

جامعة الباحة

**Research Title**

An Adaptive k-Nearest Neighbor Algorithm

**Student ID:**

**1111111**

**Student Name:**

**sssssss**

**Dr:**

**sssssss**

**Index**

1. History …………………………………………………………………….
2. Method Types…………….………………….…………………………….
3. Definition …………………………………………………………………
4. When do we use K-NN algorithm…………………………………………
5. Working of KNN Algorithm………………………………………………
6. Example………………….…………………………………………………
7. Advantages & Disadvantage……………………………………………….
8. Applications of KNN………………………………………………………
9. An Adaptive *k*-Nearest Neighbor Algorithm………………………………
10. Experiments……………………………………………………………….
11. Conclusions……………………………………………………………….
12. Reference…………………………………………………………….……

k-Nearest Neighbor Algorithm

***History:***

the k-nearest neighbors’ algorithm (k-NN) is a non-parametric classification method first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover. It is used for classification and regression. In both cases, the input consists of the k closest training examples in data set. The output depends on whether k-NN is used for classification or regression:

In k-NN 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.

In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically.

Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

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.

***Parametric vs Non-parametric Methods***

Let us look into how different is a parametric machine learning algorithm from a nonparametric machine learning algorithm.

Machine learning, in other words can be called as learning a function (f) which maps input variables (X) to the output variables (Y).

Y=f(X)

An algorithm learns about the target mapping function from the training data. As we are unaware of the form of the function, we have to evaluate various machine learning algorithms and figure out which algorithms perform better at providing an approximation of the underlying function.

Statistical Methods are classified on the basis of what we know about the population we are studying.

Parametric statistics is a branch of statistics which assumes that sample data comes from a population that follows a probability distribution based on a fixed set of parameters.

Nonparametric statistics is the branch of statistics that is not based solely on population parameters.

***Definition:***

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well

* **Lazy learning algorithm** − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
* **Non-parametric learning algorithm** − KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.

***When do we use K-NN algorithm?***

K-NN algorithm can be used for applications which require high accuracy as it makes highly accurate predictions. The quality of predictions is completely dependent on the distance measure. Thus, this algorithm is suitable for applications for which you have sufficient domain knowledge so that it can help you select an appropriate measure.

As we have already seen K-NN algorithm is a type of lazy learning, the computation for the generation is postponed until classification which indeed increases the costs of computation compared to other machine learning algorithms. But still K-NN is considered to be the better choice for applications where accuracy is more important and predictions are not requested frequently.

K-NN can be used for both regression and classification predictive problems. However, in the industry it is mostly used in classification problems.

Generally we mainly look at 3 important aspects in order to evaluate any technique:

* Ease to interpret output
* Calculation time
* Predictive Power

***Working of KNN Algorithm:***

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

**Step 1** − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

**Step 2** − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

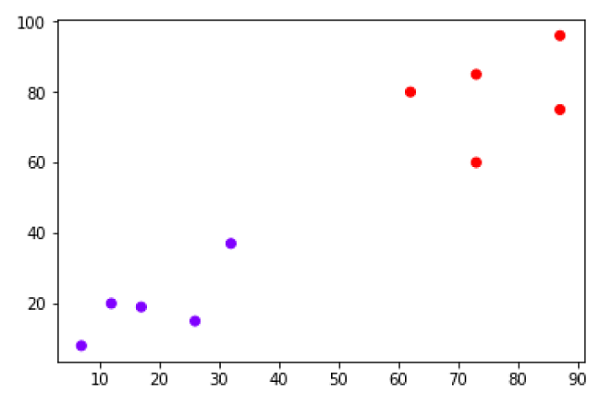
**Step 3** − For each point in the test data do the following −

* 3.1 − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
* 3.2 − Now, based on the distance value, sort them in ascending order.
* 3.3 − Next, it will choose the top K rows from the sorted array.
* 3.4 − Now, it will assign a class to the test point based on most frequent class of these rows.

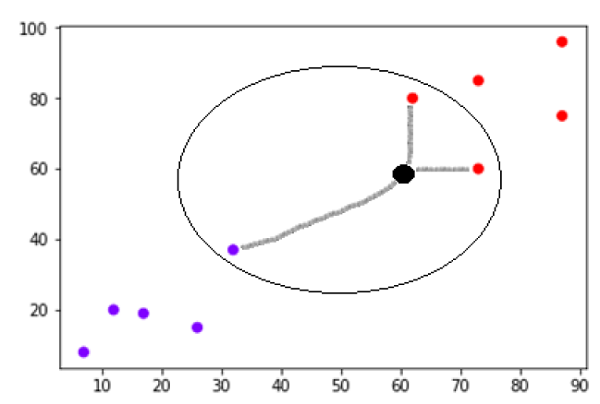
**Step 4** – End

***Example:***

The following is an example for concept of K and working of KNN algorithm ,Suppose we have a dataset which can be plotted as follows



Now, we need to classify new data point with black dot (at point 60,60) into blue or red class. We are assuming K = 3 i.e. it would find three nearest data points. It is shown in the next diagram



We can see in the above diagram the three nearest neighbors of the data point with black dot. Among those three, two of them lies in Red class hence the black dot will also be assigned in red class.

***Advantages & Disadvantage:***

**Below is the list of few of the reasons to choose K-NN machine learning algorithm:**

1. **K-NN is pretty intuitive and simple**: K-NN algorithm is very simple to understand and equally easy to implement. To classify the new data point K-NN algorithm reads through whole dataset to find out K nearest neighbors.
2. **K-NN has no assumptions**: K-NN is a non-parametric algorithm which means there are assumptions to be met to implement K-NN. Parametric models like linear regression has lots of assumptions to be met by data before it can be implemented which is not the case with K-NN.
3. **No Training Step:**K-NN does not explicitly build any model, it simply tags the new data entry based learning from historical data. New data entry would be tagged with majority class in the nearest neighbor.
4. **It constantly evolves**: Given it’s an instance-based learning; k-NN is a memory-based approach. The classifier immediately adapts as we collect new training data. It allows the algorithm to respond quickly to changes in the input during real-time use.
5. **Very easy to implement for multi-class problem:** Most of the classifier algorithms are easy to implement for binary problems and needs effort to implement for multi class whereas K-NN adjust to multi class without any extra efforts.
6. **Can be used both for Classification and Regression:** One of the biggest advantages of K-NN is that K-NN can be used both for classification and regression problems.
7. **One Hyper Parameter:** K-NN might take some time while selecting the first hyper parameter but after that rest of the parameters are aligned to it.
8. **Variety of distance criteria to be choose from**: K-NN algorithm gives user the flexibility to choose distance while building K-NN model.
   1. Euclidean Distance
   2. Hamming Distance
   3. Manhattan Distance
   4. Minkowski Distance

**Even though K-NN has several advantages but there are certain very important disadvantages or constraints of K-NN. Below are listed few cons of K-NN.**

1. **K-NN slow algorithm**: K-NN might be very easy to implement but as dataset grows efficiency or speed of algorithm declines very fast.
2. **Curse of Dimensionality:**KNN works well with small number of input variables but as the numbers of variables grow K-NN algorithm struggles to predict the output of new data point.
3. **K-NN needs homogeneous features**: If you decide to build k-NN using a common distance, like Euclidean or Manhattan distances, it is completely necessary that features have the same scale, since absolute differences in features weight the same.
4. **Optimal number of neighbors**: One of the biggest issues with K-NN is to choose the optimal number of neighbors to be consider while classifying the new data entry.
5. **Imbalanced data causes problems**: k-NN doesn’t perform well on imbalanced data. If we consider two classes, A and B, and the majority of the training data is labeled as A, then the model will ultimately give a lot of preference to A. This might result in getting the less common class B wrongly classified.
6. **Outlier sensitivity:**K-NN algorithm is very sensitive to outliers as it simply chose the neighbors based on distance criteria.
7. **Missing Value treatment:** K-NN inherently has no capability of dealing with missing value problem.

***Applications of KNN:***

The following are some of the areas in which KNN can be applied successfully –

**Banking System**

KNN can be used in banking system to predict weather an individual is fit for loan approval Does that individual have the characteristics similar to the defaulters one?

**Calculating Credit Ratings**

KNN algorithms can be used to find an individual’s credit rating by comparing with the persons having similar traits.

**Politics**

With the help of KNN algorithms, we can classify a potential voter into various classes like “Will Vote”, “Will not Vote”, “Will Vote to Party ‘Congress’, “Will Vote to Party ‘BJP’.

Other areas in which KNN algorithm can be used are Speech Recognition, Handwriting Detection, Image Recognition and Video Recognition

An Adaptive k-Nearest Neighbor Algorithm

***Introduction:***

The traditional kNN usually assumes that the training samples are evenly distributed among different classes.

In order to improve the performance of the traditional kNN in practical applications, there are propose an improved algorithm: ***Adaptive*** *k****-Nearest Neighbor Algorithm***

Adaptive k-Nearest Neighbor: Classification based on k-nearest neighbors (kNN classification) is one of the most widely used classification methods. The number k of nearest neighbors used for achieving a high accuracy in classification is given in advance and is highly dependent on the data set used. If the size of data set is large, the sequential or binary search of NNs is inapplicable due to the increased computational costs. Therefore, indexing schemes are frequently used to speed-up the classification process. If the required number of nearest neighbors is high, the use of an index may not be adequate to achieve high performance. In this paper, we demonstrate that the execution of the nearest neighbor search algorithm can be interrupted if some criteria are satisfied.

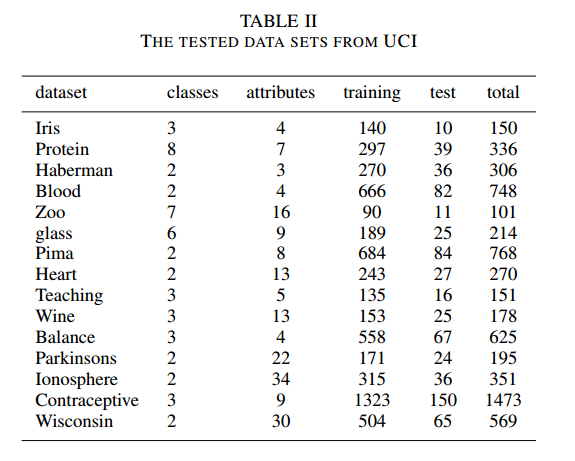
***Adaptive*** *k****-Nearest Neighbor Algorithm:***

The proposed algorithm is an improved kNN algorithm deriving from the traditional kNN. Base on the principle that nearest neighbors have similar attributes, we can assume that the test example has the most similar attributes with its nearest neighbor in the training set. The probability is high that a test example adopts the same kNN algorithm as its nearest neighbor in the training set to get its correct class label. The optimal k is the number of the fewest nearest neighbors a training example has to identify to get its correct class label when assuming it is a test example to the other training examples. Therefore, if we want to get a test example’s label, we just need to get the optimal k of its nearest neighbor in the training set. According to the above analysis, propose the idea of adaptive k-nearest neighbor algorithm (AdaNN).

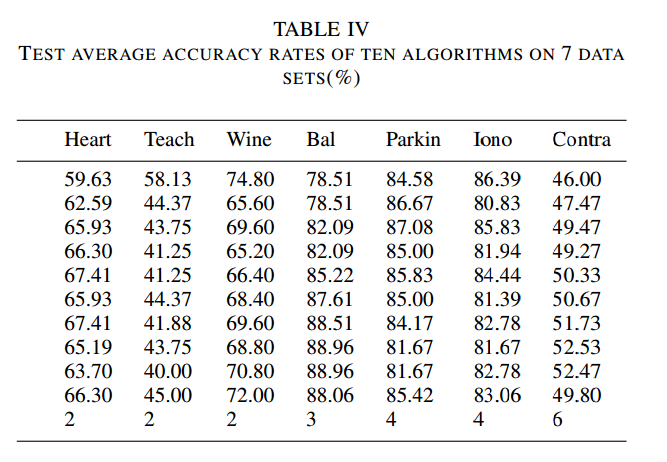
***EXPERIMENTS:***

**A. Datasets used from UCI**

There are 9 test traditional kNN algorithms, from 1NN to 9NN, and the AdaNN algorithm on a number of real world pattern recognition problems. For each data set, 90% of all examples were randomly selected as training examples and the rest 10% as testing ones. The detailed information of the 15 data sets is shown in TABLE II, where the data set name listed in the table is the first word of its full name.



**C. Experimental Results**



The experimental results for 13 data sets are summarized in TABLE IV.

There are tested the proposed algorithm on 15 data sets.

Comparing to the other 9 traditional kNN algorithms from 1NN to 9NN, the AdaNN performs the best on the data sets of Iris, protein, Haberman and Blood. It gets the second best performance on six data sets, the third best performance on one data set and the fourth best performance on two data sets.

Experimental results also reveal that the AdaNN gets the worst performance on the last two data sets. It ranks the sixth in the ten.

***Conclusions:***

An adaptive k-nearest neighbor algorithm (AdaNN) is brought forward to overcome the limitation of the traditional k-nearest neighbor algorithm (kNN) which usually identifies the same number of nearest neighbors for each test example. It is known that the value of k has crucial influence on the performance of the kNN algorithm, and improve kNN algorithm focuses on finding out the suitable k for each test example.

***References:***

- (Research) An Adaptive *k*-Nearest Neighbor Algorithm.

- (Research) Rigidity of Microsphere Heaps.

- http://archive.ics.uci.edu/ml/

- <http://www.scholarpedia.org/article/K-nearest_neighbor>

- <https://ieeexplore.ieee.org/document/5569740>

- <https://www.ncsu.edu/search/global.php>

- <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>