Indian Journal of Science and Technology, *Vol 9(3), DOI: 10.17485/ijst/2016/v9i3/72262, January 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Design and Development of Data Classification Methodology for Uncertain Data

Rashmi Agrawal*

Manav Rachna International University, Faridabad
rashmi.sandeep.goel@gmail.com

Abstract

The classification of uncertain data has become one of the tedious processes in the data-mining. The uncertain dataset contains tuples with different and multiple data and thus to predict correct output class is a complex process. In this paper, we present two algorithms for classifying the uncertain data using the KNN classifier, which handles the uncertain dataset in two different ways to discover the corresponding class. In both algorithms, we split the database into two portions. The first portion is named as training dataset and the second portion is name as testing dataset. In the first algorithm, we used the properties of uncertain data to convert the uncertain data into certain data. The algorithm 1 initially converts the uncertain data to certain data and then it utilizes the KNN algorithm to classify data through the distance measure. The second algorithm converts the uncertain data through probability distribution function (pdf). The algorithm 2, initially calculates the N number of split point for each attributes of the training part of uncertain data then it calculates pdf with respect to the selected split point. The same process is applied for testing portion of uncertain data; subsequently algorithm 2 employs the KNN algorithm to classify the converted data. Finally, we compared our proposed algorithm with the UDT (Uncertain Decision Tree) algorithm with the four real datasets such as iris dataset, ionosphere dataset, breast cancer dataset, glass dataset and we proved that our proposed algorithms performed well than the UDT algorithm in terms of accuracy.

Keywords: Classification, K Nearest Neighbor Algorithm (K-NN), Probability Distribution Function, Uncertain Data, Uncertain Data Classification

1. Introduction

The commercial and research interests in data mining is increasing expeditiously, as the amount of data generated and stored in databases of organizations is already excessive and still continuing to grow very fast[1]. As more data are gathered, with the amount of data doubling every three years, data mining has become progressively important tool to transform these data into applicable information in the desired domain. In general, data mining refers to the non-trivial extraction of implicit, previously unknown and potentially useful information from data in databases. Classification is a well-perceived data mining task and it has been studied and applied extensively in the fields of statistics, text and pattern recognition, decision theory, machine learning, neural networks and more. Classification is a supervised learning method that induces a classification model from a database. The objective of classification technique is to assign a new object to a class from a given set of classes based on the attribute values of the object[2]. Classification approaches uses a training set where all objects are already associated with known class labels. The classification algorithm learns from the training dataset and builds a model. The model is used to classify new objects[3,9]. K-NN is one of the well-known algorithms of top-10 data mining algorithm.

The k-Nearest Neighbor (K-NN) is one of the simplest classification methods used in data mining and machine

*Author for correspondence

learning. Despite the simplicity of the algorithm, it performs very well and is an important yardstick method. It constructs the classification model by getting votes of the k-Nearest Neighbors. Several reasons account for the widespread use of k-NN algorithm: It is straightforward to implement, it generally produces good recognition performance, and its complexity is independent of the number of classes. The k-NN algorithm classifies a new instance by noting its distance from each member of a database of classified examples and assigning the new instance to the class of the majority of its nearest neighbors. The k- NN is well suited for multi-modal classes as well as the applications in which an object can have many class labels[4,10]. A wide range of supervised learning algorithms has been applied to this area, such as K-Nearest Neighbor (KNN)[5] (Sebastiani[5]), Centroid-Based Classifier (CB)[6] (Han and Karypis[6]), Naive Bayes[5] decision trees and Support Vector Machines (SVM). Among all these algorithms, K-Nearest Neighbor is a widely used text classifier because of its simplicity and efficiency. In its training-phase all the examples are simply stored; Therefore recurrently it is called as lazy learner[5,11].

In despite of its merits, K-Nearest Neighbor has a major problem of inductive biases or model misfits. For example, it assumes that the training data are evenly distributed among all classes. In practice, however, there is no guarantee that the training set is balanced populated, such as Reuter-21578 and TDT-5, and especially for Reuter-21578 in which the documents are extremely unevenly distributed. In such cases, some classes have majority of examples as compared to other classes. If we implement traditional KNN algorithm to classify the test document d, the test documented tends to be assigned the majority class label. As a result, the major category tends to have high classification accuracy as compared to the minority classes which have low classification accuracy. Therefore, the total performance of KNN will be inevitably harmed[7].

In this paper, we present two algorithms using the KNN classifier, which handles the uncertain dataset to predict the corresponding class. In both algorithms, we split the database into two partitions. The first partition is named as training dataset and the second partition as testing dataset. In the first algorithm, we used the properties of uncertain data to convert the uncertain data into certain data. The algorithm 1 initially converts the uncertain data to certain data and then it utilizes the KNN algorithm to classify data through the suitable distance measure. The second algorithm converts the uncertain data through probability distribution function (pdf). The algorithm 2, initially calculates the N number of split point for each attributes of the training part of uncertain data then it calculates pdf with respect to the selected split point. The same process is applied for testing portion of uncertain data; subsequently algorithm 2 employs the KNN algorithm to classify the converted data.

The rest of the paper is organized as follows: A brief review of some of the literature works is presented in section 2. The problem identification and contribution of the paper given in section 3, the proposed design and development of data classification methodology for the uncertain data is given in section 4. The experimental results and performance analysis discussion are provided in section 5. Finally, the conclusions are summed up in section 6.

2. Related Works

Literature presents several algorithms for data classification in data mining. Among the various techniques of data classification, decision tree and K-NN classifier played major role in data mining community. Here, we review recent literature of modified algorithms from decision tree and K-NN classifier.

K-NN classifier may have problem incase training samples are uneven. The difficulty with KNN classifier is that it decreases the precision of classification in case of uneven density of training data. Lijuan Zhou[18] proposed clustering-based K-NN method. It predefines training data with the help of clustering method, and then they classify with a new KNN algorithm which implement a dynamic adjustment in each iteration for neighborhood number K. This proposed method helps to avoid uneven classification phenomenon and helps to decrease the miscalculation of boundary testing samples.

In case if there are an infinite number of samples in training set, then the possible outcome from the Nearest Neighbor classification (kNN) is independent on its assumed distance metric. But it is unfeasible that that the number of training samples is infinite. Hence, selecting distance metric becomes major issue in deciding the performance of KNN. Yunlong Gaoa[19] proposed Two-Level Nearest Neighbor Algorithm (TLNN) in order to decrease the mean-absolute error of the misclassification rate of kNN with finite and infinite number of training

samples. At low-level, Euclidean distance is used in order to determine a local subspace centered at an unlabeled test sample. At high level, they used AdaBoost as guidance to extract local information. Here data variance was maintained with TLNN method and also highly stretched or elongated neighborhoods were generated along different direction. The TLNN decreases the extreme dependence on the statistical method, which realized former knowledge from training data. Also the linear combination of few base classifier generated by weak learner in AdaBoost can produce much better kNN classifiers.

In the kNN algorithm, the predefined k values neglect the influence of category and document number of training text. Hence, choosing the accurate value of K can attain better classification results. An Gong and Yanan Liu[15] proposed a type of dynamic attain k-valued for kNN classification algorithm, their experimental results proves that the dynamic attain k-valued kNN classification algorithm with high performance. While dealing with excessive data, a significant disadvantage of existing kNN algorithm is that the class with more frequent samples tends to govern the neighborhood of test request irrespective of distance measurements, which results in suboptimal classification performance on minority class. In order to solve this problem, Wei Liu, Sanjay Chawla[15] proposed CCW (Class Confidence Weight) which make use probability of attribute value updated in the class labels to weight prototypes in kNN. The main benefit of using CCW is that it is able to precise the inherent bias to majority class in existing kNN algorithm on any distance measurement. This is proved by theoretical analysis and comprehensive experiments.

Shu Zhao, et al in paper[11] has proposed an algorithm called Multi-Instance Covering kNN (MICkNN) for mining from multi-instance data. In the first step, the constructive covering algorithm is applied to make restructure of the structure from original multi-instance data. Then, the basic kNN algorithm is employed to segregate the false positive instances. Later on in the test step, tested bag is labeled directly conferring to the similarity between the unseen bag and sphere neighbors obtained from last two steps. It was found through the experimental results that the proposed algorithm was competitive enough from most of the state-of-the-art multi-instance methods in both classification accuracy and running time. Indu Saini, et al in paper[13] have proposed digital band-pass filter is used to reduce false detection caused by interference present in ECG signal and further gradient of the signal is used as a feature for QRS-detection. They discussed that the accuracy of KNN based classifier is broadly dependent on the value of K and type of distance metric. The author's from[20] and[21] have discussed about classification of uncertain data.

The above researches are[12,13,16-19] primarily focused on modifying the KNN algorithm to improve the performance of the KNN algorithm based on certain database but in our proposed algorithm we are dealing with the uncertain database. Classification of uncertain data is not an easy task and we adapt KNN algorithm to classify the uncertain data.

3. Problem Identification and Contribution of the Paper

The main problem of this paper is classifying the uncertain data using KNN. The KNN algorithm classifies the data using the suitable distance measure for certain databases. However, calculation of distance measure process in KNN is not suitable for classification of the uncertain data. To solve the above problem in this paper, we presented two methods to adopt the KNN algorithm for classifying the uncertain data.

- Processing the uncertain data through properties of tuple data
- Processing the uncertain data through probability distribution function

4. Design and Development of Data Classification Methodology for the Uncertain Data

In this section, we describe our proposed algorithms, algorithm 1 and algorithm 2, which helps in classifying the uncertain data. In both algorithms, we employ the KNN algorithm (K Nearest Neighbor algorithm) for classification purpose.

Consider the uncertain database, which consists of M number of attributes including the class attribute with N number of tuples. Each tuple has T number of data in each attribute, which is represented as t_{mn} where the value of m represents the attribute which varies from 1

to M-1, and the Mth attribute is decision attribute. The symbol n represents the tuple id. Here, we separate the uncertain database into two parts, the first one is training part, the second one is testing part, and the size of the respective part is 80% and 20% respectively. The following process is applicable for both training and testing part of the uncertain data.

The following Figure 1 represents the overall block diagram of the proposed methodology.

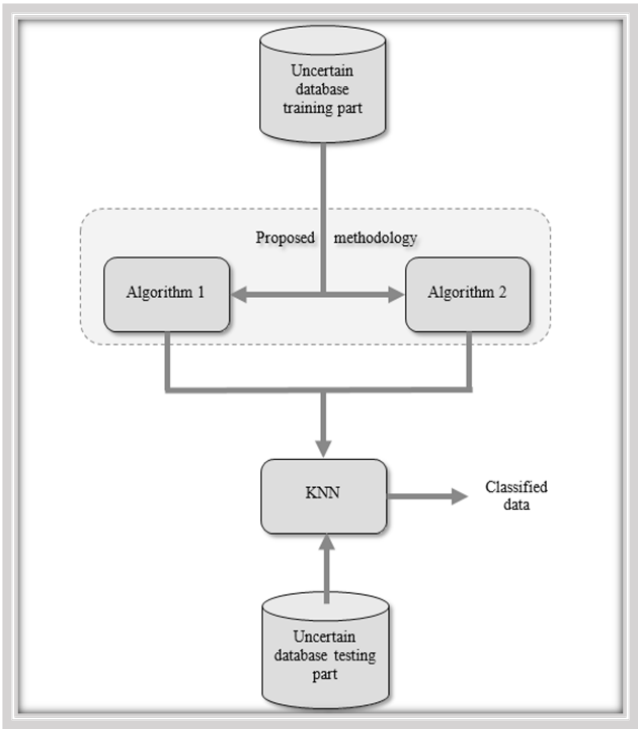

Figure 1. Represents the overall block diagram of the proposed methodology.

4.1 Algorithm 1

Algorithm 1 classifies the uncertain data by converting the uncertain data into certain data through the properties of the tuple data such as mean, mode, median and variance which is described as below. Each t_m has T number of data values in it. In order to convert this T number of data into single data we are doing the following procedure for each t_m.

- Finding the most frequent data in each t_m
- Finding the median value of t_m
- Finding the mean value of t_m
- Finding the variance value of t_m
- Calculation of data range of t_m

4.1.1 Selection of Most Frequent Data

In order to calculate the most frequent data, we calculate the count value of each data $cnt(d_i)$ in in t_m and subsequently we select most frequent data $f(d_i)$, which has max value of $cnt(d_i)$.

4.1.2 Selection of Median Value

With the intention of selecting the median of the tuple data, initially, we arrange the tuple data in ascending order. If the total number of tuple data T in t_m is odd then the position of the median is obtained through following Equation (1) and else the value of T_m is even then value of the median is calculated directly through the following Equation (2). Once the position of the median value is obtained through the above equation subsequently, the median value of the tuple data is selected from the arranged data through the selected position.

$$P\left(\mathrm{M}\left(t_{mn}\right)\right)=\frac{T+1}{2} \tag{1}$$

$$\left(\mathrm{M}\left(t_{mn}\right)\right)=\frac{\left(\left(\frac{T}{2}\right)^{th} data+\left(\frac{T}{2}+1\right)^{th} data\right)}{2} \tag{2}$$

From the above Equation (1), $P(\mathrm{M}(t_m))$ represents the position of the median (M) of t_m and the T represents the total number of data present in t_m. The symbol $(\mathrm{M}(t_m))$ from above Equation (2) represents that median M value of t_m.

4.1.3 Selection of Mean Value

The calculation of mean value is an average of the tuple data; The following Equation (3) is used to calculate the mean value of the tuple data t_m.

$$\mu\left(t_{mn}\right)=\frac{\sum_{t=1}^{T} d_t}{T} \tag{3}$$

4.1.4 Calculation of Variance Value

For calculating the variance, the following Equation (4) is used.

$$v\left(t_{mn}\right)=\sqrt{\frac{\sum_{t=1}^{T}\left(d_t-\mu\right)^2}{T}} \tag{4}$$

After calculation of above parameters, the next step is calculation of data range for each tuple data t_m. Based on the result of the data range value and the data present in t_m, the uncertain data are converted into certain data. The following Equations (5) and (6) are used to calculate the minimum and maximum data range for each t_m.

$$R_{\min}(t_m) = \mu - \nu \quad (5)$$

$$R_{\max}(t_m) = \mu + \nu \quad (6)$$

Once the data range value for t_m calculated, we used following three conditions to match the data range value $R(t_m)$ with tuple data present in t_m for selecting the single data for each t_m then the dataset becomes certain dataset. The conditions are showed in the following

Case (i): If t_m has single value within the data range of $R_{\min}(t_m)$ and $R_{\max}(t_m)$, then we select that single value and place that single value in t_m by removing other values present in t_m.

Case (ii): If t_m has more than one value within the data range of $R_{\min}(t_m)$ and $R_{\max}(t_m)$, then we place median value of t_m in by eliminating the existing data in t_m.

Case (ii): If t_m has no value within the data range of $R_{\min}(t_m)$ and $R_{\max}(t_m)$, then we place most frequent value of t_m in by eliminating the existing data in t_m.

Based on the above process, we convert the uncertain data into certain data for each t_m in the training data.

4.1.5 K Nearest Neighbor Algorithm (KNN)

Once we convert the uncertain testing data into certain data through above process, which is stated in section 4.1, then we apply suitable distance measure on this data for classification. From the result of the distance measure, we can find the corresponding class of the testing data.

4.1.6 Calculation of Distance Measure

In this paper, we calculate the distance using the standard Euclidian distance measure, which is defined in the following Equation (7).

$$D(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_{M-1} - y_{M-1})^2} \quad (7)$$

Where x1 represents first attribute of testing tuple and y1 represents the first attribute of training tuple.

Once we calculate the distance of testing data with all other training data, we selects K number of training data with its class which are having minimum distance with testing data. From the K number of selected training data, we calculate the count value of each class. Once we calculated the frequency of the class from the selected number of training data subsequently algorithm selects class for testing data, which has more frequency among the available.

Algorithm Procedure for Algorithm 1

Input: Uncertain database
Output: Classified output

Parameters

UDB- Uncertain database
UDB_R- Training part of uncertain database
UDB_{TS}- Testing part of uncertain database
t_m- Tuple data, where m represents the attribute and n represents the tuple
$cnt(d_i)$- Count value of data i in tuple data t_m
$f(d_i)$- Frequent data from tuple data t_m
T_m- Total number of data in t_m
M- Median
$P(M(t_m))$- Position of the median M
μ- Mean
ν- Variance
d_i - Tuple data from t_m
$f(d_i)$- Most frequent data in t_m
$K(t_n)$ - selecting K number of tuples which has minimum distance
$Cnt(c_i)$- count value of class

Begin
{
Read UDB_R and UDB_{TS}
For each t_m
Calculate $cnt(d_i)$
Select max of $cnt(d_i)$ à $f(d_i)$
Get T_m
Sort T_m ascending order
If T_m àodd
Calculate $P(M(t_m))$ equation (1)
M à $P(M(t_m))$ from sorted T_m
Else if T_m àeven
Calculate $M(t_m)$ equation (2)
End if

Calculate μ equation (3)
Calculate v equation (4)
Calculate R_{min} and R_{max} equations (5 and 6)
If R_{min} to R_{min} = single d_i from t_m
Remove all T from t_m
Place t_m à d_i
Else if R_{min} to R_{min} = more than 1 d_i from t_m Remove all T from t_m
Place t_m à M
Else if R_{min} to R_{min} = {} empty from t_m
Remove all T from t_m
Place t_m à $f(d_i)$
End if
End for
For each t_n from UDB_{TR}
Calculate distance with all t_n from UDB_R (equation 7)
Select min distance à K(t_n)
Calculate $Cnt(c_i)$
Select max $Cnt(c_i)$
Assign class for t_n
End for
End

4.2 Algorithm 2

In this section, we describe algorithm 2 that classify the uncertain data without converting it to certain data. Our proposed algorithm 2 utilizes the probability distribution function for classifying the uncertain data. For calculating the pdf function initially, the algorithm 2 selects N number of split points from each attribute of uncertain dataset. The selection of split point of this algorithm is based on the fuzzy entropy[24]. Once the split points for each attribute has been selected through the fuzzy entropy value subsequently, the algorithm calculates probability distribution function with respect to the each selected split point for each tuple data t_m from the training data. The above process is applied for each tuple data of training data and the result of the above process converts the uncertain training dataset into certain dataset where each data tuple consists of calculated pdf instead of uncertain data. The selection of split point and the calculation of probability distribution function are given below.

4.2.1 Selection of Split Point

The uncertain dataset has multiple data in each tuple with respect to its attribute. To solve this problem in this paper, with the help of fuzzy entropy calculation, we select N number of data as split points from each attribute to calculate the pdf function for each data tuple. With the intention of selecting the split points, we calculate the fuzzy entropy value for every data in each attribute from which we selects N number of data as split point having the minimum entropy value. The calculation of entropy value is showed in the following Equation (8).

$$H(z, A_m) = \sum_{X=p_1, p_2} \frac{|X|}{|S|} \left(\sum_{c \in C} -p_c/X \log_2 p_c/X \right) \quad (8)$$

Where p_1 contains the set of data, which are equal and lesser than the split point $p_2 = \{d\} \leq z$, the p_2 has the set of data, which are greater than split point $p_2 = \{d\} > z$, and the S represents total number of data present in the attributec A_m and the $|X|$ represents number of data belongs to X.

4.2.2 Calculation of Probability Distribution Function

With the help of the split points, we calculate the probability distribution function for each tuple belonging to each attribute. The calculation of probability distribution function is given in the following Equation (9).

$$P(t_{mn}) = \sum_{z=1}^{N} \frac{|z|}{|T|} \quad (9)$$

The above process is repeated for each testing data. At this stage, we apply the KNN algorithm directly on the converted dataset to discover the class for the testing data.

4.2.3 K Nearest Neighbor algorithm (KNN)

In this paper, we consider the calculated probability distribution function as certain data and we utilizes the calculated pdf for calculating the distance measure which is presented in the following section.

4.2.4 Calculation of Distance Measure

For classification purpose, we calculate the pdf for each attribute in testing tuple. Once we calculated the pdf for testing data, the next step is calculation of distance measure between testing data with all other training data. In this paper, we calculate the distance using the Euclidian distance measure.

Once we calculate the distance of testing data with all other training data, we selects K number of training data with its class which are having minimum distance with testing data.

From the K number of selected training data, the next step is to calculate count value of each class. Once we calculated the frequency of the class from the selected number of training data subsequently algorithm selects class for testing data, which has more frequency among the available.

Algorithm 2

Input: Uncertain database
Output: Classified output

Parameters
UDB- Uncertain database
UDB_R- Training part of uncertain database
UDB_{TR}- Testing part of uncertain database
t_m- Tuple data, where m represents the attribute and n represents the tuple
A_m- attribute m
p_1- Set of tuple data which are less or equal than split point z ($p_1 = \{d\} \leq z$)
p_2- Set of tuple data which are greater than split point z ($p_2 = \{d\} > z$))
$H(z, A_m)$- Fuzzy entropy of split point z from the attribute A_m
$pdf(t_m)$ - Probability distribution function value of t_m
minN($\{z\}$)- selecting N number of split points which has minimum entropy value
K((t_n)) - selecting K number of tuples which has minimum distance

Begin
{
Read UDB_R and UDB_{TR}
For each A_m
For each d_i
If ($d_i = \{t_m\} \leq z$)
$d_i(t_m)$à p_1
Else
$d_i(t_m)$à p_2
Calculate entropy $H(z, A_m)$ equation (8)
End for
Select à minN($\{z\}$)
End for
For each t_m
Calculate $pdf(t_m)$ w.r.t {z} Equation (9)
Remove all T from t_m
Place t_mà $pdf(t_m)$
End for
For each t_m from UDB_{TR}
Calculate distance with all t_n from UDB_R Equation (10)
Select min distance à K(t_n)
Calculate $Cnt(c_i)$
Select max$Cnt(c_i)$
Assign class for t_n
End for
}
End

5. Experimental Result and Discussion

The experimental analysis discusses the performance of the proposed algorithms for classifying the uncertain data. The performance of our proposed algorithms is compared with the decision tree algorithm for classification of uncertain data[24]. In order to compare our proposed algorithm in this paper, we have selected the four real time datasets such as Iris, Glass, Breast cancer and Ionosphere that are taken from the UCI dataset repository[15].

Table 1. Shows that the details of the dataset

Dataset	Training tuples	Number of attributes	Number of classes	Test tuples
Iris	120	4	3	30
Glass	171	9	6	43
Breast cancer	455	30	2	114
Ionosphere	280	32	2	71

5.1 Performance Evaluation based on Accuracy

In this section, we evaluate our proposed classification algorithms with the uncertain decision tree algorithm UDT[14] based on accuracy with the four dataset mentioned above. The real time uncertain data is not available on the web hence we convert the certain dataset into uncertain dataset through distributions. In this paper, we evaluated the classification algorithm by varying the distribution parameters such as width W and U.

Consider the certain data value as 50 and the value of W is 10 then the width value is 10% of 50 is 5. Then the uncertain range value becomes 45 to 55. The value U defines the number of data values present in the calculated range. For each dataset, we calculate average accuracy and best case of accuracy by varying the value of W and U value constant. In another case, we vary the value of

U and W and we calculate the best case of accuracy of classification algorithms.

5.2 Evaluation of Accuracy on Iris Dataset

The following Figure 2 represents the average accuracy of proposed algorithms and UDT algorithm.

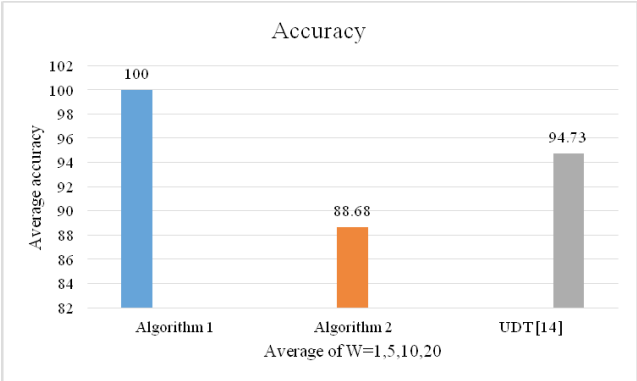

Figure 2. Evaluation of classification algorithms based on average accuracy for various values of W and constant value of U.

Here, we vary the values of W at 1, 5, 10, 20 and we make the U value constant and the results are represented at graph value. By analyzing the Figure 2, we state that our proposed algorithm 1 has achieved 100 percent of average accuracy, which is more than UDT algorithm 94.73, but the algorithm 2 performs 88.63 percent of average accuracy, which is less than UDT algorithm.

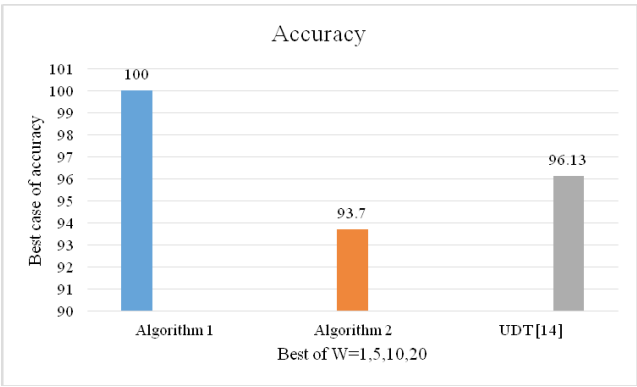

Figure 3. Evaluation of classification algorithms based on best case of accuracy for various values of W and constant value of U.

The above Figure 3 represents the best case of accuracy of proposed algorithms and UDT algorithm. Here, we vary the values of W at 1, 5, 10, 20 and we make the U value constant and the best case of accuracy results of the classification algorithms are represented at graph value. By analyzing the Figure 3, it is clear that our proposed algorithm 1 has achieved 100 percent best case of accuracy, which is more than UDT algorithm 93.0, but the algorithm 2 performs 90 percent of accuracy, which is only 3 percent lesser than UDT algorithm.

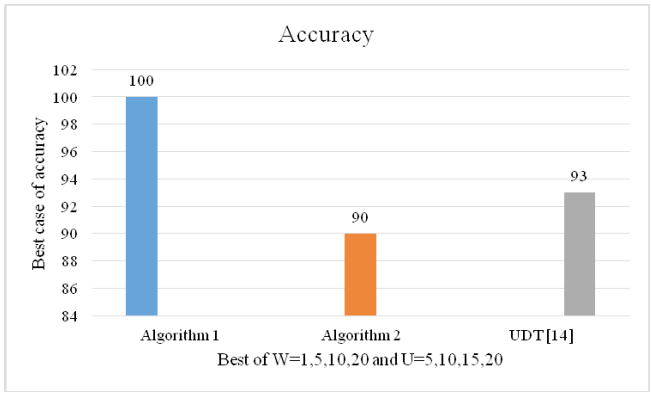

Figure 4. Evaluation of classification algorithms based on best case of accuracy for various values of W and U.

The above Figure 4 represents the best case of accuracy of proposed algorithms and UDT algorithm. Here, we vary the values of W at 1, 5, 10, 20 and varying the U value at 5, 10, 15, 20 and the best case of accuracy results of the classification algorithms are represented at graph value. By analyzing the above Figure 4, we can say that our proposed algorithm 1 has achieved 100 percent of best case of accuracy, which is more than UDT algorithm 93.0, but the algorithm 2 performs 90 percent of best case of accuracy, which is only 3 percent lesser than UDT algorithm.

5.3 Evaluation of Accuracy on Glass Dataset

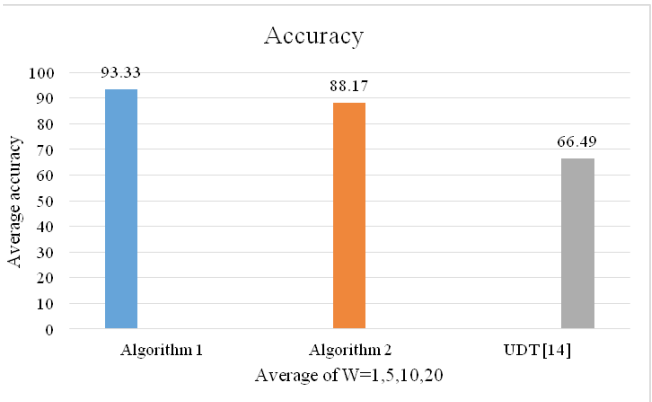

Figure 5. Evaluation of classification algorithms based on average accuracy for various values of W and constant value of U in glass dataset.

The above Figure 5 represents the average accuracy of proposed algorithms and UDT algorithm. Here, we vary the values of W at 1, 5, 10, 20 and we make the U value constant and the results are represented at graph value. By analyzing the above figure 5, our proposed algorithm 1, and algorithm 2 has achieved better results (93.33 and 88.17) than the existing UDT algorithm (66.49). Our proposed algorithms are well suited for the glass dataset and we proved that our proposed uncertain classification algorithms performed well than the UDT algorithm in terms of average accuracy.

The following Figure 6 represents the best case of accuracy of proposed algorithms and UDT algorithm. Here, we varying the values of W at 1, 5, 10, 20 and we make the U value constant and the best case of accuracy results of the classification algorithms are represented in the following graph. By analyzing the following Figure 6, our proposed algorithm 1 and algorithm 2 has achieved best case of accuracy value 95.15 and 96.68 respectively. The best case of accuracy of the proposed classification algorithms are performed well than the UDT algorithm (72.75 percent).

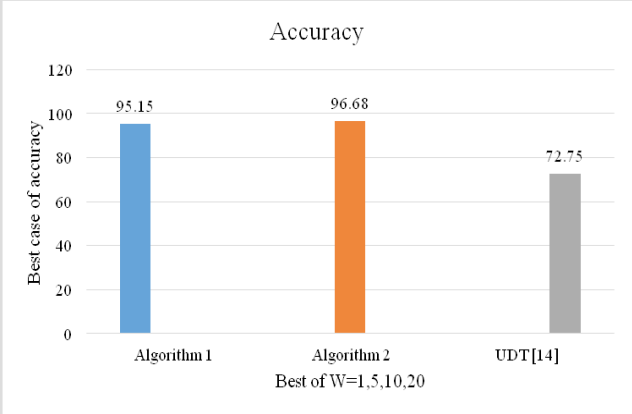

Figure 6. Evaluation of classification algorithms based on best case of accuracy for various values of W and constant value of U.

The above Figure 7 represents the best case of accuracy of proposed algorithms and UDT algorithm in glass dataset. Here, we vary the values of W at 1, 5, 10, 20 and vary the U value at 5, 10, 15, 20 and the best case of accuracy results of the classification algorithms are represented in the following graph. By analyzing the figure 7, it is clear that our proposed algorithm 1 and algorithm 2 achieved same best case of accuracy value 99.66 percent, which is nearly 100 percent. The best case of accuracy of the proposed classification algorithm is better than the UDT algorithm (93 percent).

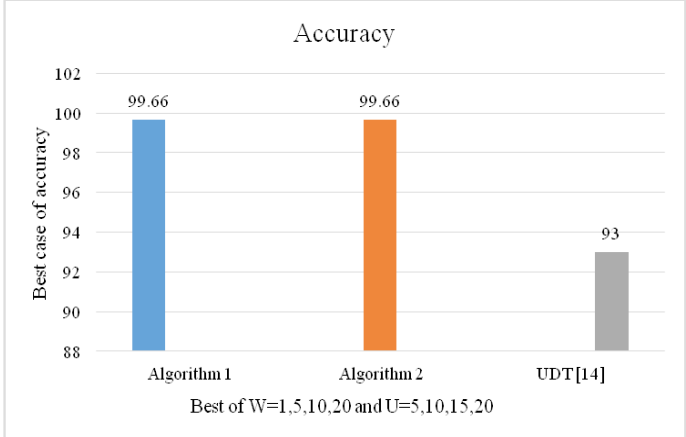

Figure 7. Evaluation of classification algorithms based on best case of accuracy for various values of W and U in glass dataset.

5.4 Evaluation of Accuracy on Breast Cancer Dataset

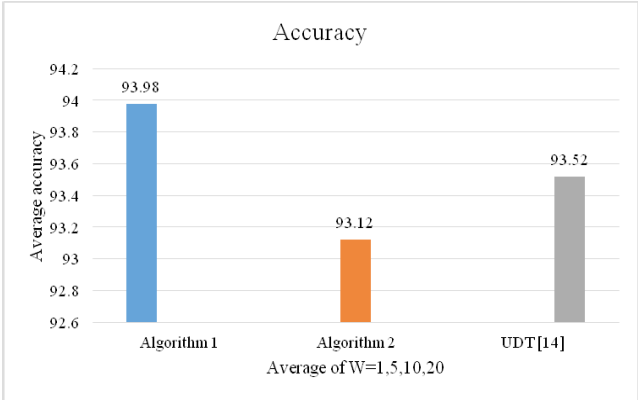

Figure 8. Evaluation of classification algorithms based on average accuracy for various values of W and constant value of U in breast cancer dataset.

The above Figure 8 represents the average accuracy of proposed algorithms and UDT algorithm breast cancer dataset. Here, we vary the values of W at 1, 5, 10, 20 and we make the U value constant and the results are represented at graph value. By analyzing the Figure 8, it is find out that our proposed algorithm 1 has achieved 93.98 percent of average accuracy, which is more than UDT algorithm 93.52, but the algorithm 2 performs 93.12 percent of average accuracy, which is slight lesser than UDT algorithm.

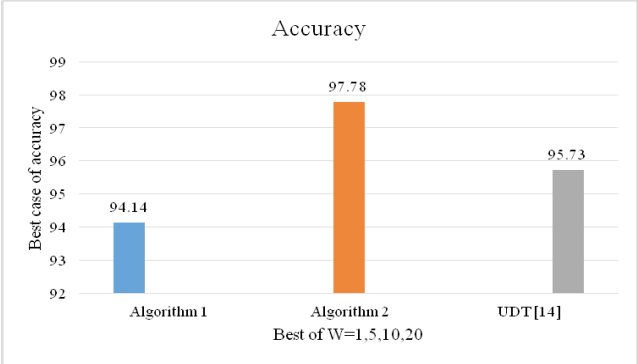

Figure 9. Evaluation of classification algorithms based on best case of accuracy for various values of W and constant value of U breast cancer dataset.

The above Figure 9 represents the best case of accuracy of proposed algorithms and UDT algorithm in breast cancer dataset. Here, we vary the values of W at 1, 5, 10, 20 and we make the U value constant and the best case of accuracy results of the classification algorithms are represented at graph value. By analyzing the above Figure 9, our proposed algorithm 2 has achieved 97.78 percent best case of accuracy, which is more than UDT algorithm 93.0, but the algorithm 1 performs 94.14 percent of accuracy, which is only 1.59 percent lesser than UDT algorithm.

The following Figure 10 represents the best case of accuracy of proposed algorithms and UDT algorithm in breast cancer dataset. Here, we vary the values of W at 1, 5, 10, 20 and varying the U value at 5, 10, 15, 20 and the best case of accuracy results of the classification algorithms are represented at graph value. By analyzing the following figure 10, it can be seen that our proposed algorithm 1 and algorithm 2 are achieved 99.663 percent of best case of accuracy, which is more than UDT algorithm 93.0.

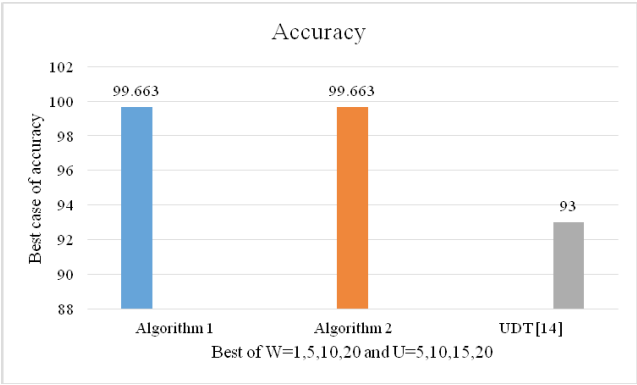

Figure 10. Evaluation of classification algorithms based on best case of accuracy for various values of W and U breast cancer dataset.

5.5 Evaluation of Accuracy on Ionosphere Dataset

The following Figure 11 represents the average accuracy of proposed algorithms and UDT algorithm in ionosphere dataset. Here, we vary the values of W at 1, 5, 10, 20 and we make the U value constant and the results are represented at graphs. By analyzing the following Figure 11, our proposed algorithm 1 has achieved 98.24 percent of average accuracy, which is 9.55 percent more than UDT algorithm (88.69), and the algorithm 2 performs 94.98 percent of average accuracy, which is 10.29 percent more than UDT algorithm. Our proposed uncertain data classification algorithm performed well than the UDT algorithm in terms of average accuracy and we proved our algorithm is well suitable for ionosphere dataset.

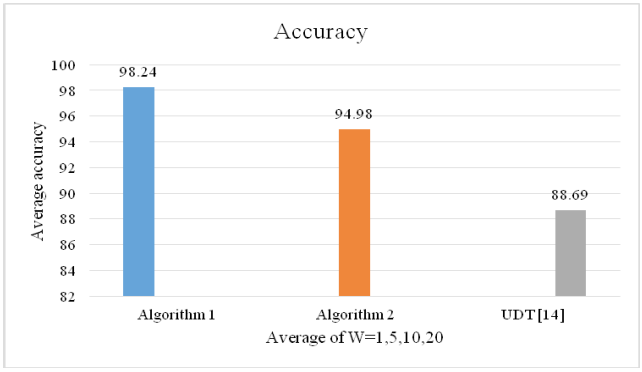

Figure 11. Evaluation of classification algorithms based on average accuracy for various values of W and constant value of U in ionosphere dataset.

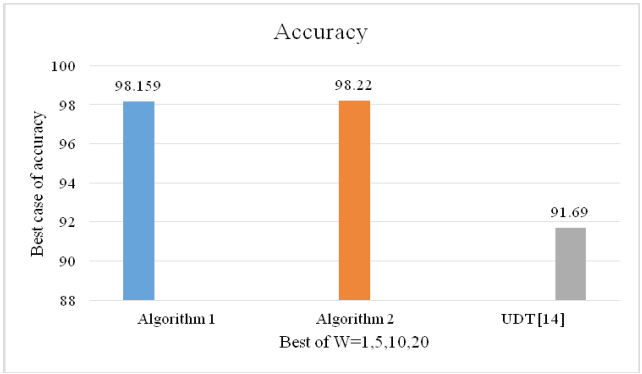

Figure 12. Evaluation of classification algorithms based on best case of accuracy for various values of W and constant value of U in ionosphere dataset.

The above Figure 12 represents the best case of accuracy of proposed algorithms and UDT algorithm.

Here, we vary the values of W at 1, 5, 10, 20 and we kept the U value constant and the best case of accuracy results of the classification algorithms are represented at graph value. By analyzing the following Figure 12, our proposed algorithm 1 has achieved 98.15 percent best case of accuracy, which is more than UDT algorithm 91.69, and the algorithm 2 performs 98.22 percent of accuracy, which is only 3 percent lesser than UDT algorithm. Our proposed uncertain data classification algorithm performed well than the UDT algorithm in terms of best case of accuracy and we proved our algorithm is well suited for ionosphere algorithm.

The following Figure 13 represents the best case of accuracy of proposed algorithms and UDT algorithm in Ionosphere Dataset. Here, we vary the values of W at 1, 5, 10, 20 and vary the U value at 5, 10, 15, 20 and the best case of accuracy results of the classification algorithms are represented at graph value. By analyzing the following Figure 13, we say that our proposed algorithm 1 has achieved 99.167 percent of best case of accuracy, which is 7.167 percent more than UDT algorithm 93.0, and the algorithm 2 performs 99.11 percent of best case of accuracy, which is 7.11 percent more than UDT algorithm. For the ionosphere dataset, our proposed algorithms are performed almost similar best case of accuracy.

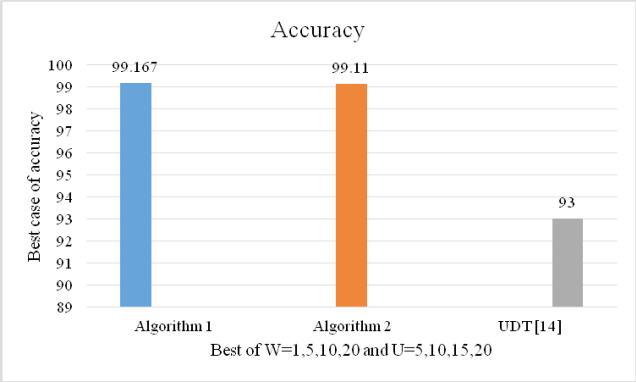

Figure 13. Evaluation of classification algorithms based on best case of accuracy for various values of W and U in ionosphere dataset.

6. Conclusion

In this paper, we presented two algorithms using the KNN classifier, which is used for the uncertain dataset to predict their corresponding class. In the first algorithm, we used the properties of uncertain data to convert the uncertain data into certain data. The algorithm 1 initially converted the uncertain data to certain data using the statistical properties of the data.

The second algorithm converted the uncertain data through probability distribution function (pdf). The algorithm 2, initially calculated the N number of split point for each attributes of the training part of uncertain data then it calculated pdf with respect to the selected split point. Finally, we evaluated our proposed algorithm with the UDT (uncertain decision tree) algorithm with the four real datasets such as iris dataset, ionosphere dataset, breast cancer dataset, glass dataset and we proved that our proposed algorithms performed well and give better results than the UDT algorithm in terms of accuracy.

7. References

1. Zhou Y, Youwen L, Shixiong X. An improved KNN text classification algorithm based on clustering. Journal of Computers. 2009; 4(3):230–7.
2. Romero C, Ventura S, Espejo PG, Hervas C. Data mining algorithms to classify students. Proceedings of the 1st Int'l conference on educational data mining; Canada. 2008. p. 8–17.
3. Zhang J, Mani I. kNN approach to unbalanced data distributions: A case study involving information extraction. Proceedings of the 20th International Conference on Machine Learning (ICML); Workshop on Learning from Imbalanced Data Sets II. 2003 Aug. p. 1–7.
4. Xindong W, Vipin K, Ross, Joydeep G, Qiang Y, Hiroshi Motoda, Geoffrey Mclachlan, Angus Ng, Bing Liu, Philip Yu, Zhi-Hua Zhou, Michael Steinbach, David Hand, Dan Steinberg. Top 10 algorithms in data mining. Knowledge and Information Systems. 2008; 14(1):1–37.
5. Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys. 2002; 34(1):1–47.
6. Han E, Karypis G. Centroid-based document classification analysis and experimental result. Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery; 2000. p. 424–31.
7. Tan S. An effective refinement strategy for KNN text classifier. Expert Systems with Applications. 2006; 30:290–8.
8. Gao, Yunlong, et al. A novel two-level nearest neighbour classification algorithm using an adaptive distance metric. Knowledge-Based Systems. 2012; 26:103–10.
9. Jiang S, Pang G, Wu M, Kuang L. An improved K-nearest-neighbour algorithm for text categorization. Proceedings of the 2nd International Conference on Applied Informatics and Computing Theory; 2011.p. 190–4.
10. Lijun W, Zhao X. Improved KNN classification algorithms research in text categorization. 2nd International

Conference on Consumer Electronics, Communications and Networks. IEEE; 2012. p. 1848–52.
11. Zhao S, Rui C, Zhang Y. MICkNN: Multi-Instance Covering kNN Algorithm. Tsinghua Science and Technology. 2013; 18(4):360–8.
12. Saini I, Singh D, Khosla A. QRS detection using < i > K < /i >-Nearest Neighbor Algorithm (KNN) and evaluation on standard ECG databases. Journal of Advanced Research 4.4. 2013. p. 331–44.
13. Tsang S, Kao B, Yip KY, Ho W-S, Lee SD. Decision tress for uncertain data. IEEE transactions on knowledge and data engineering. 2011; 23(1):64–78.
14. UCI dataset repository Available from: http://archive.ics.uci.edu/ml/datasets.html
15. Liu W, Chawla S. Class confidence weighted KNN algorithms for imbalanced data sets. Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science. 2011; 66(35):345–6.
16. Gong A, Liu Y. Improved KNN classification algorithm by dynamic obtaining K. Advanced Research on Electronic Commerce, Web Application, and Communication, Communications in Computer and Information Science. 2011; (143):320–4.
17. Zhou L, Wang L, Ge X, Shi Q. A clustering-based KNN improved algorithm CLKNN for text classification. 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR). 2010 Mar; 3(6-7): 212–5.
18. Gaoa Y, Panb J, Jia G, Yangc Z. A novel two-level nearest neighbor classification algorithm using an adaptive distance metric. Knowledge-based Systems. 2012 Feb; 26:103–10.
19. Agrawal R, Ram B. A survey of uncertain data mining techniques. International Journal of Advance Research in Education, Technology and Management. 2015 Apr; 3(1):252–56.
20. Agrawal K. K- Nearest Neighbor for Uncertain Data. International Journal of Computer Applications (0975 – 8887). 2014 Nov; 105(11):1–13.