Performance Comparison of Different Over-Sampling Rates of Decision Trees for the Class of Higher Error Rate in the Liver Data Set

Hyontai Sug

***Abstract*—**Because of the comprehensibility of decision trees they are good tools for data mining of data sets in medicine domain. Liver disorder disease is a disease in such domain, so that decision trees can be very useful data mining tools to diagnose the disease. But the accuracy of decision trees may be limited due to insufficient data in the domain. In order to generate more accurate decision trees for the disease this paper suggests a method based on over-sampling for the data instances that are in the class of high error rate so that we can compensate the insufficiency. Experiments were done with two representative decision tree algorithms, C4.5 and CART, and found very good result in accuracy which is up to 81.16%.

***Keywords*—**Decision trees, C4.5, CART, over-sampling, liver disorder disease.

I. Introduction

BECAUSE we can easily understand the knowledge structures of decision trees, they can be very good data mining tools for data sets in medicine domain, and this is one of the main reasons why decision trees are widely accepted in the domain [1, 2]. Another good point of decision trees is that it is very straightforward to transform the structure of decision trees into rules so that the rules can be used, for example, to build expert systems [3, 4]. There are many examples that use decision trees well [5, 6, 7, 8]. But, even though the good points the training algorithms of decision trees have the weak point of disdaining minor classes. Minor classes are classes that have relatively higher error rates among classes.

Decision tree generation algorithms divide the training data set based on their own branching criteria. So, due to the dividing as each subtree is being built, each branch in the subtree becomes to have less training instances. So, the reliability of lower branches becomes worse than upper branches due to the smaller size of training examples. Therefore, the classification accuracy of minor classes can become less accurate than that of major classes.

Another issue in generating decision trees is random sampling. Because we may not have a perfect data set for data mining, and we usually don't have exact knowledge about the property of data sets, we may resort to random sampling [9]. But, the trained knowledge models based on the random samples are likely dependent on the samples. Moreover, due to the data fragmentation, it is known that the decision tree algorithms are more dependent upon the training data sets, while other machine learning algorithms like neural networks [10] that do not divide the data set during training are less dependent on.

Liver is the largest internal organ in the human body, and it is known that the organ is responsible for more than one hundred functions of human body. The complexity of this organ makes it easily affected by disease of disorder. So diagnosing liver disorder disease is a high interest to researchers and doctors [11, 12], and decision trees have been considered a good data mining method because of their good understandability [13]. We are interested in finding better decision trees for the data set of liver disorder called 'BUPA liver disorder' that has relatively small number of instances. So in order to overcome the problem of neglecting minority classes with decision tree generation algorithms, we need a new technique so that the minor classes in the data set are treated more importantly.

In section 2, we provide the related work to our research, and in sections 3 we present our method of experimentation with the explanation on the principles of decision trees. Experiments were run to see the effect in section 4. Finally section 5 provides conclusions.

II. Related Work

It is known that generating optimal decision trees is NP-complete problem [14] so that we rely on greedy algorithms to split branches. As a result, the generated decision trees may not be optimal. There have been a lot of efforts to build better decision trees [15]. Among them C4.5 [16] and CART [17] can be two representative decision tree algorithms, because the two algorithms are frequently referred, and C4.5 is often referred in engineering and business domain, while CART is often referred in medicine domain. C4.5 uses an entropy-based measure to split branches based on attribute values, and the measure selects the most certain split among possible splits of candidate attributes. So classes that have more certain splits with respect to entropy are preferred.

CART uses a purity-based measure, and the algorithm splits the training data set based on how probably the subsets become purer for a class, and it spends more time to generate smaller

Manuscript received November 25, 2011: Revised version received February 27, 2012. This work was supported by Dongseo University, "Dongseo Frontier Project" Research Fund of 2011.

H. Sug is with Dongseo University, Busan, 617-716, Korea (phone: +82-51-320-1733; fax: +82-51-327-8955; e-mail: shtdaum@ hanmail.net).

trees, so CART produces relatively smaller trees than C4.5. The splitting measure of the decision tree algorithms prefers the most certain split among possible splits of candidate attributes. So, major classes are preferred also, because there are more instances of major classes in data set, and this fact makes it more certain in splitting.

Because the training process of decision trees is a kind of inductive process, and the data set is fragmented in the training process, the performance of trained decision tree is heavily dependent on the composition of training data set. In [18, 19] class imbalance has different effect in neural networks for medical domain data so that we can see the importance of data set for the task of data mining. SMOTE [20] used synthetic data generation method for minor classes to cope with the situation of data shortage in minor classes, and showed that it is effective for decision trees. A weak point of the approach is that we need to understand the characteristics of data to synthesize effective data.

There has been much research interest for better prediction models for liver disorder disease. In [21, 22] undesirable features were eliminated to find better prediction models of neural networks, and an expert system was made based on the generated knowledge models. In [23] neural network having hidden layer of adaptive activation function and output layer of fixed sigmoid function were used, and better rules were generated than [22]. A potential problem of the neural network-based approach in [23] is data over-fitting, because their knowledge models have comparatively high accuracy without providing testing conditions. This fact was shown by other papers like [24, 25]. In [24] four different data mining algorithms like Naïve Bayes classifier, C4.5, neural networks, and support vector machines were tried, and the accuracy of the algorithms is 56.52% ~ 71.59% with 10-fold cross-validation. The accuracy of C4.5 is 68.69%. In [25] outliers were removed to improve k-NN and neural networks for the same data set.

III. METHOD

We are interested in finding better decision trees for BUPA liver disorder data set [26]. Because the data set is relatively small and has somewhat high error rate, we want to compensate the property of disdaining minor classes in decision tree generation algorithms. Since decision tree algorithms do not give high priority to minor classes in splitting branches, it is highly possible that instances of minor classes will be treated in the lower part of the tree, and this treatment may increase misclassification rate for the classes. So we want decision tree algorithms to treat the instances of minor classes more importantly. In order to understand how we can make decision trees treat minor classes more importantly, let's see the principle of decision trees, C4.5 and CART.

A. Principle of C4.5 Decision Tree Generation Algorithm

Let's assume that we have r classes, q_1, q_2, ..., q_r, and attribute X can have n different values with probability of each being p_1, p_2, ..., p_n, and the probability for each value with respect to each class being p_{11}, p_{12}, ..., p_{1r}, p_{21}, p_{22}, ..., p_{2r}, ..., p_{n1}, p_{n2}, ..., p_{nr}. The entropy for a value a_i in X is as follows;

$$- \sum_{j=1 \sim r} p_{ij} \log p_{ij} \quad (1)$$

So, the expected information content of attribute X is,

$$Info_x(T) = \sum_{i=1 \sim n} p_i \left(- \sum_{j=1 \sim r} p_{ij} \log p_{ij} \right) \quad (2)$$

where T is the parameter. The information content of T is defined as the following equation;

$$Info(T) = - \sum_{i=1 \sim r} q_i \log q_i \quad (3)$$

The gain of attribute X is defined as,

$$Gain(X) = Info(T) - Info_x(T) \quad (4)$$

So, in order to avoid for key-like attributes to be preferred, the gain is divided by the following equation;

$$Split\_info(X) = -\sum_{i=1 \sim n} p_i \log p_i \quad (5)$$

Hence, gain ratio is used as splitting criterion at each root of the subtrees.

$$Gain\ ratio = Gain(X) / Split\_info(X) \quad (6)$$

During the training stage frequency ratio is used instead of probability. After generating fully grown tree, pruning is performed to make a simpler tree with smaller expected error rate.

B. Principle of Classification and Regression Tree (CART) Generation Algorithm

CART uses GINI index for splitting. GINI index calculates the purity of class value distribution in nodes. A node is purer, if the class value distribution in the node is in some skewed manner. For example, if we have two different nodes i and j like in fig. 1, node i is purer than node j, because it has more skewed distribution in the class values. In fig. 1 circles and deltas represent instances with class value of circle or delta.

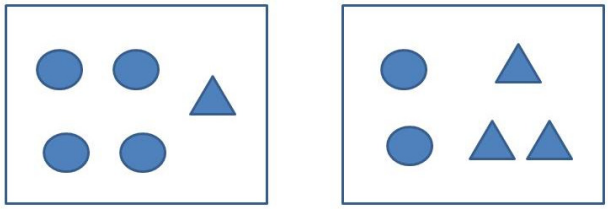

Fig. 1. The concept of purity

GINI index has the following equation;

$$G = \sum_{i=1\sim r} p_i\ (1 - p_i) = 1 - \sum_{i=1\sim r} p_i^2 \quad (7)$$

CART does binary split and selects the purist split among possible candidate attributes based on the most diminishing G values. Pruning in CART is done similarly with that of C4.5. It prunes in the following criteria;

$$\Delta(\text{predicted misclassification cost}) < \alpha\Delta(\text{tree complexity}) \quad (8)$$

More severe pruning will be performed, as α is increased.

As we see from the splitting methods of the algorithms, the algorithms fragment a training data set, and also disdain classes that do not have some skewed distribution in class values.

C. Experimentation Procedure

We are interested in finding better decision trees for BUPA liver disorder data set. Because decision tree algorithms disdain minor classes in splitting branches, we want decision tree algorithms to treat the instances of minor classes more importantly. In order to do this, we increase the number of instances of minor classes by duplication. The following is a brief description of the procedure of the experiment.

INPUT: BUPA liver disorder data set
OUTPUT: decision trees of C4.5 and CART
Begin
Do random sampling of size 50% and 80% seven times;
For each sample data set **Do**
  Generate decision trees of C4.5 and CART;
  X := 20%;
  Repeat
    Do X% more sampling for minor class;
    Generate decision trees of C4.5 and CART;
    X := X + 20%;
  Until X = 300%;
End For;

In the above procedure we duplicate the instances of minor class by 20 percents until the over-sampling rate becomes 300%. The minor class is the class that has higher error rate. The sample size is 50% and 80% of the original data set. The rest of the data is used for testing.

IV. EXPERIMENTATION

Experiments were run using a data set in UCI machine learning repository [27] called 'liver disorder' [26] to see the effect of the method. The number of instances is 345. There are 145 instances in class 1 and 200 instances in class 2. Class 1 is the minor class, because its error rate is 68/145=46.9%, while the error rate of class 2 is 40/200=25% based on 10-fold cross-validation in C4.5. The overall error rate is 31.3043%. There are six continuous attributes as independent attributes, and one attribute is class attribute that has value of 1 or 2. There are no missing values in all attributes. Please see table 1 for details of the attributes.

Table 1. The details of attributes

Attribute	Meaning
mcv	Mean corpuscular volume
alkphos	Alkaline phosphtase
Sgpt	Alamine aminotransferase
Sgot	Aspartate aminotransferase
Gammagt	Gamma-glutamy transpeptidase
Drinks	Number of half-pint equivalents of alcoholic beverages drunk per day

C4.5 and CART were used to generate decision trees for seven random sample sets. Sample sets of size 50% and 80% of original data set were used. Remaining data were used for test. The following table 2 shows the accuracy of decision trees generated by C4.5 for sample sets of size 50% of the original data set. The rest 50% were used for testing.

Table 2. The accuracy of C4.5 for sample sets of size 50% of the original data set

Sample set	Accuracy
1	65.32%
2	64.74%
3	53.45%
4	60.12%
5	66.47%
6	63.58%
7	61.27%
Average	62.14%

From table 3 to table 7 shows the accuracy for decision tree algorithm C4.5 with minor class over-sampling.

Table 3. The accuracy of C4.5 for sample sets of size 50% of the original data set with minor class over-sampling rate of 20%, 40%, 60%

Sample set	Over-samp. 20%	Over-samp 40%	Over-samp. 60%
1	63.58%	63.01%	56.65%
2	63.22%	67.81%	62.07%
3	54.60%	51.72%	66.67%
4	55.75%	56.32%	59.20%
5	65.52%	64.94%	66.67%
6	58.96%	63.58%	52.02%
7	53.76%	69.94%	62.43%
Average	59.34%	62.47%	60.82%

Table 4. The accuracy of C4.5 for sample sets of size 50% of the original data set with minor class over-sampling rate of 80%, 100%, 120%

Sample set	Over-samp. 80%	Over-samp 100%	Over-samp. 120%
1	67.63%	68.21%	63.58%

2	62.07%	62.43%	62.07%
3	55.75%	56.32%	58.05%
4	59.20%	61.27%	59.20%
5	67.82%	63.58%	55.75%
6	61.27%	57.23%	55.49%
7	62.43%	65.90%	50.86%
Average	62.31%	62.13%	57.86%

Table 5. The accuracy of C4.5 for sample sets of size 50% of the original data set with minor class over-sampling rate of 140%, 160%, 180%

Sample set	Over-samp. 140%	Over-samp 160%	Over-samp. 180%
1	67.63%	71.10%	68.21%
2	63.22%	60.34%	58.62%
3	62.64%	65.52%	63.79%
4	51.72%	61.49%	53.45%
5	60.92%	62.64%	52.88%
6	58.38%	61.85%	64.16%
7	59.54%	56.07%	58.38%
Average	60.58%	62.72%	59.93%

Table 6. The accuracy of C4.5 for sample sets of size 50% of the original data set with minor class over-sampling rate of 200%, 220%, 240%

Sample set	Over-samp. 200%	Over-samp 220%	Over-samp. 240%
1	59.54%	67.64%	60.69%
2	62.69%	59.77%	60.92%
3	56.32%	54.60%	65.52%
4	64.16%	60.34%	62.07%
5	60.92%	64.94%	60.92%
6	64.16%	61.27%	58.96%
7	56.07%	56.07%	62.43%
Average	60.58%	62.72%	59.93%

Table 7. The accuracy of C4.5 for sample sets of size 50% of the original data set with minor class over-sampling rate of 260%, 280%, 300%

Sample set	Over-samp. 260%	Over-samp 280%	Over-samp. 300%
1	66.47%	64.74%	65.32%
2	57.47%	59.77%	59.77%
3	60.34%	60.34%	55.75%
4	60.92%	56.90%	60.34%
5	58.62%	61.49%	66.67%
6	59.54%	64.74%	59.54%
7	56.07%	60.12%	59.54%
Average	59.92%	61.16%	60.99%

The following table 8 shows the accuracy of decision trees generated by CART for sample sets of size 50% of the original data set. The rest 50% were used for testing.

Table 8. The accuracy of CART for sample sets of size 50% of the original data set

Sample set	Accuracy
1	65.32%
2	61.27%
3	67.82%
4	58.38%
5	65.90%
6	60.69%
7	62.43%
Average	63.12%

From table 9 to table 13 shows the accuracy for decision tree algorithm CART with minor class over-sampling.

Table 9. The accuracy of CART for sample sets of size 50% of the original data set with minor class over-sampling rate of 20%, 40%, 60%

Sample set	Over-samp. 20%	Over-samp 40%	Over-samp. 60%
1	63.58%	57.80%	58.38%
2	60.92%	60.92%	60.92%
3	63.22%	62.64%	59.77%
4	58.05%	59.20%	53.45%
5	64.94%	63.22%	67.24%
6	59.54%	67.63%	64.16%
7	66.47%	69.94%	67.63%
Average	62.39%	63.05%	61.65%

Table 10. The accuracy of CART for sample sets of size 50% of the original data set with minor class over-sampling rate of 80%, 100%, 120%

Sample set	Over-samp. 80%	Over-samp 100%	Over-samp. 120%
1	65.32%	67.05%	64.16%
2	63.79%	63.58%	63.22%
3	59.77%	60.92%	47.70%
4	54.60%	61.85%	60.92%
5	68.39%	69.85%	65.52%
6	55.49%	57.23%	55.49%
7	65.90%	65.32%	65.32%
Average	61.89%	63.69%	60.33%

Table 11. The accuracy of CART for sample sets of size 50% of the original data set with minor class over-sampling rate of 140%, 160%, 180%

Sample set	Over-samp. 140%	Over-samp 160%	Over-samp. 180%
1	59.54%	64.16%	63.58%
2	66.67%	60.34%	64.37%
3	64.37%	51.72%	60.92%
4	60.92%	62.07%	62.07%
5	67.24%	67.24%	63.22%
6	57.86%	57.23%	49.13%

7	65.90%	62.43%	64.74%
Average	63.21%	60.74%	61.15%

Table 12. The accuracy of CART for sample sets of size 50% of the original data set with minor class over-sampling rate of 200%, 220%, 240%

Sample set	Over-samp. 200%	Over-samp 220%	Over-samp. 240%
1	52.60%	62.43%	54.91%
2	60.69%	62.64%	62.64%
3	54.02%	54.60%	54.60%
4	53.76%	59.77%	54.02%
5	67.63%	63.22%	62.07%
6	56.65%	52.60%	56.07%
7	63.58%	63.58%	61.85%
Average	58.42%	59.83%	58.02%

Table 13. The accuracy of CART for sample sets of size 50% of the original data set with minor class over-sampling rate of 260%, 280%, 300%

Sample set	Over-samp. 260%	Over-samp 280%	Over-samp. 300%
1	54.91%	57.80%	53.76%
2	61.49%	64.37%	62.64%
3	54.60%	52.87%	52.87%
4	54.60%	54.60%	55.17%
5	63.22%	61.49%	62.07%
6	56.07%	54.91%	52.02%
7	61.27%	60.69%	61.27%
Average	58.02%	58.10%	57.11%

As an alternative, random sampling in sample size of 80% of the original data set was tried. The following table 14 shows the accuracy of decision trees generated by C4.5 for sample sets of size 80% of the original data set. The rest 20% were used for testing.

Table 14. The accuracy of C4.5 for sample sets of size 80% of the original data set

Sample set	Accuracy
1	60.87%
2	66.67%
3	69.57%
4	62.32%
5	56.52%
6	72.46%
7	66.67%
Average	65.01%

From table 15 to table 19 shows the accuracy for decision tree algorithm C4.5 with minor class over-sampling.

Table 15. The accuracy of C4.5 for sample sets of size 80% of the original data set with minor class over-sampling rate of 20%, 40%, 60%

Sample set	Over-samp. 20%	Over-samp 40%	Over-samp. 60%
1	70.01%	63.77%	63.77%
2	60.87%	56.52%	60.87%
3	68.12%	63.77%	55.07%
4	66.67%	73.91%	63.77%
5	62.32%	62.32%	65.22%
6	71.01%	76.81%	66.67%
7	65.22%	66.67%	62.32%
Average	66.32%	66.25%	62.53%

Table 16. The accuracy of C4.5 for sample sets of size 80% of the original data set with minor class over-sampling rate of 80%, 100%, 120%

Sample set	Over-samp. 80%	Over-samp 100%	Over-samp. 120%
1	62.32%	59.42%	66.67%
2	69.57%	60.87%	56.52%
3	71.01%	57.97%	55.07%
4	62.32%	66.67%	63.77%
5	63.77%	59.42%	57.97%
6	78.26%	68.12%	68.12%
7	59.42%	65.22%	60.87%
Average	66.67%	62.53%	61.28%

Table 17. The accuracy of C4.5 for sample sets of size 80% of the original data set with minor class over-sampling rate of 140%, 160%, 180%

Sample set	Over-samp. 140%	Over-samp 160%	Over-samp. 180%
1	52.17%	60.87%	59.42%
2	65.22%	52.17%	55.07%
3	46.38%	59.42%	55.07%
4	68.12%	69.57%	62.32%
5	66.67%	68.12%	66.67%
6	72.46%	66.67%	72.46%
7	65.22%	65.22%	62.32%
Average	62.32%	63.15%	61.90%

Table 18. The accuracy of C4.5 for sample sets of size 80% of the original data set with minor class over-sampling rate of 200%, 220%, 240%

Sample set	Over-samp. 200%	Over-samp 220%	Over-samp. 240%
1	56.52%	55.07%	62.32%
2	52.17%	57.91%	59.42%
3	62.32%	59.42%	66.67%
4	59.42%	65.22%	66.67%
5	69.57%	69.57%	66.67%
6	69.57%	73.91%	69.57%
7	69.57%	69.57%	68.12%
Average	62.73%	64.38%	65.63%

Table 19. The accuracy of C4.5 for sample sets of size 80% of the original data set with minor class over-sampling rate of 260%, 280%, 300%

Sample set	Over-samp. 260%	Over-samp 280%	Over-samp. 300%
1	60.87%	57.97%	56.52%
2	56.52%	59.42%	57.97%
3	63.77%	65.22%	63.77%
4	71.01%	66.67%	65.32%
5	69.57%	65.22%	65.22%
6	76.81%	73.91%	75.36%
7	71.01%	69.57%	69.57%
Average	67.08%	65.43%	64.82%

The following table 20 shows the accuracy of decision trees generated by CART for sample sets of size 80% of the original data set. The rest 20% were used for testing.

Table 20. The accuracy of CART for sample sets of size 80% of the original data set

Sample set	Accuracy
1	65.22%
2	69.57%
3	72.46%
4	68.12%
5	69.57%
6	72.46%
7	71.01%
Average	69.77%

From table 21 to table 25 shows the accuracy for decision tree algorithm CART with minor class over-sampling.

Table 21. The accuracy of CART for sample sets of size 80% of the original data set with minor class over-sampling rate of 20%, 40%, 60%

Sample set	Over-samp. 20%	Over-samp 40%	Over-samp. 60%
1	66.67%	68.12%	65.22%
2	63.77%	63.77%	47.87%
3	72.46%	72.46%	59.49%
4	68.12%	65.22%	65.22%
5	63.77%	68.12%	63.77%
6	79.71%	**81.16**%	62.32%
7	72.46%	71.01%	71.01%
Average	69.57%	69.98%	62.13%

Table 22. The accuracy of CART for sample sets of size 80% of the original data set with minor class over-sampling rate of 80%, 100%, 120%

Sample set	Over-samp. 80%	Over-samp 100%	Over-samp. 120%
1	65.22%	68.12%	57.97%
2	68.12%	60.87%	59.42%
3	59.42%	60.87%	59.42%
4	68.12%	72.46%	56.52%
5	72.46%	71.01%	66.67%
6	69.57%	65.22%	65.22%
7	71.01%%	69.57%	62.32%
Average	67.15%	66.87%	61.08%

Table 23. The accuracy of CART for sample sets of size 80% of the original data set with minor class over-sampling rate of 140%, 160%, 180%

Sample set	Over-samp. 140%	Over-samp 160%	Over-samp. 180%
1	68.12%	55.07%	59.42%
2	59.42%	56.52%	62.32%
3	59.42%	59.42%	65.22%
4	60.87%	73.91%	71.01%
5	68.12%	68.12%	69.57%
6	60.87%	65.22%	62.32%
7	69.57%	63.77%	65.22%
Average	63.77%	63.15%	65.01%

Table 24. The accuracy of CART for sample sets of size 80% of the original data set with minor class over-sampling rate of 200%, 220%, 240%

Sample set	Over-samp. 200%	Over-samp 220%	Over-samp. 240%
1	57.97%	68.12%	62.32%
2	60.87%	56.52%	59.42%
3	65.22%	60.87%	60.87%
4	59.42%	66.67%	66.67%
5	68.12%	72.46%	69.57%
6	60.87%	65.22%	65.22%
7	56.52%	73.91%	63.76%
Average	61.28%	66.25%	63.98%

Table 25. The accuracy of CART for sample sets of size 80% of the original data set with minor class over-sampling rate of 260%, 280%, 300%

Sample set	Over-samp. 260%	Over-samp 280%	Over-samp. 300%
1	65.22%	65.22%	60.87%
2	62.32%	60.87%	59.42%
3	62.32%	57.97%	57.97%
4	65.22%	55.07%	57.97%
5	66.67%	69.57%	68.12%
6	75.36%	63.77%	68.12%
7	55.07%	57.97%	56.52%
Average	64.60%	61.49%	61.28%

The graph in fig. 2 shows the change of average accuracy in the previous four different experiments. In the graph X axis represents the percentage of over-sampling for class 1, and Y axis represents average accuracy in percent.

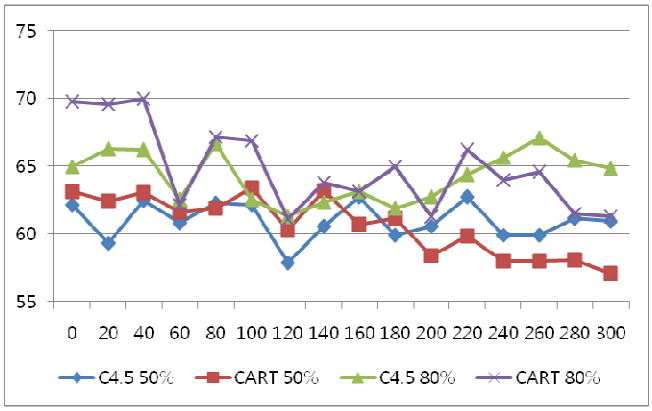

Fig. 2. The change of accuracy in four different experiments

As we can see from the graph, CART using sample sets of 80% of original data set generated the best results. The following table 26 to 29 shows the best accuracy for each experiment.

Table 26. The best accuracy of C4.5 with 50% sampling from the original data set

Sample set number	Over-sampling rate	Accuracy
1	160%	71.10%

Table 27. The best accuracy of CART with 50% sampling from the original data set

Sample set number	Over-sampling rate	Accuracy
7	40%	69.94%

Table 28. The best accuracy of C4.5 with 80% sampling from the original data set

Sample set number	Over-sampling rate	Accuracy
6	40%	76.81%
6	260%	76.81%

Table 29. The best accuracy of CART with 80% sampling from the original data set

Sample set number	Over-sampling rate	Accuracy
6	40%	**81.16%**

As we can see in table 29, sample set 6 with oversampling rate 40% generated the best result with CART in the experiment.

V. CONCLUSIONS

Liver is the largest internal organ in the human body, and it is known that the organ is responsible for more than one hundred functions of human body. The complexity of this organ makes it easily affected by disease of disorder. So diagnosing liver disorder disease is a high interest to researchers of data miners, and decision trees can be a good data mining tool to diagnose the disease. Decision trees have been considered one of good data mining tools with respect to understandability and transformability. But, weakness of decision trees arises due to the fact that their branching criteria give higher priority to classes of purity. BUPA liver disorder data set that is our interest for data mining is relatively small and has high error rate so that decision trees in conventional means may not generate good results due to the property.

In order to overcome the problem of disdaining classes of higher error rates in decision tree generation algorithms, we used over-sampling technique for the data instances of the class of higher error rates. Experiments with two representative decision tree algorithms, C4.5 and CART, showed very good results. Therefore, when we use decision trees, we may recommend over-sampling for the data set for better results in accuracy.

REFERENCES

[1] V. Podgorelec, P. Kokol, B. Stiglic, and I. Rozman, "Decision trees: an overview and their use in medicine," *Journal of Medical Systems,* Kluwer Academic/Plenum Press, vol. 26, no. 5, 2002, pp. 445-463.

[2] Y.C. Lin, "Design and Implementation of an Ontology-Based Psychiatric Disorder Detection System," *WSEAS Transactions on Information Sciences and Applications*, issue 1, vol. 7, January 2010, pp. 56-69.

[3] D. Chiang, W. Chen, Y. Wang, and L. Hwang, "Rules Generation From the Decision Trees," *Journal of Information Science and Engineering*, vol. 17, 2001, pp. 325-339.

[4] T. Tamai and M. Fujita, "Development of an expert system for credit card application assessment," *International Journal of Computer Applications in Technology,* vol. 2, no. 4, 1989, pp. 1-7.

[5] Y. Hui, Z. Longqun, and L. Xianwen, "Classification of Wetland from TM imageries based on Decision Tree", *WSEAS Transactions on Information Science and Applications*, issue 7, vol. 6, July 2009, pp. 1155-1164.

[6] S. Segrera and M.N. Moreno, "An Experimental Comparative Study of Web Mining Methods for Recommender Systems," in *Proceedings of the 6th WSEAS International Conference on Distance Learning and Web Engineering*, Lisbon, Portugal, September 22-24, 2006, pp. 56-61.

[7] V. Podgorelec, "Improved Mining of Software Complexity Data on Evolutionary Filtered Training Sets," *WSEAS Transactions on Information Science and Applications*, issue 11, vol. 6, November 2009, pp. 1751-1760.

[8] C. Huang, Y. Lin, and C. Lin, "Implementation of classifiers for choosing insurance policy using decision trees: a case study," *WSEAS Transactions on Computers*, issue 10, vol. 7, October 2008, pp. 1679-1689.

[9] P. Tryfos, *Sampling for Applied Research: Text and Cases*, Willy, 1996.

[10] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 2006.

[11] http://www.ehow.com/about_5048281_liver-disorders-diseases.html

[12] R. Ribeiro, R. Marinho, J. Velosa, F. Ramalho, and J.M. Sanches, "Chronic liver disease staging classification based on ultrasound, clinical and laboratorial data," in *Proceedings of 2011 IEEE International Symposium on Biomedical Imaging from Nano to Macro,* 2011, pp. 707-710

[13] R. Lin, "An intelligent model for liver disease diagnosis," *Artificial Intelligence in Medicine*, vol. 47, issue 1, 2009, pp.53-62

[14] S. Murthy and S. Salzberg, "Decision Tree Induction: How Effective is the Greedy Heuristic", in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pp. 222-227, 1995.

[15] L. Rokach and O. Maimon, *Data Mining with Decision Trees: Theory and Applications*, World Scientific Publishing Company, 2008.

[16] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, Inc., 1993.

[17] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth International Group, 1984.

[18] M.A. Mazuro, P.A. Habas, J.M. Zurada, J.Y. Lo, J.A. Baker, and G.D. Tourassi, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance.," *Neural Networks*, vol. 21, Issues 2-3, 2008, pp. 427-436.

[19] C. Lee, C. Tsai, and C. Chen, "A Hierarchical Shrinking Decision Tree for Imbalanced Datasets, " in *Proceedings of the 5th WSEAS Int.*

Conference on Data Networks, Communications & Computers, Bucharest, Romania, October 16-17, 2006, pp. 178-183.

[20] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research*, vol. 16, 2002, pp. 341-378.

[21] M. Neshat, M. Yaghobi, and M. Naghibi, "Designing expert system of liver disorders by using neural network and comparing it with parametric and nonparametric system," in *Proceedings of 5th International Multi-Conference on Systems, Signals and Devices*, 2008, pp. 1-6.

[22] M. Neshat, M. Yaghobi, M. Naghibi, and A. Esmaelzadel, "Fuzzy Expert System Design for Diagnosis of Liver Disorders," in 2*008 International Symposium on Knowledge Acquition and Modeling*, 2008, pp. 252-256.

[23] H. Kahramanli, N. Allahverdi, "Mining Classification Rules for Liver Disorders," *International Journal of Mathematics and Computers in Simulation*, vol. 3, issue 1, 2009, pp. 9-19.

[24] B. V. Ramana, M.S.P. Babu, and N.B. Venkateswarlu, "A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis," *International Journal of Database Management Systems*, vol. 3, no. 2, 2011, pp. 101-114.

[25] C. Dendek and J. Mań dziuk, "Improving Performance of a Binary Classifier by Training Set Selection," in *Proceedings of 18th International Conference on Artificial Neural Networks*, LNCS 5163, 2008, pp. 128-135.

[26] http://archive.ics.uci.edu/ml/datasets/Liver+Disorders

[27] A. Frank and A. Suncion, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Sciences, 2010

Hyontai Sug received the B.S. degree in Computer Science and Statistics from Busan National University, Busan, Korea, in 1983, the M.S. degree in Computer Science from Hankuk University of Foreign Studies, Seoul, Korea, in 1986, and the Ph.D. degree in Computer and Information Science & Engineering from University of Florida, Gainesville, FL, in 1998. He is an associate professor of the Division of Computer and Information Engineering of Dongseo University, Busan, Korea from 2001. From 1999 to 2001, he was a full time lecturer of Pusan University of Foreign Studies, Busan, Korea. He was a researcher of Agency for Defense Development, Korea from 1986 to 1992. His areas of research include data mining and database applications.