there are different types of attributes. For example one attribute is nominal while the other is interval.

4.4 IMPROVEMENTS TO k-NN

This section deals with the improvements or variations of k-NN in order to improve the performance and also to reduce the shortcomings of the k-NN approach.
4.4.1 Density based k-NN classifier (DB-k-NN)

Counting of the neighbors for determining the class of a test instance appears to be insufficient (Wang & Bell 2004), due to the uneven distribution of training set, k-NN classification results will have relatively large differences. For the instances in dense area, its density is clearly higher than the instance in sparse area (Shi et al. 2011). It can easily be understood that the similarity between instances in the areas that are densely populated is larger than the similarity between instances in the areas that are sparsely populated.

The instances in the dense have more chance of selection than the instances in the sparse area. In short, if the decision function used does not consider the data distribution, then this will make the instance, the density of the category of which is dense, more likely to be selected. In the decision making process, it brings a negative impact on the classification results because the prediction of smaller classes is usually not accurate.

DB-k-NN explores the potential of evaluating the neighbors in the k-NN rather than merely counting them (Voulgaris & Magoulas 2008), they provide a new look to the k-NN algorithms. Also, the distance is used, making the evaluation of the neighbors more refined. In DB-k-NN the distance between test and training instances is increased in sparse area and reduced in dense areas because it not only considers the density of test instance but also the densities of its K neighbors.
4.4.2 Variable k-NN classifier (V-k-NN)

From continuous experimentations it has been observed that the values in k-NN classification result heavily depends upon the number of neighbors \( K \) and each data has different \( K \) value that is suitable for it. It utilizes the concept of degree of certainty to do this. Degree of certainty for one classification of a classifier (Voulgaris & Magoulas 2008) is defined by the formula derived from the certainty factor CF formula in (Aydin & Guvenir 2006) which \( i \) denotes the \( i^{th} \) attribute classified, \( c \) is the class index, \( c \) is the number of classes and classification score is the score determining the classification output of the classifier. It finds the optimum \( K \) value for each classification (Voulgaris & Magoulas 2008).

In V-k-NN, an array which contains the best \( K \) value for each training set instances is made. This array is called as optimum \( K \) array and is made by performing classification for each one of the training case instances based on various neighbors. The \( K \) value that gives the maximum degree of certainty of each classification is found. Therefore, for each training set there corresponds a particular \( K \) value which is considered the best and is stored in the optimum \( K \) array. Afterwards, for each test instance, the nearest neighbor is found and its \( K \) value is assumed based on the optimum \( K \) array. Then, the k-NN classifier is applied to that test instance, using that \( K \) value. According
to (Voulgaris & Magoulas 2008) this is something similar to one of the ideas presented in (Khan et al. 2002).

### 4.4.3 Class based k-NN classifier (CB-k-NN)

Many of datasets have the same common problem when it comes to classification that is the difference of their class sizes, meaning that one class will have too many instances while others have too few instances. When finding classification based upon the k-NN classification the classes with very few instances are not selected. The Class Based k-NN (CB=k-NN) was developed due to this fact (Voulgaris & Magoulas 2008). In CB-k-NN, for every test instance, the K nearest instances of each class is taken. The value of K is automatically selected to maximize the degree of certainty of the classification; this selection is done by the classifier to decrease the influence of the most distance instance. Afterwards, the harmonic mean of the distances of these neighbors (the inverse of the mean of the inverses of distances) is calculated (so that it is not influenced so much by the most distant instances).

### 4.4.4 Discernibility k-NN classifier (D-k-NN)

D-k-NN is similar in structure to the original k-NN extension of the DB-k-NN. The aim was to make an algorithm that is quite fast, without losing accuracy (Voulgaris & Magoulas 2008). It takes into account the distance and discernibility of each neighbor. It takes the discernibility of each instance
which is measured by taking a radius around each instance that is the average
distance between this instance and the other instances of the same class. After
that the instances with the same class and the total instances within that radius
are divided and the result is called discernibility of that case. By dividing the
discernibility by the distance, a score is produced, for each one of the
neighbors. Then, the scores for each class are averaged to produce one
classification score for each one of the classes. The class yielding the highest
classification score is selected for the classification.

4.4.5 **Weighted k-NN Classifier (W-k-NN)**

A W-k-NN performs an evaluation of the attributes of the instances. The concept of W-k-NN is similar to the DB-k-NN classifier, which performs evaluation on the cases. Each attribute is evaluated to obtain a weight value based on how useful this attribute is for discerning the classes of the dataset. Just like in the V-k-NN classifier these weights are stored in a vector format with one value for each attribute.

The W-k-NN classifier works in the manner that each one of the attributes of the training set is evaluated using the index of discernibility. The index of discernibility is a measure developed for assessing how easily distinguishable the classes of a dataset are and evaluating individual attributes by applying it on them. The weights are applied to both the training and the testing set when the distances between their cases are being calculated.
4.5 k-NN PATTERN CLASSIFICATION

For classification of data it is extremely important that the nature of data is known which is either parametric or non-parametric. Parametric data means that there is some statistical distribution between its instances which is known beforehand and vice versa, this information is essential because different algorithms and techniques are more suitable for different types of data. For parametric data there exist many different techniques, but Bayesian analysis will give the optimal result (Cover & Hart 1967). k-NN is a classification technique that assumes the class of an instance to be the same as the class of the nearest instance.

It adopts a similarity metric to measure the proximity of an instance to other instances. It is one of the most simple non-parametric decision rules. Here the term non-parametric refers to the fact that there is no prior knowledge of the statistical distribution of the data. Nearest neighbor assumes that instances in the data are independently and identically distributed, so the instances which are in close proximity have the same classification (Cover & Hart 1967).

k-NN classification is one type of instance-based learning methods which are sometimes called as lazy learning methods. k-NN is purely lazy, it simply stores the entire training set and postpones all efforts towards inductive
generalization until classification time (Wettschereck et al. 1997). Instance-based learning methods are defined by their three properties:

- They store all of the training data during the learning process.

- Generalization beyond the training data is delayed until a value is predicted for a new case because any new query is answered by comparing the new case to the training data.

- From the training data they search for a case that is similar to the new case.

In this chapter, the theory of existing k-NN classifier is discussed along with its different variants. The performance of the proposed e-voting system is also evaluated using k-NN classifier and the performance measures obtained while using k-NN are given in Chapter 7 with SVM, Naïve Bayes and random tree. Based on the theoretical background, the pseudo code for the proposed e-voting system using k-NN classifier is as follows:

**Pseudo code of proposed system using k-NN classifier**

**Input:** MP Election ARFF datasets, \( Data = \{(y_1, x_1), \ldots, (y_N, x_N)\} \) and \( y = (y_1, \ldots, y_N) \) new instance to be classified

**Output:** Predict (P) the Precision, Recall(R), F-Measure (F), Mean Absolute Error MAE, Root Mean Square Error (RMSE)
Procedure:

Step 1: For each labeled instance \((y_j, c_j)\) calculate \(d(y_j, y)\).

Step 2: Order \(d(y_j, y)\) from lowest to highest, \((i = 1, ..., N)\).

Step 3: Select the \(K\) nearest instances to \(x: \text{Data}^K_x\).

Step 4: Assign to \(x\) the class in \(\text{Data}^K_x\).

Step 5: Predict the accuracy;

Step 6: Predict classification error;

Step 7: Display the P, R, F and MAE & RMSE;