International Journal of Software Engineering and Its Applications
Vol. 7, No. 2, March, 2013

Insights of Data Mining for Small and Unbalanced Data Set Using Random Forests

Hyontai Sug

Division of Computer and Info. Eng., Dongseo University, Busan, Korea
sht@gdsu.dongseo.ac.kr

Abstract

Because random forests are generated with random selection of attributes and use samples that are drawn by boostraping, they are good for data sets that have relatively many attributes and small number of training instances. In this paper an efficient procedure that considers the property of data set having many attributes with relatively small number of attributes in arrhythmia is investigated to predict cardiac arrhythmia is shown. Even though several research results have been published already to find better prediction accuracy based on other methods, a new and better result has been found with the suggested method.

Keywords: *Data mining, random sampling, decision trees*

1. Introduction

Effective data mining in various application areas attracted many researchers' attention [1-3], and decision tree based methods are one of the most used data mining methods, because of their understandability and fast building property of the trees.

Decision tree induction algorithms divide a training data set based on some branching criteria during tree building process, so the performance of trained decision trees is more heavily dependent on the training data sets than other data mining algorithms, for example, such as neural networks [4]. In order to avoid this property, we may prepare several samples based on sampling with replacement, and generate decision trees from the samples. Random forests [5] are based on such method called boostraping [6], and multiple of decision trees are generated with limited random selection on attributes, and do no pruning. The time complexity to generate a decision tree without pruning is less than the number of attributes multiplied by O(n log n) where n is the number of training instances, since decision tree grows up to the number of attributes at most, and we may need sorting for continuous attributes at each level of the tree. Hence, tree generation time is relatively fast in random forests. The performance of random forests is known to be good for the cases like small training data set size and some possible errors in the data set.

We are interested in finding some better data mining model for the data set called 'arrhythmia'. The main concern is to predict presence or absence of cardiac arrhythmia accurately. There are some previous researches for the data set. Because the data set contains relatively many attributes than the number of instances, researchers focused on effective attribute selection methods. Pappa, *et al.*, [7] used genetic algorithm to select appropriate attributes. Genetic algorithm needs proper fitness function so that we need good domain knowledge to set the fitness fuction. Cohen and Ruppin [8] used game theory-based algorithm called contribution selection algorithm (CSA) to select features. They transformed the data set into binary classification problem to make the data set to be solvable by using their algorithm. But, because the original data set consists of 16

classes, we need some ensemble method for us to use the result. Random forests do not have any limitation in the number of classes.

In Section 2, we present our method in detail, and Section 3 we show the result of experiment. Finally Section 4 provides some conclusions.

2. The Method

Assume that N is the number of instances in a training data set. In order to build a tree, we generate a training data set by doing sampling with replacement N times. So, identical instances can be sampled several times in the sampled data. We use this new sampled data to generate a tree. Statistically, 63.2% of instances are sampled in average from the original training data set, and the sampling method is called bootstrap method [9].

In the bootstrap method each instance in the training data set is sampled with the probability of 1/N. So the probability of not being selected is 1- 1/N. Therefore, if we repeat the sampling with replacement process N times, the probability of not being selected is $(1 - 1/N)^N$. If N is large, this probability is similar to $e^{-1} = 0.368$. So the probability of not being selected in the bootstrap method is 36.8%. In other words, 36.8% of the instances in the training set are not selected in average in the bootstrap method.

After training data set is made, we can generate decision trees without pruning. Decision tree generation method of random forests consider yet unselected attributes in each subtree of the tree, and the number of attributes is limited to some predefined number, and, moreover, the attributes are chosen randomly. The predefined number of attributes may be set by the default value suggested by Breiman [10]. The number can be the first integer less than $\log_2 A + 1$, or square root of A and may be half and double of the number, where A is the number of attributes. But we should be careful for the number, because different number can generate different result, so that we should consider the property of target data set in setting the number. Because the data set contains relatively large number of attributes which is 280, we will consider it increasingly from default value.

Another factor that affects the accuracy of random forests is the number of decision trees in the forests. We may generate exhaustively, if we have enough time. Because the data set will be generated random sampling with replacement and the sampling will be performed N times, we will generate the trees as multiples of N to reflect the sampling method. We will follow the following procedure to generate random forests.

Procedure:
Begin

1. C := square root of |A| where |A| is the number of attributes;
2. **Repeat**
3. **For** t = {T | N, N*2}
4. Generate random forests with parameters C & t where C is the number of attributes to pick randomly and t is the number of tree in the forests
5. **End For**;
6. C := C*2;
7. **Until** C < |A|; **End.**

3. Experiment

Experiments were run using a data set called ‘arrhythmia’ in UCI machine learning repository [11]. The arrhythmia data set was prepared to distinguish between the presence and absence of cardiac arrhythmia. The number of instances in the data set is 452, and the number of attributes is 280. There is one class attribute which has 16 different values. Class 1 represents 'normal' ECG, and classes 2 to 15 represents different classes of arrhythmia, and class 16 represents unclassified ones. The number of instances for each class is in Table 1.

Table 1. The Number of Instances for each Class

Class	Number of instances
1	245
2	44
3	15
4	15
5	13
6	25
7	3
8	2
9	9
10	50
11	0
12	0
13	0
14	4
15	5
16	22
Total	452

So, the data set has unbalanced class distribution. Especially there are no instances of class 11, 12, and 13. Figure 1 shows the class distribution graphically. In the figure x axis represents class label, and y axis represents the number of instances.

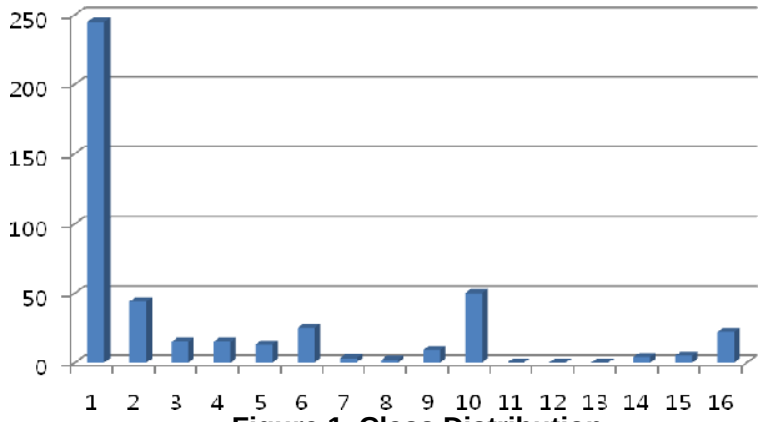

Figure 1. Class Distribution

Inthe experiment 10-fold cross validation was used. Random forests in weka were used for the experiment. Weka is a data mining package written in java [12]. The following Table 2 shows the accuracy based on the number of attributes to consider and multitude of trees. The number of trees is 407, and 814. The number of attributes to pick to generate tree in the forests is 16, 32, 64, 128, and 256. So, total of 10 different random forests were made.

Table 2. The Accuracy of Random Forests for each Number of Attributes to pick Randomly and Different Number of Trees

No. of trees No. of attr. to pick randomly	407	814
16	71.6814%	71.2389%
32	73.8938%	74.3363%
64	76.5487%	**76.7699**%
128	75.6637%	75.4425%
256	75.4425%	75.4425%

Random forests(64, 814) has the accuracy 76.7699%, and this is the best accuracy according to literature survey [7]. Single tree by C4.5 [13] achieved the accuracy 64%~70%. Figure 2-14 shows ROC curve for each class. In the figures left figure shows ROC of the first random forests, say RF1, which have parameters (16, 407) and right figure shows the ROC of the best random forests, say RF6, which has accuracy of 76.7699%. The weighted average of ROC area of RF1 and RF6 is 0.893 and 0.9 respectively.

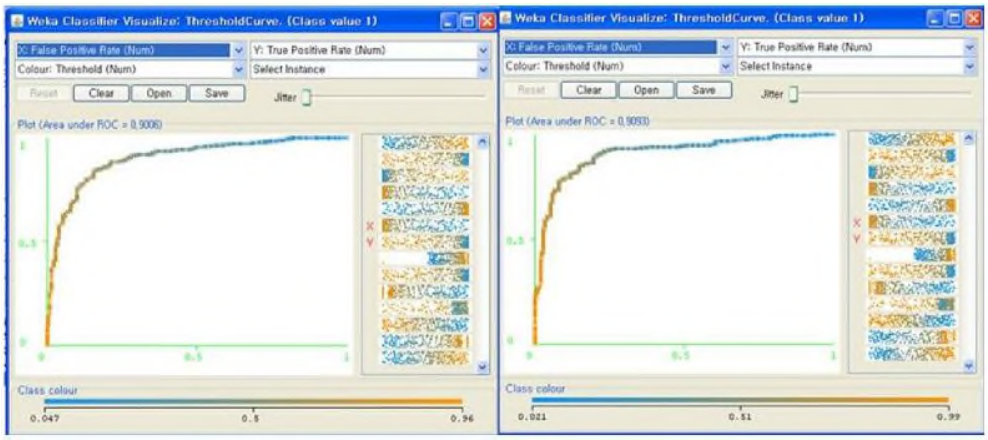

Figure 2. ROC of Class 1 for RF1 (left) and RF6 (right)

Area under ROC of class 1 is 0.9006 and 0.9093 for RF1 and RF6 respectively. Note that the number of instances of class 1 is 245. Class 1 is majority class of the data set. The total number of instances is 452.

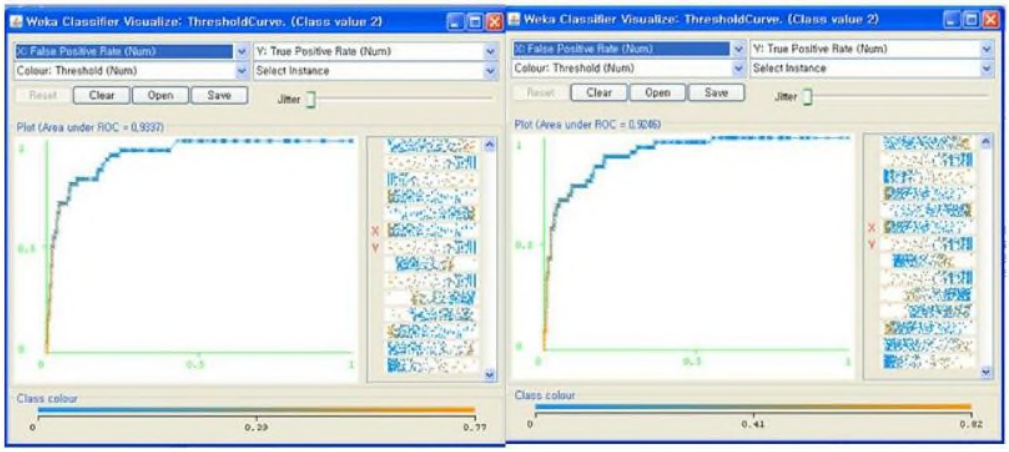

Figure 3. ROC of Class 2 for RF1(left) and RF6(right)

Area under ROC of class 2 is 0.9337 and 0.9246 for RF1 and RF6 respectively. So, ROC of RF6 is slightly worse than that of RF1. Note that the number of instances of class 2 is 44.

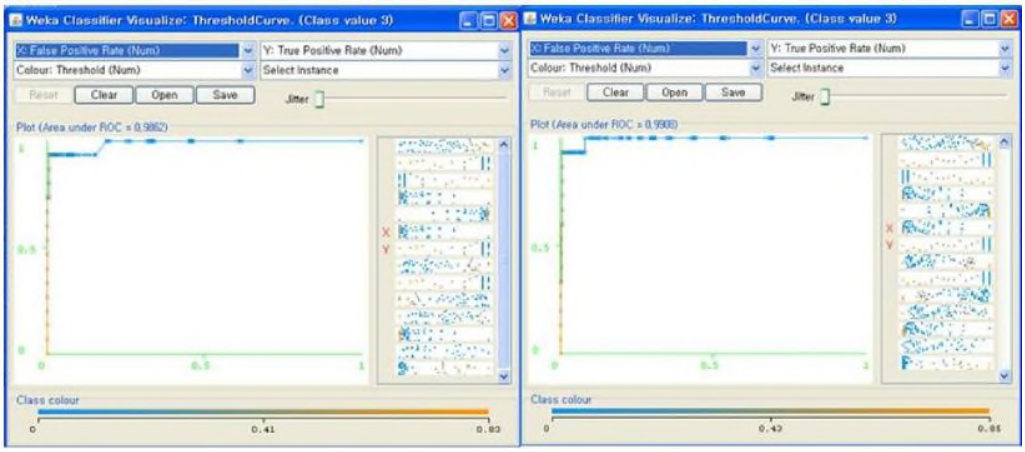

Figure 4. ROC of Class 3 for RF (left) and RF6(right)

Area under ROC of class 3 is 0.9862 and 0.9908 for RF1 and RF6 respectively. Note that the number of instances of class 3 is 15.

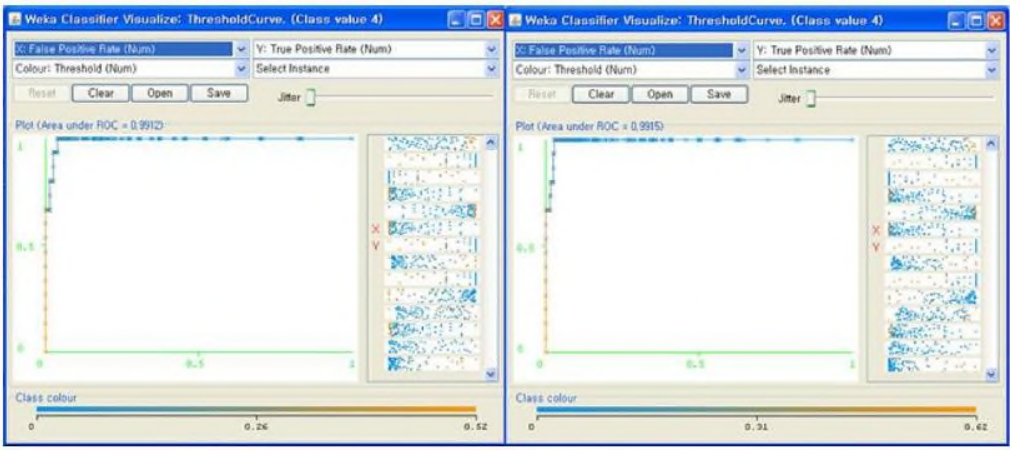

Figure 5. ROC of Class 4 for RF1(left) and RF6(right)

Area under ROC of class 4 is 0.9912 and 0.9915 for RF1 and RF6 respectively. Note that the number of instances of class 4 is 15.

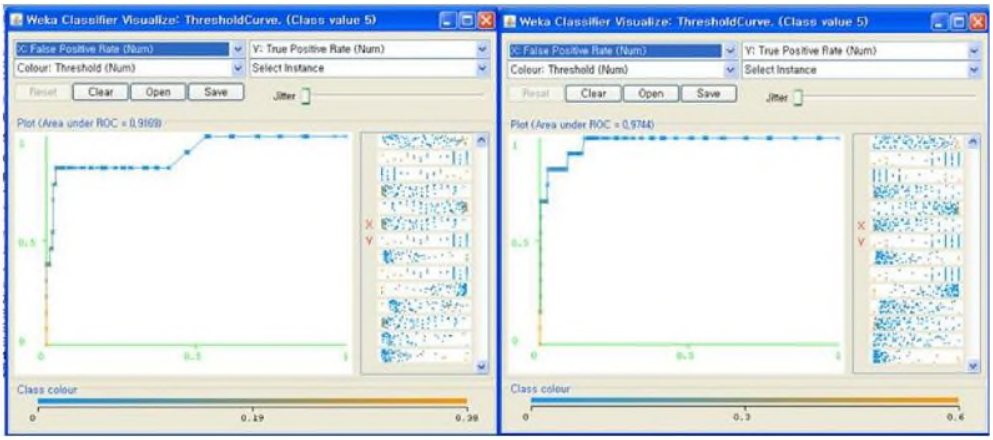

Figure 6. ROC of Class 5 for RF1(left) and RF6(right)

Area under ROC of class 5 is 0.9169 and 0.9744 for RF1 and RF6 respectively. Note that the number of instances of class 5 is 13.

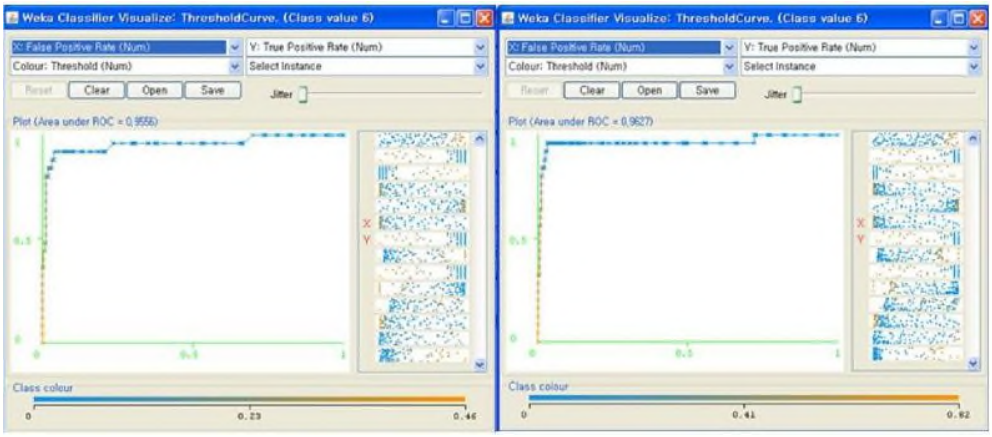

Figure 7. ROC of Class 6 for RF1(left) and RF6(right)

Area under ROC of class 6 is 0.9556 and 0.9627 for RF1 and RF6 respectively. Note that the number of instances of class 6 is 25.

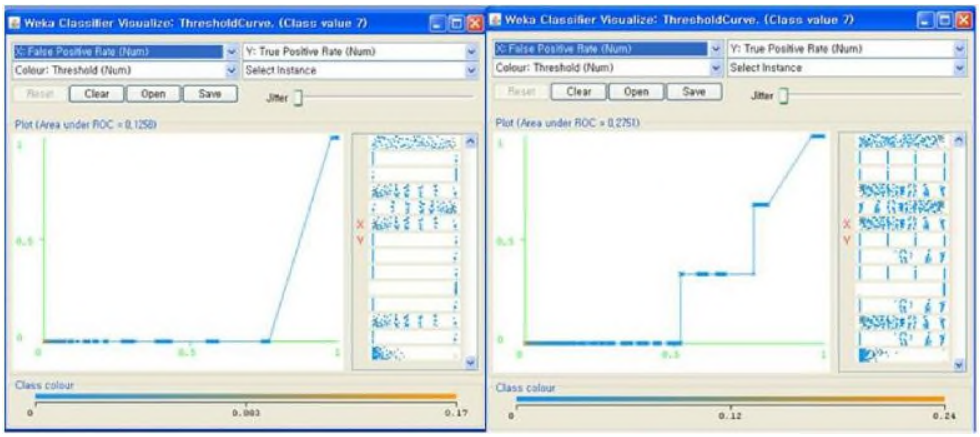

Figure 8. ROC of Class 7 for RF1(left) and RF6(right)

Area under ROC of class 7 is 0.1258 and 0.2751 for RF1 and RF6 respectively. Note that the number of instances of class 7 is 3.

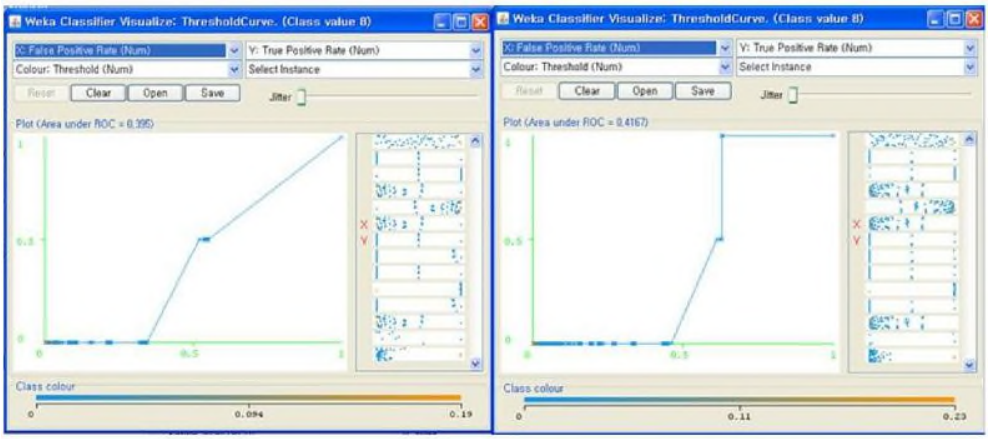

Figure 9. ROC of Class 8 for RF1(left) and RF6(right)

Area under ROC of class 8 is 0.395 and 0.4167 for RF1 and RF6 respectively. Note that the number of instances of class 8 is 2.

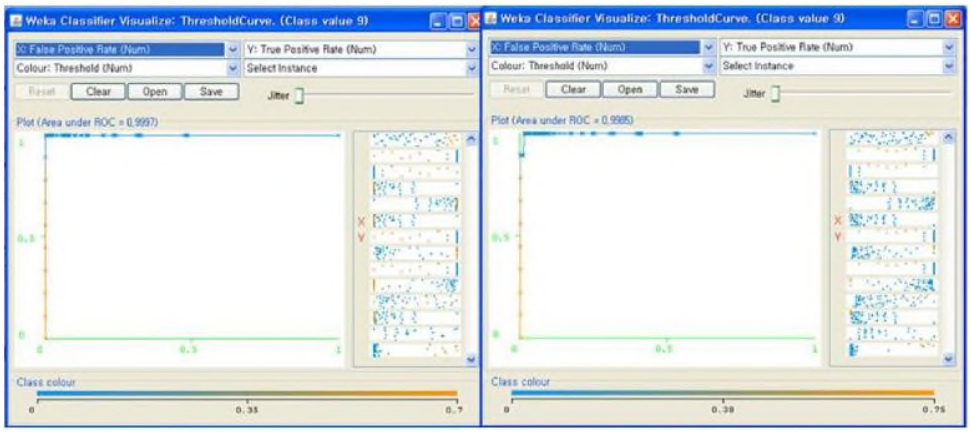

Figure 10. ROC of Class 9 for RF1(left) and RF6(right)

Area under ROC of class 9 is 0.9997 and 0.9985 for RF1 and RF6 respectively. Note that the number of instances of class 9 is 9.

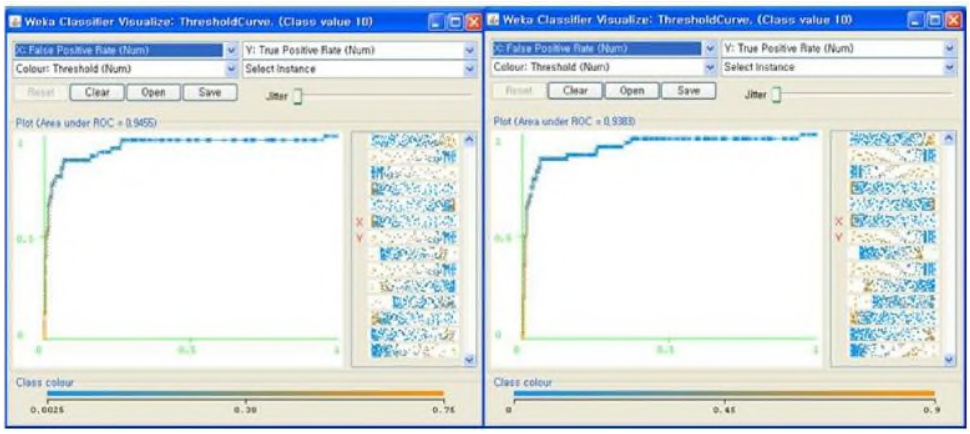

Figure 11. ROC of Class 10 for RF1(left) and RF6(right)

Area under ROC of class 10 is 0.9455 and 0.9383 for RF1 and RF6 respectively. The ROC of RF6 is slightly worse than that of RF1. Note that the number of instances of class 10 is 50.

For class 11, 12, 13, area under ROC is not available, because there are no instances.

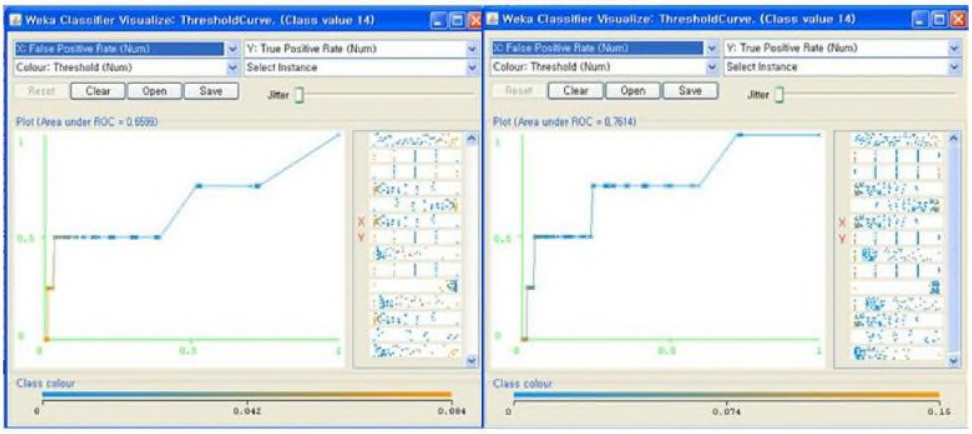

Figure 12. ROC of Class 14 for RF1(left) and RF6(right)

Area under ROC of class 14 is 0.6599 and 0.7614 for RF1 and RF6 respectively. Note that the number of instances of class 14 is 4.

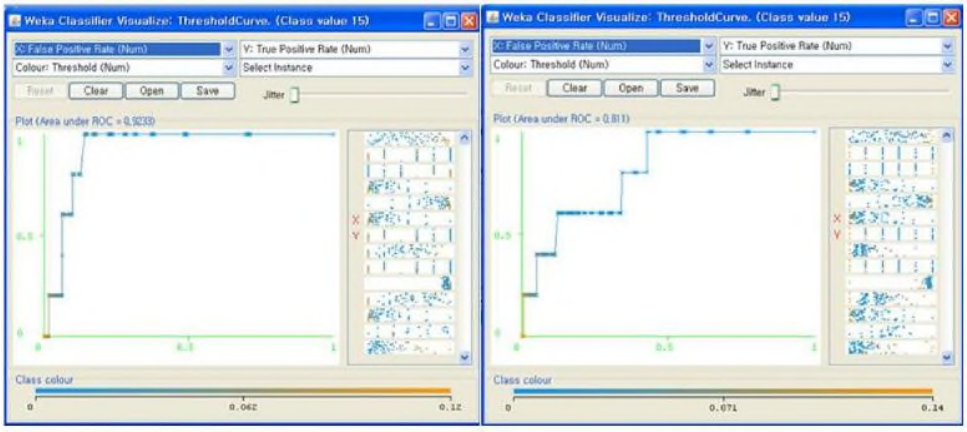

Figure 13. ROC of Class 15 for RF1(left) and RF6(right)

Area under ROC of class 15 is 0.9233 and 0.811 for RF1 and RF6 respectively. The ROC of RF1 is better than that of RF6. Note that the number of instances of class 15 is 5.

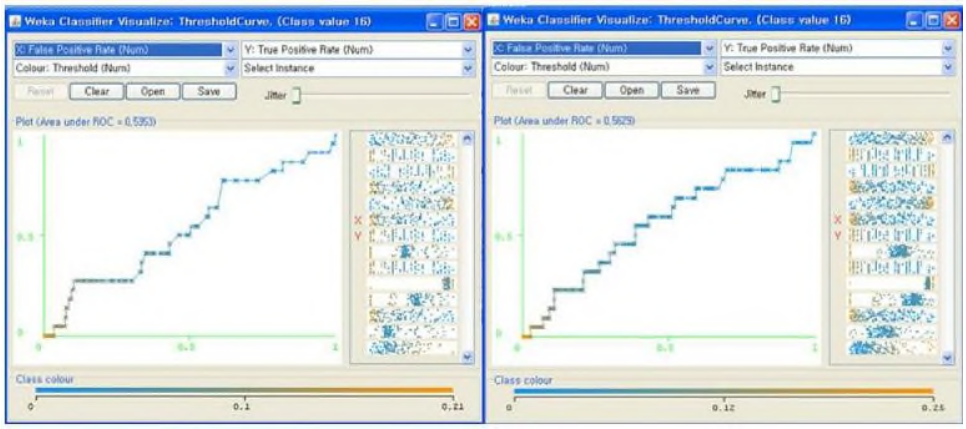

Figure 14. ROC of Class 16 for RF1(left) and RF6(right)

Area under ROC of class 16 is 0.5353 and 0.5629 for RF1 and RF6 respectively. Note that the number of instances of class 16 is 22.

4. Conclusions

Random forests are good for data of some insufficient information and irrelevant attributes. Constructing random forests generally does not need much computing resources, because post processing like pruning is net performed, unless the forests consist of tremendous number of trees.

Therefore, because we believe that 'arrhythmia' data set has many attributes and not large number of instances, we want to find some best predictive accuracy with random forests to predict presence or absence of cardiac arrhythmia. Even though several researches were done using other data mining methods to find better classification accuracy for the data set, we have found that we could find some better results. An effective method considering the properties of the data set is suggested to find the result. The experiments supported us by providing a good result in prediction accuracy.

Acknowledgement

This work was supported by Dongseo University, "Dongseo Frontier Project" Research Fund of 2012.

References

[1] K. Lee and H. Cho, "Performance of Ensemble Classifier for Location Prediction task: Emphasis on Markov Blanket Perspective", International Journal of u-and e-Service, Science and Technology, vol. 3, no. 3, **(2010)**.
[2] B. Vo and B. Le, "Fast Algorithm for Mining Generalized Association Rules", International Journal of Database Theory and Applications, vol. 2, no. 3, **(2009)**.
[3] A. Majid and T. Choi, "A New Ensemble Scheme for Predicting Human Protains Subcellular Locations", International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 3, no. 1, **(2010)**.
[4] L. O. Hall and X. Liu, "Why are Neural Networks Sometimes Much More Accurate than Decision Trees: An Analysis on a Bio-Informatics Problem", Proceedings of IEEE International Conference on Systems, Man & Cybernetics, **(2003)**.

[4] M. D. Rahman and M. Z. Islam, "A Decision Tree-based Missing Value Imputation Technique for Data Pre-processing", Proceedings of the Ninth Australasian Data Mining Conference (AusDM 11), **(2011)**, pp. 41-50.
[5] L. Breiman, "Random Forests", Machine Learning, 45, **(2001)**.
[6] J. Han, M. Kamber and J. Pei, "Data Mining: concepts and techniques, 3rd ed., Morgan Kaufmann publishers, **(2011)**.
[7] G. L. Pappa, A. A. Freitas and C. A. A. Kaestner, "A Multiobjective Genetic Algorithm for Attribute Selection", Proc. 4th Int. Conf. on Recent Advances in Soft Computing (RASC-2002), **(2002),** pp. 116-121.
[8] S. Cohen and E. Ruppin, "Feature selection based on the shapley value", Proceeding IJCAI'05 Proceedings of the 19th international joint conference on Artificial intelligence, **(2005),** pp. 665-670.
[9] B. Efron and R. Tibshirani, "Improvements on Cross-Validation: The 632 Bootstrap Method", Journal of the American Statistical Association, vol. 92, pp. 438, **(1997)**.
[10] L. Brieiman, "A. Cutler, random Forests", http://www.stat.berkeley.edu/users/breiman/Random Forests/
[11] A. Frank and A Asuncion, "UCI Machine Learning Repository", [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, **(2010)**.
[12] Weka, http://www.cs.waikato.ac.nz/ml/weka/
[13] J. R. Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, Inc., **(1993)**.

Author

Hyontai Sug received BS degree in computer science and statistics from Busan national university, Korea, and MS degree in applied computer science from Hankuk university of foreign studies, Korea, and Ph.D. degree in computer and information science and engineering from university of Florida, USA. Currently, he is the associate professor of Dongseo University, Korea. His research interests include data mining and database applications.