Dhaka Univ. J. Sci. 61(2): 179-183, 2013 (July)

A New Algorithm for Classification Based on *K* Nearest Neighbor and Decision Tree

Anamul H. Sajib[1] and Jafar A. Khan[*]

Department of Statistics, Biostatistics and Informatics, Dhaka University, Dhaka-1000, Bangladesh

(Received: 01 October 2012; Accepted: 01 April 2013)

Abstract

In a classification problem with binary outcome attribute, if the input attributes are both continuous and categorical, the *K* Nearest Neighbor (KNN) technique cannot be used. On the other hand, the Decision Tree (DT) technique handles the continuous attributes by discretization which leads to loss of information.

To overcome the limitations of the KNN and DT techniques, we propose a new technique in this study which is called *K* Nearest Neighbor Decision Tree (KNNDT). The proposed technique uses a combination of KNN and DT to classify the test instances. KNNDT first uses the KNN technique to select homogeneous groups of training instances by using the continuous attributes and then builds local decision trees on these homogeneous groups by using the categorical attributes.

An extensive simulation study was conducted to compare the performances of KNNDT and DT. In general, the proposed KNNDT gives better results compared to DT.

***Keywords*:** KNN, DT, Classification problem

I. Introduction

The classification problem with binary output attribute is considered when input attributes are both continuous and categorical. If all the input attributes are continuous, the *K* Nearest Neighbor (KNN) technique[1] uses Euclidean distance, Mahalanobis distance etc. to select the nearest neighbors of the test instance in the training data. If all the input attributes are categorical, the KNN uses complex similarity measurements like Hamming distance, Jaccard index, Tanimoto coefficients etc. However, when input attributes are both continuous and categorical, KNN is not suitable. On the other hand, the Decision Tree (DT) technique[2] is able to deal with both continuous and categorical attributes for classification. However, this technique handles the continuous attributes by discretization. This approach has two limitations. First, this treats a continuous attribute as a discrete one which leads to the loss of information. Second, it is always difficult to decide how many categories to make when we are performing the discretization. To overcome the limitations of the KNN and DT, a new technique called ***K*** Nearest Neighbor Decision Tree (KNNDT) is proposed. The modification uses a combination of KNN and DT techniques to classify the test instances. The rest of the article is organized as follows. In Section 2 the proposed KNNDT is introduced. In Section 3 the KNNDT is illustrated with an example. In Section 4, an extensive simulation study is conducted to compare the performances of KNNDT with the existing DT technique. The discussion and conclusion are presented in Section 5 and 6 respectively.

II. The Proposed KNNDT Technique

In order to avoid the discretization of the continuous attributes in DT and the calculation of complex similarity measurements for the categorical attributes in KNNs, a new technique called KNNDT is proposed. The proposed technique uses a combination of the KNN and DT techniques to classify the test instances. The motivation is to improve the performance of DT by synchronously using the following:

(a) Combine the power of KNN and DT techniques over different attributes.

(b) Deploy a local decision tree on the *K* nearest neighbors of a test instance.

(c) Learn the best value of *K* for each test instance in training time.

(d) Control a large tree by deploying local decision trees.

The KNNDT Algorithm

To classify each of the test instances, the proposed technique proceeds in two steps. In the first step, the KNN technique is applied over the continuous attributes to select ***K*** training instances that are closest to the test instance. This is reasonable, since numerically close instances are supposed to have the same characteristics. In this step, the categorical attributes are not used. In the second step, instead of taking a simple voting scheme as in KNN, the proposed technique builds a local decision tree based on the ***K*** selected training instances by using only the categorical attributes. The particular test instance is then classified by using this local decision tree. The proposed algorithm may be summarized as follows: (i) use KNN over the continuous attributes to select ***K*** nearest neighbors of the test instance, (ii) use the set of ***K*** training instances obtained from 1 to build a local model using DT over the categorical attributes, (iii) use the model built in 2 to classify the test instance.

III. Example

To illustrate the proposed technique, let us consider a data set of size $n = 30$ in which there are 3 continuous and 3 categorical input attributes while the same data set contains one binary output attribute. Continuous attributes were generated using standard normal distribution. Three categorical attributes were generated from Bernoulli distribution with $\theta = .40, .45$ and $.5$ respectively. In this

[*]Author for correspondence, *emai l:* jkhan66@gmail.com

case Logistic model was used to create binary categorical output. Continuous attributes were denoted by X_1, X_2 and X_3 while for categorical attributes U_1, U_2 and U_3 are used. Note that categorical binary output attribute was denoted by Y. Column 2, 3 and 4 of Table 1 show the generated values of X_1, X_2 and X_3, respectively, rounded to two decimal places. Column 5, 6 and 7 of the same Table show the generated values of categorical attributes while column 8 shows the values of binary categorical output attribute.

Table 1. Generated data from Standard normal, Bernoulli and Logistic distributions

Serial no.	X_1	X_2	X_3	U_1	U_2	U_3	Y
01	0.33	-0.31	0.41	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
29	-1.17	-0.15	-0.67	0	1	0	0
30	-2.33	-1.32	-0.74	0	0	0	0

Suppose we have to classify the following test instance

31 **-0.05** **0.82** **-0.53** 0 1 1 ?

Now, the Euclidean distances between the training instances and the test instance can be calculated based on the continuous attributes. Then we select *K*=10 training cases that are closest to the test case based on these distances. Table 2 shows the selected nearest neighbors with only the categorical input attributes and the binary output attribute. The local decision tree can be constructed based on the data given in Table 2.

Table 2. Selected *K* nearest neighbors

Serial no.	U_1	U_2	U_3	Y
23	0	1	1	1
⋮	⋮	⋮	⋮	⋮
13	0	0	0	0

Decision Tree Construction

Several algorithms have been developed to create a decision tree. Some of the more popular ones are ID3[3], C4.5[4], CART[5] and CHAID[6]. While they all differ in some way, they all share the common idea of building the tree using a technique based on information theory. In this study Quinlan ID3 algorithm was used to create decision tree.

From tree to rules

A decision list is a set of $if-then$ statements. It is searched sequentially for an appropriate $if-then$ statement to be used as a rule. Now, the rules we established from the decision tree which can directly be used for classification of new instance.

Table 3. Decision rules for decision tree

$if(condition)$	$then(decision)$
1. $category\ of\ U_2\ is$ "0" and $category\ of\ U_1\ is$ "0"	$instance\ gets\ class$ 0
2. $category\ of\ U_2\ is$ "0" and $category\ of\ U_1\ is$ "1"	$instance\ gets\ class$ 1
3. ***category of U_2 is 1***	*instance gets class 1*

Now we can classify new instance by using above decision rules. For this new instance, it is clear that category of attribute U_2 is "1" so on the basis of rule 3 we can classify this new instance as class 1.

IV. Simulation

An extensive simulation study was conducted to compare the performances of the existing DT technique with the performances of the proposed KNNDT technique. The performances of these two techniques are determined by misclassification rate and hit curve. The standard normal and Bernoulli distributions were used to generate the input attributes, while logistic model was used to generate the output attribute. For continuous and categorical attributes the standard normal and Bernoulli (θ) distributions were used respectively. After generating the continuous and categorical attributes, the binary outcomes attribute was generated by using these generated attributes. For simplicity independence of continuous and categorical attributes was assumed and consequently no interaction between them (no

interaction effect) was considered. The logistic model is in the form of $P(Y = 1)$, where $P(Y = 1) = \frac{\exp(\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4U_1+\beta_5U_2+\beta_6U_6)}{1+\exp(\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4U_1+\beta_5U_2+\beta_6U_6)}$ is used to create the binary outcomes attribute Y, a pseudorandom realization, U of a uniform $(0,1)$ attribute was simulated and compared with P. We set $Y = 1$, if $P > U$, otherwise we set $Y = 0$. The above idea for generating data is suggested by Veeranun Pongsapukdee[7]. $\theta = .5,\ .3$ and $.3$ were considered to generate 3 categorical attributes from Bernoulli distribution respectively. Now to choose the coefficients of logistic model i.e. $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6)$, emphasis was given to the value of P. If the value of P is large enough then almost all the values of Y will be "1". On the other hand if the value of P is too small then almost all the values of Y will be "0". Considering these issues, β was chosen in such way that the value of P is moderate in size. Two different set of values of β was selected such that the average value of P becomes approximately 0.30 and 0.45 i.e. the datasets contain approximately 30% and 45% $Y =$ "1" respectively. For each β, 3 different sample sizes: $n = 50, n = 100$ and $n = 150$ was considered. Thus, there are 6 different situations based on the values of β and n. For each situation, 1000 (one thousand) datasets were generated.

Table 4. Parameters used in logistic model for 2 different situations

Parameters	Mean$(P) \approx .30$	Mean$(P) \approx .45$
β_0	-2.00	-1.80
β_1	0.60	1.00
β_2	0.10	1.00
β_3	0.81	1.00
β_4	0.18	1.00
β_5	0.50	1.00
β_6	0.71	1.00

For each of the two techniques existing DT and the proposed KNNDT, the number of misclassifications were considered for each of the dataset. For both of the techniques how many times (out of one thousand dataset) misclassification occurred at numbers $T = 0$, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 where T stands for number of misclassified test cases out of 10 were counted. Also the average misclassifications rate was calculated for both of the techniques over the datasets from which those misclassifications were calculated. Table 5 shows the results obtained for $\beta = (-2,\ 0.6,\ 0.10,\ 0.81,\ 0.18,\ 0.5,\ 0.71)$ and $n = 100$. The first column shows the 11 possible numbers considered from 0 to 10 with an increment of 1 at which misclassification occur. The second column (count1) shows how many times misclassification occur corresponding that number for the proposed KNNDT technique while the third column (count 2) of the same table shows how many times misclassification occur corresponding that number for existing DT technique over 1000 (one thousand datasets). First column of Table 6 shows the different techniques while second column of the same Table shows the average misclassification rate over one thousand datasets corresponding technique.

Table 5. Frequency of misclassifications of proposed KNNDT and existing DT for $\boldsymbol{\beta = (-2,\ .6,\ .10,\ .81,\ .18,\ .5,\ .71)}$ and $\boldsymbol{n = 100}$ over 1000 datasets at $\boldsymbol{T}$

Number(T)	Count1	Count2
0	**43**	30
1	**162**	113
2	**269**	228
3	247	**279**
4	165	**198**
5	76	**97**
6	31	**44**
7	6	**10**
8	1	**1**
9	0	**0**
10	0	**0**
Total	1000	**1000**

Table 6. Average misclassification rate of proposed KNNDT and existing DT for $\boldsymbol{\beta = (-2,\ .6,\ .10,\ .81,\ .18,\ .5,\ .71)}$ and $\boldsymbol{n = 100}$

Method	Average misclassification rate (%)
KNNDT	27
DT	30

From Table 5 it can be seen that, in case of the proposed KNNDT technique, frequency of misclassification at $T = 0$ (100% accuracy) is 43 times, while in case of DT technique it is 30 times which less than the previous one. Frequencies of misclassifications at $T = 1, 2$ in the proposed KNNDT are 162, 269, while in DT these are 113, 228 respectively which means that for the proposed KNNDT less number of errors occurs in more datasets compared to the existing DT technique.

On the other hand in case of the proposed KNNDT technique, frequencies of misclassifications at $T = 3, 4, 5, 6$ and 7 are 247, 165, 76, 31 and 6, while in case of DT technique these are 279, 198, 97,44 and 10 times respectively which means that for the proposed KNNDT large number of errors occurs in less datasets compared to the existing DT technique. It can also be noted that, frequencies of misclassifications are 0 (zero) for both of techniques at $T = 9$ and 10 while 1 misclassification occur for both of techniques at $T = 8$ which means that in case of very high misclassification (10 out of 10, 9 out of 10) both methods show similar patterns. Table 6 shows the average misclassification rate of proposed KNNDT (27%) is less than DT (30%) which also suggests that KNNDT outperforms than DT. Figure 1 plots the two frequency distributions corresponding to the proposed KNNDT (solid line) and existing DT (dashed line) for $\beta = (-2, .6, .10, .81, .18, .5, .71)$and $n = 100$. The horizontal axis shows the number of misclassified test cases out of 10 which is denoted by T while the vertical axis shows the frequency of each value of T, i.e., how many times out of 1000 a particular value of T occurs. This plot shows that the proposed KNNDT technique has more frequencies for small values of T (number of misclassified test cases out of 10), which means that for this method less number of error occur in more datasets compared to the existing DT technique. On the other hand, proposed KNNDT technique has less frequencies for large values of T (number of misclassified test cases out of 10), which means that for proposed KNNDT large number of errors occurs in small datasets compared to the existing DT. Figure 2 plots the two hit curves for $\beta = (-2, .6, .10, .81, .18, .5, .71)$ and $n = 1000$ corresponding to the proposed KNNDT (solid line) and existing DT (dashed line). The horizontal axis shows the number of test cases while the vertical axis shows the number of actual hits. This plot shows that the number of actual hits for KNNDT is always greater than that of the DT technique. The difference between number of actual hits of the two techniques increases when the size of the test cases increases. The results for other situations are not presented here since they lead to similar plots. For both of the techniques, average missclassification rate increases as the percentage of both of the categories become close enough. We also see that the number of actual hits in KNNDT technique is greater than the number of actual hits in DT technique which means that the KNNDT gives better performance compared to DT technique.

V. Discussion

For all six situations based on β and n, proposed KNNDT has smaller misclassification rate than existing DT. The reasons are as follows. The KNNDT technique treats continuous attributes as a continuous one and builds a decision tree based on relatively homogeneous training instances (K nearest neighbors) to make a decision. On the other hand, the DT technique treats continous attribute as a discrete one as a result full information is not utilized here. Moreover, DT technique built a decision tree based on original training instances. As a result it often creates large tree for which overfitting may arise. As the KNNDT uses full information for continuous attributes and builds decision tree based on homogeneous training instances, so it gives smaller misclassification rate compared to DT. For this same reasons the number of actual hits in KNNDT is greater compared to the number of actual hits in DT technique. The misclassifications rates of both techniques for $\beta = (-1.8, 1, 1, 1, 1, 1, 1)$ are greater than those for $\beta = (-2, .6, .10, .81, .18, .5, .71)$ for all sample sizes. The reason is as follows. When $\beta = (-1.8, 1, 1, 1, 1, 1, 1)$ i.e. mean(P) is approximately $.45$ which means that the datasets contain approximately 45% $Y =$ "1". For this situation percentage of both of the categories of output attribute becomes close enough compared to ($\beta = (-2, .6, .10, .81, .18, .5, .71)$). As a result, entropy is high.

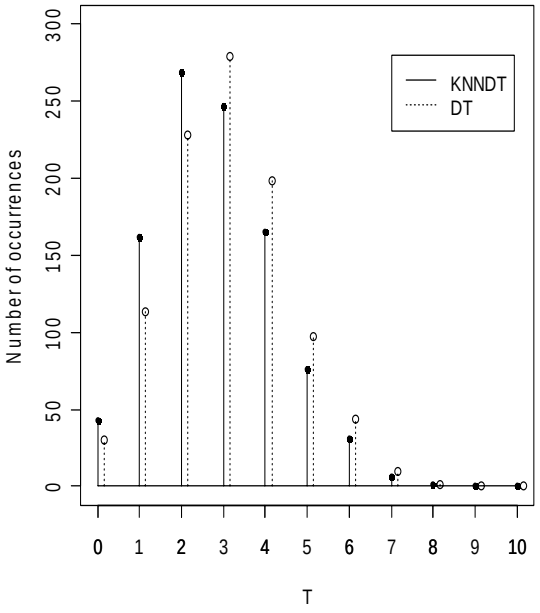

Fig. 1. Number of misclassification of KNNDT and DT for $\beta = (-2, .6, .10, .81, .18, .5, .71)$ $n = 100$

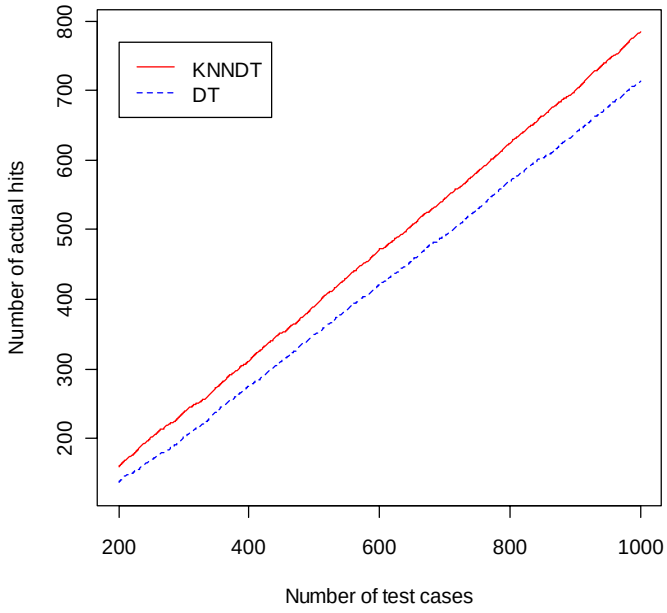

Fig. 2. Hit curves for $\beta = (-2,\ 0.6,\ 0.10,\ 0.81,\ 0.18,\ 0.5,\ 0.71)$ and $n = 1000$.

VI. Conclusion

In a classification problem with binary outcome attribute, if all the input attributes are continuous, then the K Nearest Neighbor (KNN) technique[1] uses these attributes to find the distances of each training case from the test case. However, the KNN technique does not handle categorical attributes properly, because it requires complex similarity measurements (Hamming distance, Jaccard index, Tanimoto coefficients etc.) for these categorical attributes. Therefore, KNN is not suitable classification technique when input attributes are both continuous and categorical. On the other hand Decision Tree (DT) technique [2] is able to process both continuous and categorical attributes for classification. However, this technique handles the continuous attributes by discretization. This approach has two limitations. First, this treats a continuous attribute as a discrete one. As a result we do not get full information. Second, it is always difficult to decide how many categories to make when we are performing the discretization. To overcome the limitations of the KNN and DT techniques (to avoid discretization in DT and calculate complex similarity measurements for categorical attributes in KNN techniques), a new technique called *K* Nearest Neighbor Decision Tree (KNNDT) technique is proposed. In the proposed technique, one does not need to discretize the continuous attributes or calculate complex similarity measurements for categorical attributes. The modification uses a combination of KNN and DT techniques to classify new instances. An extensive simulation study was conducted to compare the performance of the existing DT technique with the performance of the proposed KNNDT technique. In general, proposed KNNDT technique gives better result compared to the existing DT technique.

References

1. Fix, E. and J. L. Hodges, 1951. Discriminatory analysis, nonparametric discrimination. *USAF School of Aviation Medicine, Randolph Field, Texas*, Project 21-49-004, Report 4, Contract AF41(128)-31.
2. Morgan, J. A. and J. N. Sonquist, 1963. Problems in the analysis of survey data and a proposal. *Journal of the American Statistical Association*, **58**, 415- 434.
3. Quinlan, J. R., 1986. Induction of decision trees. *Journal of Machine Learning*, **1**, 81-106.
4. Quinlan, J. R., 1993. Simplifying decision tree. *International Journal of Man Machine Studies*, **51**,497-491.
5. Breiman, L., J. H., Friedman, R.A., Olshen, and C. J., Stone, 1984. Classification and Regression trees, Wadsworth International Group.
6. Kass, G.V., 1980. An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Journal of Applied Statistics*, **29**, 119-127.
7. Pongsapukdee, V., 2002. Analysis of Category Comparisons of Binary Response Data with the Combination of Continuous and Categorical Variables. *Silpakorn University International Journal*, **2(2).**