Hybrid Classification Algorithm for Knowledge Acquisition of Biomedical Data

Shovon K. Pramanik, Subrata Pramanik, Bimal K. Pramanik,
M. K. Islam Molla and Md. Ekramul Hamid

Dept. of Computer Science and Engineering, University of Rajshahi, Bangladeh
ekram_hamid@yahoo.com

Abstract

This paper presents a new hybrid classifier that combines the k-Nearest Neighbor (k-NN) distance based algorithm with the classification tree paradigm based on the ID3 algorithm. The k-NN algorithm is used as a preprocessing algorithm in order to obtain a modified training database for the posterior learning of the classification tree structure. Then the incorrectly classified instances are duplicated with the previous data set and finally ID3 is applied to complete the classification procedure of biomedical data. In this approach a boosting technique is incorporated in such way that the incorrectly classified instances in the training set are identified using the k –NN algorithm. The performance of the proposed method is compared with the related algorithms. Experimental results show that the newly proposed approach performs better than the other existing techniques.

Keywords: *Cross validation, decision tree, ID3 algorithm, k-NN algorithm*

1. Introduction

The amount of data kept in computer files and databases is growing at a phenomenal rate. At the same time, the users of these data are expecting more sophisticated information from them. Simple structured or query languages are not adequate to support these increased demands for information. Classification and prediction has been used by human beings from many years ago. At the early times men used hand calculation methods, perception, and knowledge for classification and prediction. This was done on limited amount of data. But with the growing need of time, size of data increases rapidly. For example, statistical bureau, stock exchange, banking sector, drug industries, etc have been experiencing millions of data every day. So, it is not possible for human beings to classify or make prediction using age old hand calculation methods for such huge amount of data. For this, usage of computer, various algorithms, and programming strategies has become a must.

There are various database management systems for manipulation of data, but extraction of information or knowledge from data is more complex than mere data manipulation. The objective of our research is to gain knowledge from the trained dataset and apply the knowledge on the test dataset. Accuracy is then measured to identify classification efficiency.

Classifier Combination is an extended terminology used in the Machine Learning [16], more specifically in the Supervised Pattern Recognition area, to point out the supervised classification approaches in which several classifiers are brought to contribute to the same task of recognition [14]. In this research, we have used hybrid approaches, in which the foundations of two or more different classification systems are implemented together in one classifier [15]. In the hybrid approach lays the concept of reductionism, where complex problems are solved through stepwise decomposition [2].

In this paper, we present a new hybrid Classifier based on two families of well known classification methods; the first one is a classification tree paradigm [12] and the second one is the distance based Classifier [13] which is combined with the former in the classification process. The k-NN algorithm is used as preprocessing algorithm in order to obtain a modified training database for the posterior learning of the classification tree structure. We have shown the results obtained by the new approach and a compare with the results obtained by the ID3 classification tree induction algorithm [17].

2. Overview of the Work

In this paper there are two phases of data processing. In the first phase, we have applied ID3 decision tree algorithm to classify data. The block diagram shows the process:

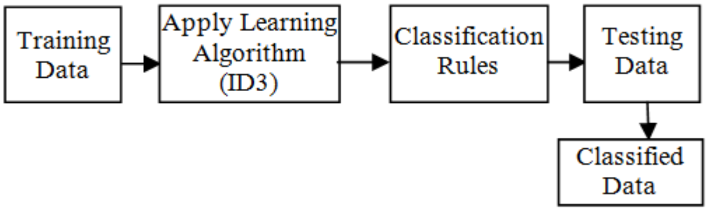


Figure 1. Classification Process using ID3 Algorithm

In the second phase, the k-NN algorithm is used to preprocess the train dataset so that the learning algorithm classifies the data better than the previous phase. The block diagram shows the process:

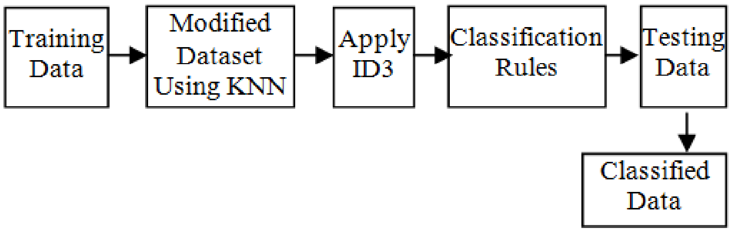


Figure 2. Classification Process using k-NN Boosting

3. Decision Tree Algorithm

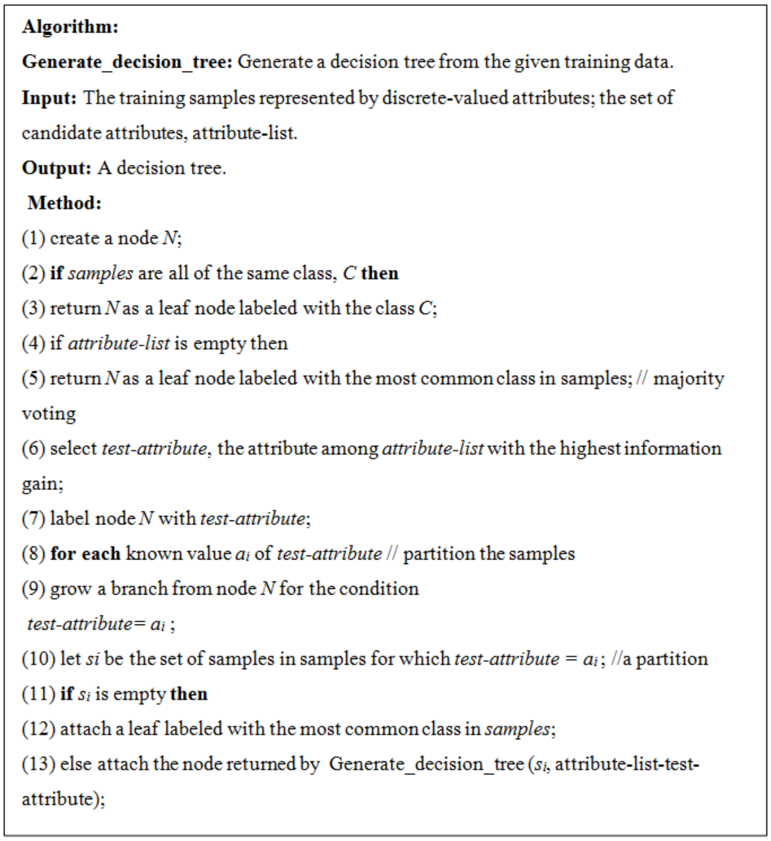
Decision tree learning algorithms have been successfully used in knowledge discovery. They use induction in order to provide an appropriate classification of objects in terms of their attributes, inferring decision tree rules. In their learning phase, explicit rules or interactions among relevant features are induced [1, 2, 4, 7]. Such a learning method differs from non-linear classifiers such as support vector machines or neural networks where the learning phase is to determine the parameters of the non-linear kernel functions. This advantage of rule-based classifiers is increasingly used in the analysis of bio-medical data, financial data in banking securities, stocks trading data, policy and claim data in insurance and the data obtained from other sources [3, 5, 6, 7].

A decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. Each node of the tree implements a decision rule that splits the examples into two or more partitions. New nodes are created to handle each of the partitions and a node is considered terminal or leaf node based on a stopping criteria. This approach to decision tree construction corresponds to a top-down greedy-algorithm that makes locally optimal decisions at each node. The sequence of the decisions made from the root node to the eventual labeling of a test input is easy to follow and also the approach can be extended to non-numeric domains where the attributes are categorical rather than numerical. This ability of decision tree induction method is used for data exploration by uncovering a mapping from independent to dependent variables used to analyze the data set [4, 9].

3.1. Decision Tree Induction Using ID3 Algorithm

In this research work, ID3 decision tree induction algorithm is used to extract rules from bio-medical data set. Figure-3 shows the ID3 algorithm for inducing a decision tree from training samples. The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide and conquer manner. The basic strategy is as follows:

1. The tree starts as a single node representing the training samples.
2. If the samples are all of the same class, then the node becomes a leaf and is labeled with that class.
3. Otherwise, the algorithm uses an entropy-based measure known as information gain as a heuristic for selecting the attribute that will best separate the samples into individual classes. This attribute becomes the “test” or “decision” attribute at the node.
4. A branch is created for each known value of the test attribute, and the samples are partitioned accordingly.
5. The algorithm uses the same process recursively to form a decision tree for the samples at each partition. Once an attribute has occurred at a node, it is not considered in any of the node’s descendents.
6. The recursive partitioning stops only when all samples for a given node belong to the same class or there are no more remaining attributes on which the samples may be further partitioned. In this case majority voting is employed. This involves converting given node into a leaf node and labeling it with the class in majority among the samples. Also the recursive partitioning stops when there are no samples for the branch test-attribute = a_i. In this case, a leaf is created with the majority class in the samples [2, 4, 10, 11].



```
Algorithm:
Generate_decision_tree: Generate a decision tree from the given training data.
Input: The training samples represented by discrete-valued attributes; the set of
candidate attributes, attribute-list.
Output: A decision tree.
Method:
(1) create a node N;
(2) if samples are all of the same class, C then
    (3) return N as a leaf node labeled with the class C;
(4) if attribute-list is empty then
    (5) return N as a leaf node labeled with the most common class in samples; // majority voting
(6) select test-attribute, the attribute among attribute-list with the highest information gain;
(7) label node N with test-attribute;
(8) for each known value $a_i$ of test-attribute // partition the samples
    (9) grow a branch from node N for the condition
     test-attribute= $a_i$ ;
    (10) let si be the set of samples in samples for which test-attribute = $a_i$ ; //a partition
    (11) if $s_i$ is empty then
        (12) attach a leaf labeled with the most common class in samples;
    (13) else attach the node returned by  Generate_decision_tree ($s_i$, attribute-list-test-attribute);
```

Figure 3. Basic Algorithm for Inducing a Decision Tree from Training Samples

3.2. Attribute Selection

Selection of the appropriate attributes at each node is the crucial part of this research work. The information gain measure is used to select the test attributes at each node in the tree. Gain measures how well a given attribute separates training examples into target classes. The attributes with the highest information gain is chosen as the test attribute for the current node. This attribute minimizes the information needed to classify the samples in the resulting partitions. Such an information-theoretic approach minimizes the expected number of tests needed to classify an object and guarantees that a tree is found [1, 4, 9]. In order to define gain, we first borrow an idea from information theory called entropy. Entropy measures the amount of information in an attribute.

The expected information needed to classify a given sample is given by

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m} P_i \log(P_i) \tag{1}$$

where s_i is the number of samples in class C_i and P_i is the probability that an arbitrary sample belongs to Class C_i and is estimated by s_i /*s*.

The entropy or expected information based on the partitioning into subsets by attribute *A*, is given by

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + s_{2j} + ... + s_{mj}}{s} I(s_{1j}, s_{2j} ... s_{mj}) \qquad (2)$$

where s_{ij} are the number of samples of class C_i in a subset. The term (s_{1j},...., s_{mj})/s acts as the weight of the *j*th subset and is the number of samples in the subset divided by total number of samples in *S*. For a given subset

$$I(s_{1j,} s_{2j}, ..., s_{mj}) = -\sum_{i=1}^{m} p_{ij} \log_2 (p_{ij}) \qquad (3)$$

where $p_{ij} = s_{ij}/|S_j|$ and is the probability that a sample in S_j belongs to class C_i. The encoding information that would be gained by branching on *A* is

$$Gain(A) = I(s_1, s_2, ..., s_m) - E(A) \qquad (4)$$

The *Gain(A)* is the expected reduction in entropy caused by knowing the value of attribute *A*. The attribute with highest information gain is chosen as the attribute for the given set *S*.

3.3. Tree Pruning

When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning method addresses this problem of over fitting the data [4, 10]. Prepruning approach is used for pruning a tree by halting its construction early by deciding not to further split or partition the subset of training samples at a given node. Upon halting, the node becomes a leaf. When constructing a tree, information gain measure is used to assess the goodness of a split. If partitioning the samples at a given node result in a split that falls below a prespecified threshold value, then further partitioning of the given subset is halted.

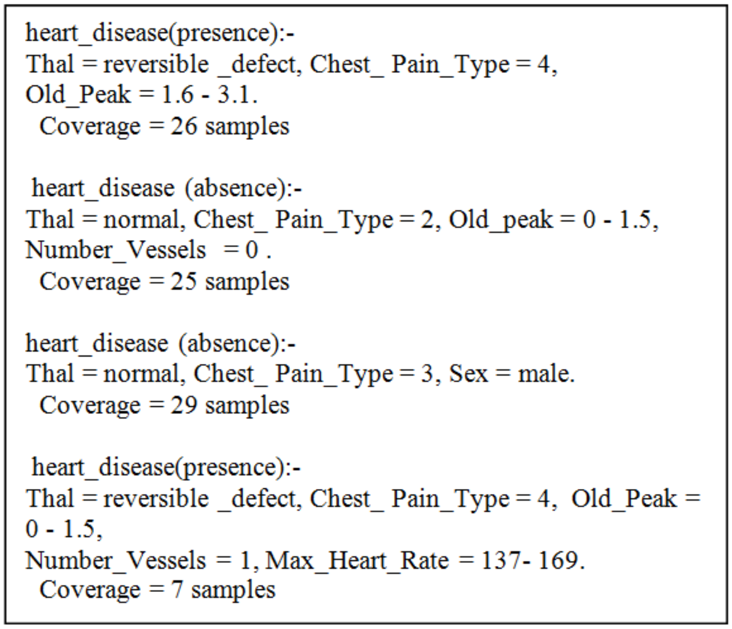
3.4. Review Stage

The heart disease data set of 270 patients is used in this experiment. The data set is obtained from UCI Machine Learning Repository [8]. Thirteen features are used to represent the heart disease data set as shown in Table 1. The features are: age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, the occurrence of exercise induced angina, ST depression induced by exercise relative to rest, the slope of peak exercise ST segment, number of major vessels colored by fluoroscopy, and thal. The classes for the data set are presence or absence of heart disease. Rules are extracted from this data set using decision tree induction method for heart disease diagnosis and expressing the patterns hidden within the data set.

Some of the rules extracted from data set are shown in Figure 4. The rules show the patterns extracted from heart disease data set for the presence or absence of heart disease. The first rule states that a patient will suffer from heart disease if thal shows reversible defect, chest pain type is asymptomatic and old_pick that is ST depression induced by exercise relative to rest is between 1.6 and 3.1. The second rule indicates that a patient will not suffer from heart disease

Table 1. Features of Dataset

1	Age:	29-40, 41-52, 53-64, 65-77
2	Sex:	male, female
3	Chest pain type:	1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic
4	Resting blood pressure:	94-120, 121-147, 148-174, 175-200
5	Serum cholesterol in mg/dl:	126-213, 214-301, 302-389, 390-477, 478-564
6	Fasting blood sugar > 120 mg/d:	yes, no
7	Resting electrocardiographic results:	0: normal, 1: having ST-T wave abnormality, 2: showing probable or definite left ventricular hypertrophy by Estes' criteria,
8	Maximum heart rate achieve:	71-103, 104-136, 137-169, 170-202
9	Exercise induced angina:	yes, no
10	ST depression induced by exercise relative to rest:	0-1.5, 1.6-3.1, 3.2-4.7, 4.8-6.2
11	The slope of the peak exercise ST segment:	1: up sloping, 2: flat, 3: down sloping
12	Number of major vessels colored By fluoroscopy:	0, 1, 2, 3
13	Thal:	normal, fixed defect, reversible defect



heart_disease(presence):-
Thal = reversible _defect, Chest_ Pain_Type = 4,
Old_Peak = 1.6 - 3.1.
Coverage = 26 samples

heart_disease (absence):-
Thal = normal, Chest_ Pain_Type = 2, Old_peak = 0 - 1.5,
Number_Vessels = 0 .
Coverage = 25 samples

heart_disease (absence):-
Thal = normal, Chest_ Pain_Type = 3, Sex = male.
Coverage = 29 samples

heart_disease(presence):-
Thal = reversible _defect, Chest_ Pain_Type = 4, Old_Peak = 0 - 1.5,
Number_Vessels = 1, Max_Heart_Rate = 137- 169.
Coverage = 7 samples

Figure 4. Extracted Rules from Heart Disease Dataset

if that shows no defect, chest pain is atypical angina type that is the chest pain is not typically brought on by excitation caused by an inadequate blood supply to the heart, old pick that is ST depression induced by exercise relative to rest is between 0 and 1.5 and the number of major vessels colored by fluoroscopy is zero. Similarly the third and fourth rules show the patterns for heart disease diagnosis. The rule in this format is easily understandable and can be used to interpret the patterns in bio-medical data set [8].

3.5. Boosting Using k-NN Algorithm

In boosting techniques, a distribution or set of weights over the training set is maintained. On each execution, the weights of incorrectly classified examples are increased so that the base learner is forced to focus on the hard examples in the training set [14, 16, 17]. Following the idea of focusing in the hard examples, we wanted to know if one algorithm could be used to boost a different one, in a simple way. We have chosen two well-known algorithms, k-NN and ID3, and our approach (in the following we will refer to it as k-NN-boosting) works as follows:

- Find the incorrectly classified instances in the training set using k-NN over the training set but the instance to be classified
- Duplicate the instances incorrectly classified in the previous step
- Apply ID3 to the augmented training set

Let us note that this approach is equivalent to duplicate the weight of incorrectly classified instances, according to k-NN. In this manner, the core of this new approach consists of initiating the training database adding the cases misclassified by the k-NN algorithm, and then learns the classification tree from the new database obtained. It has to be said that this approach increases the computational cost only in the model induction phase, while the classification costs are the same as in the original ID3 paradigm [20, 21].

4. Features of k-NN

The k-nearest-neighbor classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. Let X_i be an input sample with p features $(x_{i1}, x_{i2},\ldots, x_{ip})$, n be the total number of input samples $(i= 1,2,\ldots,n)$ and p the total number of features $(j= 1,2,\ldots,p)$. The Euclidean distance between sample X_i and X_l $(l= 1,2,\ldots,n)$ is defined as

$$d(X_i, X_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \ldots + (x_{ip} - x_{lp})^2} \tag{5}$$

The k-nearest-neighbor classification rul e is to assign to a test point the majority category label of its k nearest training points. In practice, k is usually chosen to be odd, so as to avoid ties. The k=1 rule is generally called the nearest-neighbor classification rule [14].

4.1. Decision Rule and Confusion Matrix for Classification

Classification typically involves partitioning samples into training and testing categories. Let x_i be a training sample and x be a test sample, and let ω be the true class of a training sample and $\acute{\omega}$ be the predicted class for a test sample $(\omega, \acute{\omega}=1, 2,\ldots, \Omega)$. Here, Ω is the total number of classes.

During the training process, we use only the true class ω of each training sample to train the classifier, while during testing we predict the class $\acute{\omega}$ of each test sample. It warrants noting that k-NN is a "supervised" classification method in that it uses the class labels of the training data. Unsupervised classification methods, or "clustering" methods, on the other hand, do not employ the class labels of the training data [17, 20, 21]. With 1-nearest neighbor rule, the predicted class of test sample X is set equal to the true class ω of its nearest neighbor, where m_i is a nearest neighbor to X if the distance

$$d(m_i, x) = \min_j \{d(m_j, x)\} \tag{6}$$

For k-nearest neighbors, the predicted class of test sample X is set equal to the most frequent true class among k nearest training samples. This forms the decision rule $D: X \rightarrow \acute{\omega}$.

The confusion matrix used for tabulating test sample class predictions during testing is denoted as C and has dimensions $\Omega \times \Omega$. During testing, if the predicted class of test sample X is correct (i.e. $\acute{\omega} = \omega$), then the diagonal element C$\omega\omega$ of the confusion matrix is incremented by 1. However, if the predicted class is incorrect (i.e., $\acute{\omega} \neq \omega$), then the off-diagonal element C$\omega\acute{\omega}$ is incremented by 1. Once all the test samples have been classified, the classification accuracy is based on the ratio of the sum of diagonal elements C$\omega\omega$ to the sum of off-diagonal elements C$\omega\acute{\omega}$ in C given in the form

$$Acc = \frac{\sum_{\omega}^{\Omega} C\omega\omega}{n_{total}} \tag{7}$$

where, n_{total} is the sum of off-diagonal elements of C.

4.2. Performance Assessment with Cross-Validation

A basic rule in classification analysis is that class predictions are not made for data samples that are used for training or learning. If class predictions are made for samples used in training or learning, the accuracy will be artificially biased upward. Instead, class predictions are made for samples that are kept out of training process.

The performance of most classifiers is typically evaluated through cross-validation, which involves the determination of classification accuracy for multiple partitions of the input samples used in training. For example, during 5-fold, K=5 cross-validation training, a set of input samples is split up into 5 partitions $D_{1,}$ $D_{2,}$....,D_5 having equal sample sizes to the extent possible. The notion of ensuring uniform class representation among the partitions is called stratified cross-validation, which is preferred. To begin, for 5-fold cross-validation, samples in partitions $D_{1,}$ $D_{2,}$....,D_5 are first used for training while samples in partition D_1 are used for testing. Next, samples in groups $D_{1,}$ $D_{2,}$....,D_5 are used for training and samples in partition D_2 used for testing. This is repeated until each partition has been used singly for testing. It is also customary to re-partition all of the input samples e.g. 10 times in order to get a better estimate of accuracy.

5. Result and Discussion

In this research work we made an attempt to determine whether a person has heart disease or not based on the analysis of the heart disease dataset. We have used heart disease data set of 270 patients obtained from UCI Machine Learning Repository. The system starts by taking sample dataset. The total dataset is divided into five groups or folds. Then dataset of these folds are trained and tested using the cross validation technique. Rules are generated for each pattern found in the dataset. These extracted rules are then preserved and applied on test dataset for classification. Some of the rules are as follows:

Heart_disease(absence):-
Thal= normal, Chest_Pain_Type=2, Old_Peak= 0-1.5.
Coverage = 26 samples

Heart_disease(absence):-
Thal=normal, Chest_Pain_Type=3, Old_Peak= 0-1.5.
Coverage = 30 samples

Heart_disease(presence):-
Thal= rev. defect , Chest_Pain_Type=4, Old_Peak= 1.6-3.1.
Coverage = 23 samples

Heart_disease(presence):-
Chest_Pain_Type=4, Thal= rev. defect , Pressure=121-147.
Coverage = 28 samples

Heart_disease(presence):-
Thal= rev. defect , Chest_Pain_Type=4, Slope_Peak=2.
Coverage = 34 samples

Heart_disease(absence):-
Thal=normal, Number_Vessels=0, Cholestoral=126-213.
Coverage = 27 samples

Heart_disease(absence):-
Thal=normal, Number_Vessels=0, Cholestoral=214-301,
Pressure=121-147.
Coverage = 22 samples

Heart_disease(absence):-
Chest_Pain_Type=2, Thal=normal, Old_Peak=0-1.5,
Number_Vessels=0.
Coverage = 22 samples

Heart_disease(presence):-
Thal=rev. defect, Chest_Pain_Type=3, Old_Peak=1.6-3.1.,
Cholestoral=214-301, Slope_Peak=2.
Coverage = 6 samples

Heart_disease(presence):-
Thal=normal, Chest_Pain_Type=4, Max_Heart_Rate=71-103,
Angina=yes.
Coverage = 7 samples

Heart_disease(presence):-
Thal=rev. defect, Chest_Pain_Type=4, Number_Vessels=0,
Old_Peak=1.6-3.1.
Coverage = 9 samples

Heart_disease(presence):-
Thal=rev. defect, Chest_Pain_Type=4, Number_Vessels=1.
Coverage = 15 samples

The result of this research work has two parts:

(i) In the First stage of this research, we calculate the accuracy by applying decision tree induction using ID3 algorithm.

(ii) Then we calculate the accuracy by applying boosting technique; here we have used the distance based K-Nearest Neighbor (k-NN) algorithm.

The aforementioned Five-fold cross validation technique is applied to illustrate the accuracy of the proposed system. Accuracy for each fold and the average are shown in Table 2. Then, the performance of the proposed classification algorithm is computed using ID3 algorithm with k-NN (to measure the distance) as demonstrated in Table 3. The comparison of classification performance of the mentioned two algorithms (ID3 algorithm and ID3 with k-NN to measure the distance) is shown in Figure 5.

Table 2. Classification Performance using ID3 Algorithm

Fold	Size of dataset	Misclassified	Accuracy
Fold1	54	15	72.22%
Fold2	51	11	78.43%
Fold3	55	10	81.82%
Fold4	56	12	78.57%
Fold5	54	11	79.63%
Average accuracy			78.13%

Table 2. Classification Performance using ID3 Algorithm

Fold	Size of dataset	Misclassified	Accuracy
Fold1	54	11	79.63%
Fold2	51	10	80.39%
Fold3	55	09	83.64%
Fold4	56	10	82.14%
Fold5	54	10	81.48%
Average accuracy			81.46%

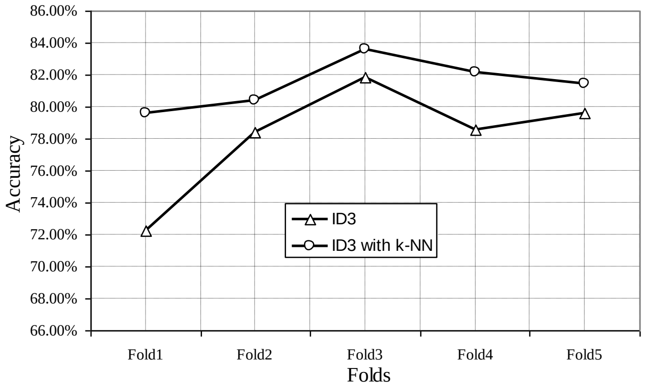


Figure 5. Comparison of Classification Performance for ID3 and ID3 with k-NN Algorithms

6. Conclusions

The proposed algorithms successfully classify heart disease dataset used in this research work. Significant rules are extracted which are useful for understanding the data pattern and behavior of experimental dataset. Although this system is developed with heart disease dataset, it can be used in many other fields where prediction and classification is a must; i.e. breast cancer data set, diabetic data set, Ozone level detection data set and so on.

This system shows a way to design a hybrid classifier and it will help a lot in understanding classification, prediction and other related research fields of data mining. Albeit, attempts have been taken to make this system a complete one it has some sorts of limitation such as dataset is needed in a particular format. Further improvement of this is on process and interested individuals are welcomed for its development.

References

[1] C. Apte and S. Weiss, “Data Mining with Decision Trees and Decision Rules”, Future Generation Computer Systems, vol. 13, **(1997)**, pp. 197-210.

[2] J. R. Quinlan, “Induction of Decision Trees”, Machine Learning, vol. 1, **(1986)**, pp. 81-106.

[3] Md. Nazmul Huda, Masayuki Numao and Takashi Okada, “Multiple Instance Based Learning for Feature Extraction from Chemical Compound”, 21st Century COE International Symposium, **(2005)**, pp. 132-133.

[4] J. Han and M. Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers, **(2004)**.

[5] S. K. Ahmed, U. N. Chowdhury and M. N. Huda, “Extraction of Rules from Chemical Compound Data using ALEPH”, National Conference on Electronics, Information and Telecommunication, Bangladesh, **(2007)**.

[6] Z. Zheng, “Constructing New Attributes for Decision Tree Learning”, Ph.D. Thesis, The University of Sydney, Australia, **(2002)**.

[7] A. K. M. A. Hossain, M. N. Huda, M. Rokanujjaman and R. C. Debnath, “Off-line Fingerprint Verification Using Supervised Neural Network”, International Conference on Computer & Information Technology (ICCIT), vol. 2, **(2003)**, pp. 586-589.

[8] http://archive.ics.uci.edu/beta/datasets.html.

[9] M. Singh, P. K. Wadhwa and P. S. Sandhu, “Human Protein Function Prediction using Decision Tree Induction”, International Journal of Computer Science and Network Security, vol. 7, no. 4, **(2007)**, pp. 92-98.

[10] S. K. Murthy, “Automatic Construction of Decision Trees from Data: A Multi Disciplinary Survey”, Data Mining and Knowledge Discovery, **(1998)**, pp. 345-389.

[11] V. H. Garc á, R. Monroy and M. Quintana, “Web Attack Detection Using ID3”, International Symposium on Professional Practice in AI, vol. 219, **(2006)**, pp. 323-332.

[12] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification”, IEEE Trans. IT, vol. 13, **(1967)**, pp. 21-27.

[13] R. G. Cowel, A. P. Dawid, S. L. Lauritzen and D. J. Spiegelharter, “Probabilistic Networks and Expert Systems”, Springer, **(1999)**.

[14] B. V. Dasarathy, ‘Nearest neighbor (nn) norms: Nn pattern recognition classification techniques”, IEEE Computer Society Press, **(1991)**.

[15] T. G. Dietterich, “Machine learning research: four current directions”, AI Magazine, vol. 18, no. 4, **(1997)**.

[16] Y. Freund and R. E. Schapire, “A short introduction to boosting”, Journal of Japanese Society for Artificial Intelligence, vol. 14, no. 5, **(1999)**, pp. 771-780.

[17] J. Gama, “Combining Classification Algorithms”, PhD Thesis, University of Porto, **(2000)**.

[18] V. Gunes, M. Menard and P. Loonis, “Combination, cooperation and selection of classifiers: A state of the art”, International Journal of Pattern Recognition, vol. 17, **(2003)**, pp. 1303-1324.

[19] T. K. Ho and S. N. Srihati, “Decision combination in multiple classifier systems”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, **(1994)**, pp. 66-75.

[20] B. Sierra and E. Lazkano, “Probabilistic-weighted k nearest neighbor algorithm: a new approach for gene expression based classification”, Proceedings of KES‘02, **(2002)**, pp. 932-939.

[21] J. M. Martinez-Otzeta, B. Sierra, E. Lazkano and A. Astigarraga, "K Nearest Neighbor Edition to Guide Classification Tree Learning", Department of Computer Science and Artificial Intelligence, University of the Basque Country, Spain, **(2005)**.

Authors

Shovon Kumar Pramanik achieved his MSc degree in Computer Science and Engineering from Rajshahi University, year of 2006. Currently he is working as Senior Software Engineer at Brainstation-23 software Development Company in Bangladesh. Previously he has worked as Assistant Programmer at Databiz Soft Limited and Software Engineer at Nordic Soft Ltd, Finland.

Subrata Pramanik did his M.Sc. in Computer Science and Technology from University of Rajshahi, Bangladesh and again in Computer Science from University of Northern British Columbia, Canada. Currently he has been working as an Associate Professor in Dept. of Computer Science & Engineering, University of Rajshahi, Bangladesh.

Bimal K. Pramanik received the B.Sc. degree from the University of Rajshahi, Rajshahi, Bangladesh, in 1996 and the M.Sc. degree from the department of Microelectronics and Information Technology, Royal Institute of Technology, Stockholm, Sweden, in 2004. He has received his Ph.D. from the department of Electronic and Phonic Systems Engineering, Kochi University of Technology, Japan. Now he is working as Associate Professor in The department of Computer Science & Engineering, University of Rajshahi, Bangladesh.

Md. Khademul Islam Molla received B.Sc. and M.Sc. degrees in Electronics and Computer Science from Shahjalal University of Science and Technology, Bangladesh in 1995 and 1997 respectively. He joined in the same Department as a lecturer in 1997. He obtained PhD degree from the Department of Frontier Informatics under the Graduate School of Frontier Sciences, The University of Tokyo, Tokyo, Japan in 2006. He is serving as an Associate Professor in the Dept. of Computer Science and Engineering of the University of Rajshahi, Bangladesh from August, 2006. He was a JSPS postdoctoral fellow in the Dept. of Information and Communication Engineering, The University of Tokyo, Tokyo, Japan from Sep, 2006 to Sep, 2008. Dr. Molla was working as postdoctoral researcher in the field seismic signal processing at the University of Alberta, Edmonton, AB, Canada from Nov, 2010 to Oct, 2011. His research interest includes speech and audio signal processing, blind source separation, statistical and environmental signal processing, biomedical signal and image processing. He is a member of the Institute of Electrical and Electronics Engineers (IEEE).

Md. Ekramul Hamid received his B.Sc and M.Sc degree from the Department of Applied Physics and Electronics, Rajshahi University, Bangladesh. After that he obtained the Masters of Computer Science from Pune University, India. He received his PhD degree from Shizuoka University, Japan. During 1997-2000, he was a lecturer in the Department of Computer Science and Engineering, Rajshahi University. Dr. Hamid was working as Assistant Professor at the King Khalid University, Abha, KSA from 2009 to 2011. He is currently working as an Associate Professor and Chairman in the Department of Computer Science and Engineering, Rajshahi University. His research interests include Digital Signal Processing, Analysis and synthesis of speech signal, Speech Enhancement and Image Processing.