International Journal of Computational and Theoretical Statistics
ISSN (2210-1519)
Int. J. Comp. Theo. Stat. 6, No. 1 (May-2019)

http://dx.doi.org/10.12785/ijcts/060101

Quantiles based Neighborhood Method of Classification

S. Sampath[1] and S. Suresh[2]

[1,2] *Department of Statistics, University of Madras, Chennai 600 005, India*

Received August 23, 2018, Revised January 21, 2019, Accepted February 17, 2019, Published May 01, 2019

Abstract: Classification of objects is an important problem that has received the attention of several researchers in Data Mining. Necessity for classification of an object into one of the predefined classes arises in several domains of research which include market research, document classification, diagnosing the presence of disease etc. A widely studied and applied popular classifying method which has attracted many data mining researchers is k-nearest neighbor algorithm. It is a distance based algorithm in which classification of an object is done on the basis of the memberships of its neighboring objects. The main problem one faces in the application classification is deciding a suitable value for the neighborhood parameter. In this paper, a method similar to classification in which the number of neighbors to be used in the classification process is determined by the distribution of distances between units in the training set has been proposed. Performance of the proposed method has been studied using simulated multivariate normal data sets as well as some benchmark data sets.

Keywords: Classification, Neighborhood, Error rate, Training set, Test set

1. INTRODUCTION

Fix and Hodges [9] introduced the nearest neighbor algorithm to determine the class of an object based on the concept of nearest neighbor. Followed by this work, several neighborhood based methods have been developed. The k-nn algorithm is one of the simplest algorithms among all machine learning algorithms. Cover and Hart [7] proposed the k-nn method of classification where the result of new instance query is classified based on majority of k-nn category, k being a positive integer, typically small. It may be noted that, if $k=1$ then the object is simply assigned the class of its nearest neighbor. The purpose of this algorithm is to classify a new object based on memory.

Enas and Choi [8] compared the efficiency of k-nn algorithm with linear discriminant function and logistic regression. Yingquan, Krassimir and Govindaraju [15] proposed two effective techniques, namely, template condensing and preprocessing, to significantly speed up k-nn classification while maintaining the level of accuracy. Chang and Chen [6] proposed nearest neighbor classification with cam weighted distance. The experimental results show that cam weighted distance nearest neighbor classification method gives better classification performance for most of the benchmark data sets. Parvin, Alizadeh and Behrouz [13] introduced a modified k-nn algorithm by using a weighted distance concept in order to reduce the error rate in the classification process. Experimental studies carried out using five different data sets have shown considerable improvement in accuracy when compared to the conventional k-nn method. Liu, Zhang and Mo [12] proposed a new learning algorithm based on k-nn . The algorithm adopted mutual nearest neighbors, rather than k-nn , to determine the class labels of unknown instances. The experimental results based on UCI repository data sets reveal that the performance achieved by the proposed method is better than the classical k-nn methods. Zhenyun, Zhu, Cheng and Zhang [16] investigated the recent progress on big data using k-nn method.

Govindarajan and Chandrasekaran [11] considered k-nn algorithm for a classification problem related to direct marketing data set. Akhil Jabbar, Deekshatulu and Chandra [2] studied the utility of k-nn classification method using various machine learning data sets taken from UCI repository. Bhuvaneswari and Brintha Therese [4] proposed genetic k-nn algorithm for early stage lung cancer detection. Zhongheng [17] made detailed studies on factors such as the

E-mail address: sampath1959@gmail.com, sureshstat22@gmail.com

value of k, distance calculation and other aspects using R software. Ashish and Bijnan [3] made an attempt to map the protein secondary structure prediction problem as pattern classification problem using k-nn. Yihua Liao and Rao[14] developed a new approach, based on the k-nn classifier for intrusion detection. Bruno, Sasa and Dzenana [5] examined the possibility of using k-nn algorithm with Term Frequency-Inverse Document Frequency (TF-IDF) method to develop a framework for text classification.

It is pertinent to note that the performance of k-nn classifier depends heavily on the choice of k. The choice of k is based on the dimension of the data vector, the size of the data set and the covariance structure of the underlying variables. Studies related to the choices of k have been carried out by some researchers. Based on experimental studies, Ghosh [10] concluded that most of the popular cross validation techniques for existing classifiers often fail to identify an optimal value for k in the presence of multiple minimizers of the estimated misclassification rate. Motivated by this, Ghosh [10] proposed a Bayesian method to solve the problem of choosing the value of k and also tested its usefulness with some bench mark data sets. The experimental results revealed that the method proposed by Ghosh [10] identifies the value of k which leads to enhanced classification performance. Ahmad, Ali and Altarawneh [1] identified another solution depending on the idea of ensemble learning, for choosing the parameter k in the k-nn algorithm. The efficiency of the proposed method has been studied with the help of 28 different data sets taken from the UCI Machine Learning Repository.

It may be noted that mere increase in the value of k does not guarantee decrease in error rate. The following diagrams which give average error rates corresponding to different values of k for *iris* and *urine* data sets justify this observation.

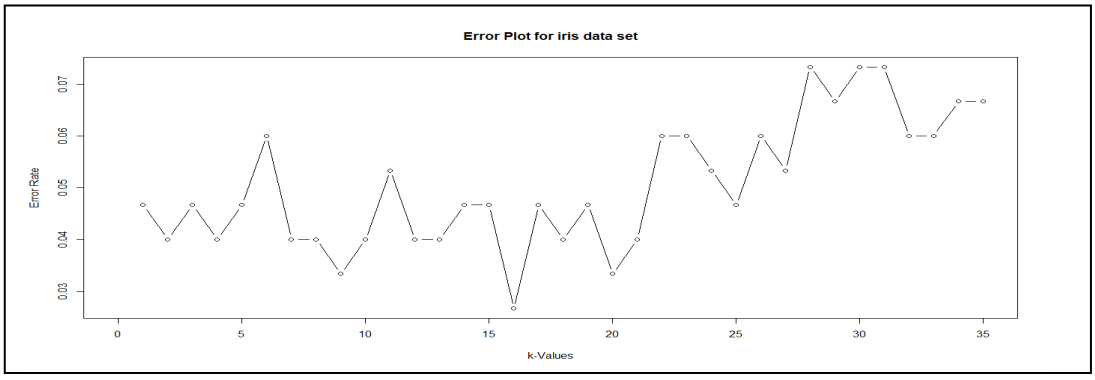

Figure 1. Error rate corresponding to different values of k for Iris data set

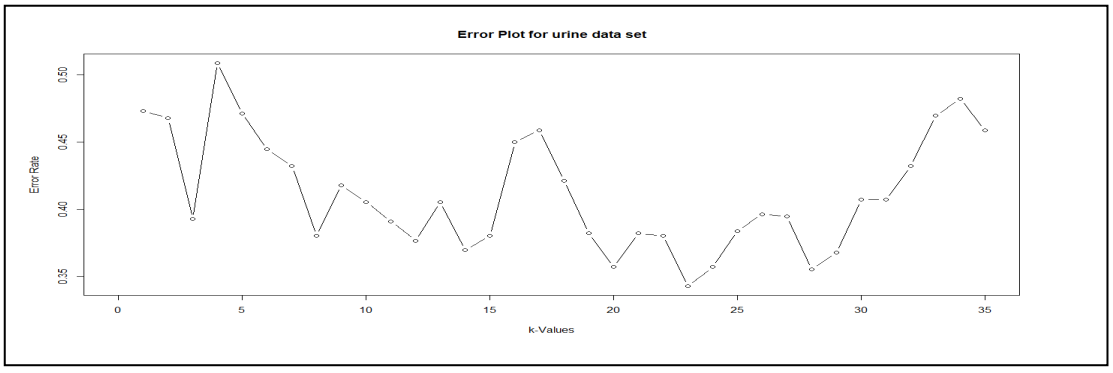

Figure 2. Error rate corresponding to different values of k for Urine data set

The error rates are obtained using default calls of *tune.knn* function available in R-package *'e1071'*. It is clear that values of *k* and error rates do not seem to have any correlation.

Existing neighborhood methods do not take in to account the distribution of distances between training set objects in deciding the value of k. In this paper, we propose a new approach based on distances, which takes into account the distribution of distances between training set objects.

This paper is organized as follows: The second section of the paper gives a complete description on k-nn algorithm and introduces a new classification procedure. The third section is devoted for the problem of choosing the optimum value of the neighborhood parameter k based on extensive experimental studies. The final and concluding section discusses the outcomes of the experimental results carried out with the help of simulated and bench mark data sets and gives certain recommendations for the practitioners when it is desired to use the proposed method.

2. NEW QUANTILE BASED CLASSIFICATION

A. *Existing k-nn Algorithm*

The nearest neighbor classifiers require no preprocessing of the labeled sample set (training set) prior to their use. The nearest neighbor classification rule assigns an input sample vector y, (test object) with unknown class label to the class corresponding to its nearest neighbor. This idea can be extended to the k-nn with vector y being assigned to the class that is represented by a majority among the k nearest neighbors. The main steps of the k-nn algorithm are as follows:

1. Assume there are *N* training objects where each object has *t*- attributes and an object can belong to only one of the m-classes.
2. Let *O* be an object to be classified.
3. Compute the distances between the object *O* and each of the training objects.
4. Let $d_1, d_2, \ldots, d_N$ be the resulting distances.
5. Arrange the distances in ascending order and identify first k objects corresponding to the first k smallest distances to get the set C_k.
6. Let x_r denote the number of objects in the set C_k which belong to the class $r, (r = 1,2,\ldots,m)$. We assign the object to the class λ if, $x_\lambda = \max(x_1, x_2, \ldots, x_m)$

B. *Proposed Method*

The conventional k-nn classifier makes use of smallest k distances between the test object and training objects in the classification process. It is to be noted that objects with the smallest distance need not necessarily be members of the same class. There are many real life situations where objects may belong to different classes but still have smaller distance. Hence, it is sensible to devise a classification method which takes in to account the class distribution based on pairs of objects with significantly smaller distances. The proposed method makes use of all possible pairs of objects having distances smaller than a pre-specified level as determined by training set members in the classification task. A test set object is assigned to the class corresponding to maximum votes by taking in to account the objects in the training set having a distance less than a pre-specified value. Steps involved in the classification process are summarized below.

Given a training data set D consisting of n objects with m classes and a pre specified $p \in (0,1)$, determine the threshold value θ_p using the following steps.

- Compare all the n_{C_2} distances between the pairs of objects in the data set D.
- Identify all distances between pairs of objects which belong to the same class and order such distances between matched pairs in ascending order.
- Compute the quantile of order p using the ordered distances obtained in the previous step.
- Take the quantile value identified in the previous step as θ_p.
- A test object O is assigned to the class c by proceeding as follows:
  - Compute the distances between test object and every object in the training set D.
  - Identify the objects in the training data set having distances less than or equal to θ_p and compute the proportions $p_1, p_2, .., p_m$ where p_i is the proportion objects in the training set belonging to the class $c(c = 1,2,..,m)$.

- The test object is assigned to the class c if p_c is maximum.

A careful analysis of the algorithm proposed in this section leads to the conclusion that the choice of the threshold θ_p plays a vital role in the process of classification. Intuitively it is clear that there cannot be a globally best choice for θ_p and the nature of the data set used in the study will have significant impact on the choice of. In the following section, a comprehensive experimental study on the choice of θ_p which gives smaller error rate has been carried out with the help of a variety of simulated and bench mark data sets.

3. EXPERIMENTAL STUDIES

In this section, it is proposed to make a detailed study on the choice of θ_p using data sets simulated from multivariate normal distributions with different parametric settings and some popular bench mark data sets.

A. *Simulation Based Study*

In the simulation study a wide variety of trivariate data sets where each data set contains objects from a predetermined number of classes are considered. Data vectors corresponding to a given class are identified with the help of the mean vector used in the simulation. Construction of a data set representing m classes requires m number of three component vectors. For each choice of the mean vector a predetermined number of observations from a trivariate normal population are obtained by fixing the variance covariance matrix. For example, the three vectors $[35\ 30\ 25], [30\ 25\ 20]$ and $[50\ 45\ 40]$ can be taken as mean vectors of three multivariate normal distributions in order to construct a three class data set. The data vectors simulated from the distribution with mean vector $[35\ 30\ 25]$ can be treated as values of objects belong to the first class. Similarly the data vectors simulated using the remaining two mean vectors namely, $[30\ 25\ 20]$ and $[50\ 45\ 40]$ can be treated as the data vectors corresponding to the second and third classes.

In the present study, the following four sets of mean vectors are considered for simulation of data sets.

Set 1: $[35\ 30\ 25], [30\ 25\ 20]$ and $[50\ 45\ 40]$

Set 2: $[35\ 30\ 25], [30\ 25\ 20]$ and $[40\ 35\ 30]$

Set 3: $[35\ 30\ 25], [30\ 25\ 20]$ and $[32\ 27\ 23]$

Set 4: $[31\ 29\ 25], [30\ 28\ 26]$ and $[32\ 27\ 23]$

The components of the mean vectors in a group are determined so that the amount of separation between the data vectors with respect to the three classes could be gauged. For example, the parameters in the first data set can be expected to give data vectors from three classes with good amount of separation. On the other hand, the last set is expected to provide data vectors from the three classes which have poor separation. The present study will focus on the influence of the separation between data values in deciding the optimal (minimum error rate) percentile order. In order to quantify the amount of separation of a simulated data set consisting of objects, we define an index called **Seperation Index (SI)** as follows.

B. *Seperation Index*

The separation index of a training data set consisting of objects from *m* classes is defined as

$$SI = 1 - \frac{1}{m}\sum_{i=1}^{m}\frac{d_i}{d} \tag{1}$$

where d_i is the maximum of the distances among objects belonging to the class $C_i (i = 1,2,\ldots,m)$ and d is the maximum of the distances among all the objects in the training set. It may be noted that the separation index always lies between 0 and 1. A value closure to 1 indicates the data objects coming from different classes being considered are well separated.

The separation indices for data sets simulated using the four sets of parametric values involved in the simulation for data sets as defined earlier were found to be approximately, 0.70, 0.50, 0.25 and 0.20. A higher value of SI indicates

the objects belonging to different classes are well separated and a smaller value of SI indicates the objects are not well separated.

It is to be admitted that the magnitudes of the components of the mean vectors alone do not decide the margin of separation. The variances and covariances between the components are likely to have significant impact on the quantum of separation, Hence, it is necessary to assign the variance covariance matrices for the distribution in a systematic manner so that meaningful conclusions could be drawn from the results arrived at. In this study, variance covariance matrices are constructed by suitable modification on the elements of matrices of the form $\begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{bmatrix}$ where ρ is assigned values from -0.5 to 0.9 in steps of 0.1. Throughout this study, it is assumed that the variances of three components are $\sigma_{11}=4, \sigma_{22}=9$ and $\sigma_{33}=5$. A variance covariance matrix can be constructed by multiplying the three rows by the standard deviations of the three variables taken in order and then repeating the same operation for columns again in the same order. It is to be noted that, for $-0.9<\rho<-0.5$, the variance covariance matrices constructed in the above fashion cease to be positive definite. Hence those values are eliminated from the study. The experimental study assumes the variance covariance matrices are equal for all the three trivariate normal populations considered in simulating a data set representing three classes.

As mentioned earlier, the objective of the study is the identification of the order of percentile which leads to minimum error rate. In order to reach credible conclusions regarding the optimal choice (minimum error rate) of θ_p, thirty test data sets have been simulated with same parametric setting as used in the construction of training set. For each level p, the algorithm proposed in this work is applied for each one of the thirty test data sets and the resulting error rates are computed. Thus for a given value of p, thirty values are available for the error rates. The average of these thirty values is taken as the error rate corresponding to the usage of a given level p. On scrutinizing these error rates for different values of p, we conclude that the value of p for which the error attains minimum as the optimal error rate. This process is presented below in a summarized comprehensible form.

Step 1: For a given choice of three mean vectors and variance covariance matrix generate 30 data sets consisting of given number of objects with predetermined class distribution of objects.
Step 2: For each one of the thirty data sets, compute the error rates corresponding to every level of percentiles used, namely 0.1(0.1) 0.9.
Step 3: Average the error rates for every level of percentiles and identify the level of percentiles which makes the error rate minimum.

The entries in Table I are based on the thirty test data sets simulated with the given set of mean vectors mentioned under Set 1 and common variance covariance matrix generated using a specific ρ value. It gives the minimum error rates and the corresponding orders of percentiles when equal number of objects namely 20,30,40 and 50 are represented in the training data set as per the specification of the parametric value listed in Set 1. That is, the three mean vectors are $[35\ 30\ 25], [30\ 25\ 20]$ and $[50\ 45\ 40]$. In the case of 20 objects from each class, the order of percentile yielding minimum rate is either 0.7 or 0.8. When the number of objects represented are either 30 or 40 the optimal order of percentile is either 0.6 or 0.7. In the last case, where 50 objects are generated from each class the optimal order happens to 0.6, 0.7 or 0.8.

TABLE I. MINIMUM ERROR RATES FOR DATA SET WITH SEPARATION INDEX 0.7

ρ	Separation Index 0.7								
	$N=20$		$N=30$		$N=40$		$N=50$		
	0.7	**0.8**	**0.6**	**0.7**	**0.6**	**0.7**	**0.6**	**0.7**	**0.8**
-0.5	0.007	0.006	0.014	0.01	0.005	0.005	0.004	0.005	0.006
-0.4	0.012	0.011	0.006	0.011	0.007	0.007	0.006	0.007	0.005
-0.3	0.009	0.009	0.007	0.008	0.006	0.005	0.004	0.005	0.01
-0.2	0.01	0.009	0.011	0.005	0.006	0.004	0.005	0.005	0.009
-0.1	0.008	0.008	0.011	0.011	0.006	0.007	0.005	0.004	0.007

0	0.006	0.003	0.006	0.005	0.009	0.006	0.004	0.005	0.003
0.1	0.011	0.008	0.009	0.005	0.007	0.006	0.006	0.004	0.006
0.2	0.007	0.009	0.005	0.006	0.006	0.007	0.007	0.006	0.012
0.3	0.007	0.009	0.006	0.006	0.003	0.003	0.005	0.003	0.004
0.4	0.005	0.006	0.004	0.005	0.005	0.003	0.005	0.006	0.005
0.5	0.008	0.004	0.007	0.004	0.007	0.009	0.004	0.006	0.005
0.6	0.007	0.006	0.006	0.007	0.009	0.01	0.003	0.006	0.005
0.7	0.007	0.011	0.01	0.014	0.004	0.003	0.01	0.007	0.006
0.8	0.007	0.003	0.01	0.007	0.004	0.006	0.003	0.003	0.003
0.9	0.006	0.005	0.006	0.008	0.004	0.008	0.006	0.008	0.008
KW test	0.29676		0.23286		0.00708		1.9103		
df	1		1		1		2		
$p-value$	0.5859		0.6294		0.9329		0.3848		

One can see from the above table, for a given value of ρ, error rates attain minimum value for different order of percentiles. Hence, a clear cut recommendation for the choice of percentile to be used cannot be given based on the entries of the above table. Hence, it is decided to examine whether the differences in the error rates are statistically significant with respect to the optimal choices of percentiles in each one of the four cases. Towards this, it is decided to perform one way analysis of variance for the four cases considered in this study. The results of analysis of variance are presented below.

The above table gives the results of one way analysis of variance performed using the four data sets extracted from the columns of Table I. For example, the two columns corresponding to the levels 0.70 and 0.80 with $N = 20$ have been used in the first ANOVA. Similarly the columns corresponding to the levels 0.60 and 0.70 with $N = 30$ and $N = 40$ are used in the second and third ANOVA. Finally, the three columns corresponding to the levels 0.60, 0.70 and 0.80 for $N = 50$ have been used in the last ANOVA. In all the four cases the ρ values are on the higher side which leads us to conclude that there is no significant difference between the error rates corresponding to optimum levels of percentiles. Hence, we conclude that the user can make use of any order of percentile listed in the columns of Table I. Such usage is unlikely to affect the performance of the classifier in terms of the error rate. For example, when $N = 20$, either 0.7 or 0.8 (or possibly their average 0.75) can be taken as the optimal order of percentile Similar conclusions can be arrived at for the remaining three cases as well. Instead of leaving several options to the user, we recommend the average of the optimal percentile for the implementation of the algorithm, Thus for the cases, $N = 20,30,40 \text{ and } 50$ we recommend 0.75, 0.65, 0.65 and 0.70 respectively irrespective of the value of ρ.

Table II given below provides the optimal percentiles along with error rate computed by adopting the methodology followed using the parametric settings listed in Set 2.

TABLE II. MINIMUM ERROR RATES FOR DATA SET WITH SEPARATION INDEX 0.5

	Separation Index 0.5								
ρ	$N = 20$		$N = 30$		$N = 40$		$N = 50$		
	0.7	**0.8**	**0.6**	**0.7**	**0.6**	**0.7**	**0.5**	**0.6**	**0.7**
-0.5	0.022	0.024	0.008	0.009	0.01	0.008	0.016	0.012	0.015
-0.4	0.014	0.018	0.01	0.008	0.011	0.018	0.009	0.01	0.01
-0.3	0.016	0.028	0.019	0.018	0.008	0.009	0.006	0.009	0.01
-0.2	0.012	0.017	0.009	0.011	0.012	0.011	0.01	0.011	0.01
-0.1	0.018	0.013	0.011	0.018	0.013	0.012	0.012	0.012	0.009
0	0.013	0.028	0.014	0.017	0.008	0.009	0.012	0.013	0.013
0.1	0.017	0.013	0.009	0.014	0.011	0.013	0.011	0.01	0.008

0.2	0.023	0.017	0.012	0.009	0.012	0.016	0.011	0.011	0.013
0.3	0.02	0.019	0.011	0.009	0.009	0.01	0.012	0.013	0.013
0.4	0.013	0.018	0.013	0.02	0.017	0.011	0.013	0.01	0.01
0.5	0.013	0.017	0.01	0.01	0.009	0.008	0.009	0.013	0.013
0.6	0.014	0.017	0.014	0.013	0.009	0.012	0.017	0.012	0.011
0.7	0.014	0.019	0.013	0.017	0.009	0.011	0.012	0.014	0.013
0.8	0.011	0.011	0.011	0.016	0.008	0.014	0.012	0.009	0.011
0.9	0.012	0.015	0.013	0.01	0.014	0.014	0.009	0.01	0.014
KW test	2.7849		0.50359		1.1418		0.23887		
df	1		1		1		2		
$p-value$	0.09516		0.4779		0.2853		0.8874		

Contents of the above table follow the same pattern as in the case of Set 1. The following are the results of ANOVA when applied for the cases $N = 20,30,40 \text{ and } 50$.

The p values indicate the error rates are do not show statistically significant differences with respect to percentile orders corresponding to minimum error rate. Hence, proceeding as in the case of Set 1, we recommend the values 0.75, 0.65, 0.65 and 0.6 for the cases $N = 20,30,40 \text{ and } 50$.

Table III gives results of the experimental study pertaining to Set 3. As in the previous two cases, the minimum error rates were obtained for two or three values of levels of percentiles. However, one can notice a drastic change in the values of optimal percentiles orders. Unlike the previous cases, here the optimal orders are considerably low ranging from 0.3 to 0.5. The changed pattern may be attributed to the behavior of the data set. The specifications given in Set 3 creates data sets showing lesser separation between the objects coming from the three classes when compared to the previous data sets.

TABLE III. MINIMUM ERROR RATES FOR DATA SET WITH SEPARATION INDEX 0.25

Separation Index 0.25									
ρ	$N = 20$		$N = 30$		$N = 40$		$N = 50$		
	0.3	**0.4**	**0.3**	**0.4**	**0.4**	**0.5**	**0.3**	**0.4**	**0.5**
-0.5	0.187	0.171	0.172	0.188	0.187	0.174	0.143	0.146	0.153
-0.4	0.213	0.212	0.181	0.179	0.159	0.15	0.17	0.168	0.174
-0.3	0.166	0.174	0.184	0.186	0.154	0.162	0.151	0.151	0.148
-0.2	0.203	0.169	0.196	0.176	0.142	0.178	0.156	0.169	0.163
-0.1	0.173	0.181	0.18	0.177	0.154	0.151	0.157	0.161	0.182
0	0.191	0.165	0.182	0.175	0.166	0.153	0.146	0.143	0.148
0.1	0.202	0.205	0.144	0.152	0.186	0.179	0.163	0.156	0.151
0.2	0.176	0.182	0.152	0.157	0.152	0.166	0.148	0.142	0.152
0.3	0.177	0.183	0.163	0.164	0.164	0.154	0.168	0.15	0.155
0.4	0.236	0.266	0.16	0.165	0.15	0.146	0.152	0.162	0.16
0.5	0.207	0.227	0.161	0.158	0.149	0.144	0.175	0.163	0.185
0.6	0.187	0.191	0.193	0.188	0.164	0.174	0.15	0.157	0.146
0.7	0.202	0.177	0.166	0.144	0.153	0.158	0.156	0.154	0.165
0.8	0.212	0.212	0.197	0.17	0.149	0.16	0.16	0.158	0.154
0.9	0.18	0.174	0.176	0.161	0.18	0.177	0.15	0.154	0.157
KW test	0.41416		0.72382		0.084489		0.42381		

df	1	1	1	2
$p-value$	0.5199	0.3949	0.7713	0.809

The results of ANOVA corresponding to the four choices of N, namely 20,30,40 and 50 summarized in Table III show that there is no significant difference between error rates irrespective the order of optimal percentile orders identified as the best.

Hence, we recommend their averages namely, 0.35,0.35, 0.45 and 0.4 as orders of percentiles to be used for the case $N = 20,30,40 \text{ and } 50$ when a data set similar to Set 3 is being studied. Results related the optimal percentile orders and the findings of ANOVA when test data sets have been simulated as per the settings of Set 4 is presented in Table IV.

TABLE IV. MINIMUM ERROR RATES FOR DATA SET WITH SEPARATION INDEX 0.20

	Separation Index 0.20										
ρ	$N=20$			$N=30$		$N=40$			$N=50$		
	0.4	**0.5**	**0.6**	**0.4**	**0.5**	**0.5**	**0.6**	**0.7**	**0.4**	**0.5**	**0.6**
-0.5	0.513	0.49	0.486	0.506	0.511	0.457	0.471	0.469	0.484	0.491	0.481
-0.4	0.503	0.504	0.481	0.52	0.508	0.515	0.513	0.507	0.475	0.467	0.472
-0.3	0.536	0.512	0.518	0.522	0.522	0.479	0.472	0.466	0.476	0.476	0.47
-0.2	0.522	0.492	0.506	0.472	0.461	0.479	0.474	0.483	0.473	0.488	0.469
-0.1	0.488	0.501	0.475	0.466	0.455	0.462	0.454	0.453	0.451	0.454	0.446
0	0.499	0.497	0.492	0.462	0.44	0.465	0.456	0.474	0.448	0.463	0.458
0.1	0.500	0.506	0.503	0.477	0.493	0.465	0.469	0.487	0.49	0.495	0.474
0.2	0.480	0.493	0.515	0.476	0.467	0.461	0.447	0.462	0.478	0.488	0.482
0.3	0.494	0.469	0.469	0.47	0.462	0.489	0.476	0.509	0.468	0.452	0.471
0.4	0.448	0.475	0.482	0.48	0.483	0.458	0.475	0.457	0.447	0.452	0.448
0.5	0.452	0.474	0.457	0.473	0.458	0.468	0.471	0.478	0.501	0.49	0.484
0.6	0.467	0.472	0.467	0.494	0.499	0.451	0.474	0.452	0.452	0.467	0.468
0.7	0.515	0.477	0.494	0.458	0.492	0.491	0.483	0.482	0.463	0.469	0.486
0.8	0.501	0.473	0.479	0.502	0.486	0.461	0.466	0.482	0.483	0.484	0.473
0.9	0.505	0.492	0.476	0.508	0.489	0.456	0.466	0.456	0.466	0.45	0.464
KW test	1.8045			0.22773		0.60915			0.3722		
df	2			1		2			2		
$p-value$	0.4057			0.6332		0.7374			0.8302		

Contents of the above table lead to conclusions which are in line with those corresponding to the previous three sets. However, the numbers of possible levels leading to minimum error rates happen to be 3, 2, 3 and 3 whereas in the previous sets those were 2, 2, 2 and 3. Even though, the numbers of possible levels have increased, the ANOVA tables show the difference in the error rates do not differ significantly as one can see from the p values. Following the approach used in those cases, the choices 0.5, 0.45, 0.6 and 0.5 are recommended for the cases $N = 20,30,40 \text{ and } 50$ respectively.

C. Natural Data Based Study

In the experimental studies explained above conclusions on the choice of orders of percentiles have been drawn using data sets simulated from multivariate normal distribution. However in the practical point of view, similar studies have to be carried out for natural data sets in order to reach credible and practically useful conclusions regarding the efficiency of the proposed algorithms. Towards this four popular data sets used in machine learning studies have been considered. Brief descriptions on the data sets are given below.

iris data : This data set gives measurements on four characteristics of 150 flowers from three species of namely Setosa, Versicolor and Virginca. Under each one of these three classes 50 instances are considered.

urine data: The *urine data set* is available in Andrews and Herzberg (1985) The data is related to 79 urine specimens considered for analysis carried out in order to determine if certain physical characteristics of the urine might be related to the formation of calcium oxalate crystals. The data set consists of 2 classes of 45 instances having the presence of calcium oxalate crystals and 34 instances constitutes the class that refers to the absence of calcium oxalate crystals.

Salmon Fish data : The *salmon fish* data set is available in the R-package '*rrcov'* under the default settings. The data set consists of two measurements namely, the growth rings on the scale of Alaskan and Canadian salmon as well as the gender of the fishes. The data set consist of two classes of 50 Alaskan-born and 50 Canadian-born salmon.

Breast Cancer data: The *Wisconsin Breast Cancer data set* is obtained from UCI machine learning repository. The data set contains information about 569 breast FNA cases including 30 descriptive attributes and one class variable (malignant and benign). The data set consists of 2 classes of 357 cases of benign breast changes and 212 cases of malignant breast cancer.

Table V gives the error rates corresponding to different levels of percentiles for the four natural data sets considered in this study. It may be noted that the error rate increases as the order of percentile increases in all the four data sets. Even though, we succeeded in recommending an optimal order for the percentile to be used in the case of simulated data sets on the basis of separation index, the task of identifying the optimal choice remains a difficult one in the case of bench mark data sets. The optimal orders of percentiles could not be associated with the separation index in order to minimize the error rate. This is contrary to the conclusions drawn in the case of simulated data sets. However, one can notice that irrespective of the value of separation index, the error rates continue to increase of the level of percentiles increase. Hence, involving high percentage of distances in the classification process leads to increased error rate. Hence, the proposed algorithm is recommended for use with a lower order percentile, say around 0.3.

TABLE V. MEAN ERROR RATES FOR NATURAL DATA SETS

Data Set	Levels of Percentiles									Separation Index
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	
Iris	0.027	0.033	0.013	0.033	0.033	0.033	0.053	0.080	0.113	0.58
Urine	0.155	0.207	0.272	0.286	0.286	0.299	0.299	0.338	0.399	0.10
Salmon	0.06	0.07	0.07	0.08	0.08	0.08	0.11	0.13	0.15	0.32
Breast Cancer	0.028	0.037	0.042	0.054	0.063	0.076	0.090	0.127	0.207	0.40

4. CONCLUSION

In this work, a new neighborhood based classification method which makes of the distributional characteristics, in particular percentiles, of the distances between units in the training set has been developed. The proposed algorithm addresses certain limitations of the popular neighborhood based method, namely, k-nn classifier. The new algorithm does not expect the user to specify the value of k, namely the number of training set objects which are closure to the test object. It is fundamentally different from the lazy learner k-nn, since, the classifier makes use of the distribution of distance based only on the training set objects. The performance of the proposed algorithm depends on the choice of the percentiles based on the distribution of the distances between training set objects. Detailed experimental studies have been carried out to identify the choice of percentiles using a variety of simulated and bench mark data sets. It was found that the choice of percentiles depends on the value of the separation index. For training data sets, having smaller separation index, the usage of a smaller percentile is recommended. Similarly for data sets with larger separation index the error rate becomes smaller if higher order percentiles are used. The following are the findings of the numerical studies.

- For data sets with separation index 0.70 (simulation corresponding to Set 1), the optimal orders of percentiles were found to be 0.75, 0.65, 0.65 and 0.70 for the cases, $N = 20,30,40 \text{ and } 50$.
- The optimal orders of percentiles were 0.75, 0.65, 0.65 and 0.6 for the cases $N = 20,30,40 \text{ and } 50$ when the separation index is 0.50. Such data sets arise when simulation is carried out under the parametric setting given in Set 2.

- In the case of separation index 0.25 (based on simulation performed under parametric setting given in Set 3), the optimal orders of percentiles were found to be 0.35, 0.35, 0.45 and 0.4 for the cases $N = 20,30,40 \text{ and } 50$ respectively.
- The choices 0.5, 0.45, 0.6 and 0.5 are recommended for the orders of percentiles for the cases $N = 20,30,40 \text{ and } 50$ respectively when the separation index is 0.20. Such data sets can be simulated using the specification mentioned in Set 4.

For all the four bench mark data sets, a lower order percentile around 0.30 gives smaller error rate.

REFERENCES

[1] B. Ahmad, M. Ali, and G. Altarawneh, "Solving the problem of the k parameter in the k-nn classifier using an ensemble learning approach," International Journal of Computer Science and Information Security, vol.12(8), pp. 33-39,2014.

[2] M. Akhil Jabbar, B.L. Deekshatulu and P. Chandra, "Classification of heart disease using k-nearest neighbor and genetic algorithm," Journal of Procedia Technology, vol.10, pp. 85-94, 2013.

[3] G. Ashish and P. Bijnan, "Protein secondary structure based on distance based classifier," International Journal of Approximate Reasoning, vol. 47, pp.37-44, 2008.

[4] P. Bhubaneswar and A. Brintha Therese, "Detection of cancer in lung with k-nn classification using genetic algorithm," Journal of Procedia Material Science, vol.10, pp. 433-440, 2015.

[5] T. Bruno, M. Sasa and D. Dzenana, "KNN with TF-IDF based framework for text categorization," Procedia Engineering, vol. 69, pp. 1356-1364, 2014.

[6] Y. Chang and Y. Chen, "Improving nearest neighbor classification with cam weighted distance," Journal of Pattern Recognition, vol. 39, pp. 635-645, 2006.

[7] T.M. Cover & P.E.Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13(1), pp. 21-27, 1967.

[8] G.G. Enas and S.C. Choi, "Choice of the smoothing parameter and efficiency of k-nearest neighbor classification," Journal of Computer & Maths with Applications, vol. 12A(2), pp. 235-244, 1986.

[9] E. Fix and J. Hodges, "Discriminatory analysis, nonparametric discrimination: consistent properties," Tech. Rep. 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[10] A.K. Ghosh, "On optimum choice of k in nearest neighbor classification," Journal of Statistics and Data Analysis, vol.50, pp.3113-3123, 2006.

[11] M. Govindarajan and R.M. Chandrasekaran, "Evaluation of k-nearest neighbor classifier based on direct marketing," Journal of Expert Systems with Applications, vol.37, pp. 253-258, 2010.

[12] H. Liu, S. Zhang, J. Zhao and Y. Mo, "New Classification algorithm using mutual nearest neighbors," 9th International Conference on Grid and Cloud Computing, IEEE Computer Society, pp. 52-57, 2010.

[13] H. Parvin, H. Alizadeh and M. Behrouz, "MKNN: Modified k-nearest neighbor classification," Proceedings of the World Congress on Engineering and Computer Science, October 22 - 24, San Francisco, USA, 2008.

[14] Yihua Liao and V. Rao, "Use of K-nearest neighbor classifier for intrusion detection,". Journal of Computer and Security, vol. 21(5), pp. 439-448, 2002.

[15] W. Yingquan, I. Krassimir and V. Govindaraju, "Improved k-nearest neighbor classification," Journal of Pattern Recognition, vol. 35, pp. 2311-2318, 2002.

[16] D. Zhenyun, X. Zhu D. Cheng and S. Zhang, "Efficient kNN classification algorithm for big data," Journal of Neuro Computing. Vol.195, pp.143-148, 2016.

[17] Z. Zhongheng "Introduction to machine Learning: k-nearest neighbors," Ann Transl Med,Vol. 4(11), pp. 01-07.