An Effective Sampling Method for Decision Trees Considering Comprehensibility and Accuracy

HYONTAI SUG
Division of Computer and Information Engineering
Dongseo University
Busan, 617-716
REPUBLIC OF KOREA
hyontai@yahoo.com http://kowon.dongseo.ac.kr/~sht

Abstract: - Because the target domain of data mining using decision trees usually contains a lot of data, sampling is needed. But selecting proper samples for a given decision tree algorithm is not easy, because each decision tree algorithm has its own property in generating trees and selecting appropriate samples that represent given target data set well is difficult. As the size of samples grows, the size of generated decision trees grows with some improvement in error rates. But we cannot use larger and larger samples, because it's not easy to understand large decision trees and data overfitting problem can happen. This paper suggests a progressive approach in determining a proper sample size to generate good decision trees with respect to generated tree size and accuracy. Experiments with two representative decision tree algorithms, CART and C4.5 show very promising results.

Key-Words: - decision trees, proper sample size determination

1 Introduction

For the tasks of data mining decision trees have been very widely used in a variety of fields, because their structures are easy to understand and they are good in prediction tasks [1, 2, 3, 4]. So finding decision trees with the smallest error rate as well as smaller size for a given data set has been a major concern for their success [5]. But even though decision trees are one of the most successful data mining methodologies, there are some weak points due to the fact that they are built based on greedy algorithms with limited target data sets. Because decision tree generation algorithms divide training data in each root of subtrees in the decision tree, as a decision tree is being built, each branch becomes to have less training examples as the result of the branching. Therefore, the reliability of lower branches becomes worse than upper branches due to the smaller size of training examples than the upper branches.

In addition, the greedy algorithms of decision trees assume that local optima are also global optima. In other words, when the algorithms determine root attribute for each subtree, the algorithms use some heuristic measures that calculate entropy or purity of each candidate root attribute decisively. In order to overcome the weak point of decision trees somewhat, pruning is performed based on some heuristic measure.

Moreover, because most target databases for data mining are very large, we need sampling process from the target databases. But the task of determining proper sample sizes is arbitrary and the found knowledge based on the random samples is prone to sampling errors or sampling bias. According to statistics a proper sample size for a feature is 30 or so [6]. For example, to determine the average height of people, we need to do random sampling for 30 people. But, in general, the target databases of data mining contain a lot of features, so if we do sampling like this, the sample size can become enormous. Therefore, we need an alternative strategy for sampling.

In the principle of Occam's razor [7, 8] simpler and smaller knowledge models are preferred to larger and more complex ones, because simpler knowledge models can cover more cases so that the predictability in the future cases becomes better. In this paper we suggest some clever way to do sampling that allows us to consider simpler decision trees.

In section 2, we provide the related work to our research, and in sections 3 we present our method. Experiments were run to see the effect of the method in section 4. Finally section 5 provides some conclusions.

2 Related Work

Because the problem of generating optimum decision trees is a NP-complete problem, decision tree algorithms resort to some greedy search methods so that generated decision trees are not optimum and some improvement may be possible. There have been a lot of efforts to build better decision trees so that branching or splitting measure is a major concern. For example, one of standard decision tree algorithm C4.5 [9] uses an entropy-based measure, and CART [10] uses a purity-based measure to split the branches. Because CART spends relatively large computing time for optimization, it is known that the algorithm generates smaller decision trees than other decision tree algorithms like C4.5. So many people prefer CART.

There have been also scalability related efforts to generate decision trees for large databases such as SLIQ [11], SPRINT [12], PUBLIC [13], and RainForest [14]. SLIQ saves some computing time especially when the database contains many continuous attributes by using a pre-sorting technique in tree-growth phase, and SPRINT is an improved version of SLIQ to solve the scalability problem by building trees with parallel processing algorithm. PUBLIC tries to save some computing time by integrating the tasks of pruning and generating branches together. However, these methods may generate very large decision trees for very large data sets so that the problem of comprehensibility and overfitting data in the generated decision trees may occur.

Generating right-sized decision trees requires a universal application of pruning [9, 10, 15, 16] so that overpruning was a natural consequence to generate comprehensively sized decision trees. In his Ph.D. dissertation, 'mega induction' for very large databases [15], J. Catlett relied on overpruning to obtain satisfactory decision trees. As a result of this overpruning, the generated tree may not have sufficient accuracy compared to near optimal, similar sized trees.

Sampling has been studied to find more accurate decision models. Several progressive sampling methods have been studied. In [17] arithmetic progressive sampling was applied to relatively small sized data sets in UCI machine learning repository [18]. They duplicated many data to make them big, and tested their sampling method to find more accurate naive Bayesian classifier. Due to the limitation of arithmetic sampling that increases sample size arithmetically the number of trials was limited. In [19] three progressive sampling schedules, arithmetic sampling, geometric sampling, and dynamic prgramming sampling, were analyzed for induction algorithms with polynomial time complexity. The researchers expected that geometric progressive sampling which incrases sample size geometrically is asymptotically optimal with respect to computing time and error rate among the three sampling methods. Experiments were done for the three methods with C4.5 and three relatively middle sized data sets. In [20] the researchers conducted experiments on the effect of sample size for six commercial data mining tools. But they did not reveal any information about the tools. They used two real world data sets. The first data set has 50,000 records and among them 40,000 records were used for trainng and 10,000 records were used for testing. The second data set has 1.5 million records, and 1.45 million records were used for training and 50,000 records were used for testing. The sample sizes were increased geometrically from the initial sample size of 500 to the final sample size of 32,000. So, total of six sample sizes were tested. They found similar results like [19]. But some tools showed nonmonotonic increases in accuracy as the sample size was increased.

3 The Method

3.1 Sampling methods

3.1.1 Arithmetic sampling

Arithmetic sampling is a progressive sampling method. Sample size is increased arithmetically so that sample sizes are in arithmetical progression. We can define sample size for sample i in arithmetic sampling with the following equation:

$$Si = S_0 + i \times C \qquad (1)$$

Here, S_0 is the initial sample size and C is a constant. So, we can have an arithmetical progression of samples in size, S_0, $S_0 + C$, $S_0 + 2C$, $S_0 + 3C$, and so on. For example, if S_0 = 1,000 and C = 100, then S_1 = 1,100, S_2 = 1,200, and so on.

So, if we use arithmetic sampling with some proper C value, we can trace the bahavior of induction algorithms throughly. But this property may become a drawback of the method. We may have to do a lot of sampling so that we need a lot of computing time, because the increase rate in sample size is small. For example, let's assume we have 1,000,000 records in a data set, and we start from 10,000 records as an initial

sample size and the constant C value is 500. We have to do sampling 9,800 times to reach to the half of the target data set. Because most target data sets for data mining contain lots of data, it is highly possible that arithmetic sampling alone cannot be used efficiently.

3.1.2 Geometric sampling

Geometric sampling is also a progressive sampling method. Sample size is increased geometrically so that sample sizes are in geometrical progression. We can define sample size for sample i in geometric sampling with the following equation:

$$Si = S_0 \times C^i \qquad (2)$$

Here, S_0 is the initial sample size and C is a constant. So, we can have a geometrical progression of samples in size, S_0, $S_0 \cdot C$, $S_0 \cdot C^2$, $S_0 \cdot C^3$, and so on. For example, if S_0 = 1,000 and C = 2, then S_1 = 2,000, S_2 = 4,000, and so on. As we can see from the example, if we use geometric sampling, very soon we can see very big sample size. So, the target data set may be exhausted within a few rounds.

As an example, let's assume that we have 1,000,000 records in a data set as before, and we start from 1,000 records as an initial sample size and the constant C value is 2. So, sample size goes like 1,000, 2,000, 4,000, 8,000, 16,000, 32,000, 64,000, 128,000, 256,000, 512,000. It takes only 10 rouns to reach to the half of the target data set.

Another noticeable one in geometric sampling is that the sample size values are very sparse at the later stage of the sampling. So, geometric sampling can be a good sampling strategy, if used induction algorithms have the tendency of monotonic increase in classification accuracy as well as the complexity of knowledge model. Please look at Fig. 1 that depicts learning curve for some induction algorithm A. Because there is no sudden drop in prediction accuracy or complexity of knowledge model, sparseness in sample sizes will not cause any problem. An example of induction algorithm with this property is C4.5.

But let's assume that we have a learning curve that have some sudden drops with very small improvements in accuracy as the training size grows. Because geometric sampling method has very sparse sampling interval with respect to sample size at the later stage of the sampling schedule, we might miss the points.

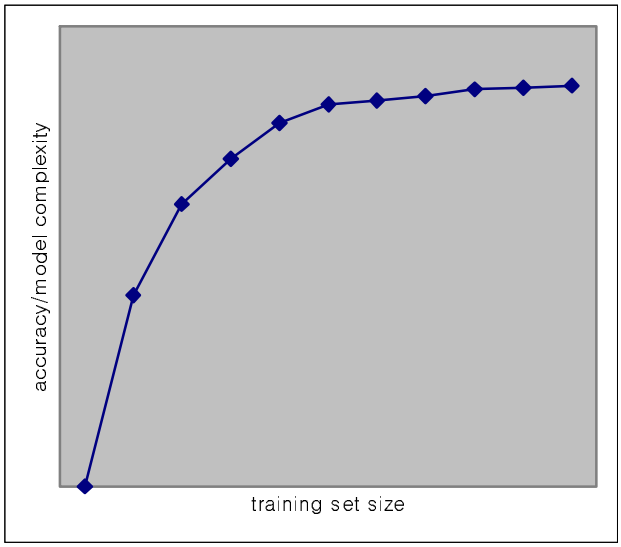

Fig. 1 Learning curve for some induction algorithm A

Please look at Fig. 2 that depicts learning curve for some induction algorithm B. Because there is a sudden drop in complexity of knowledge model, sparseness in sample sizes may not detect the point. An example of induction algorithm having this property is CART.

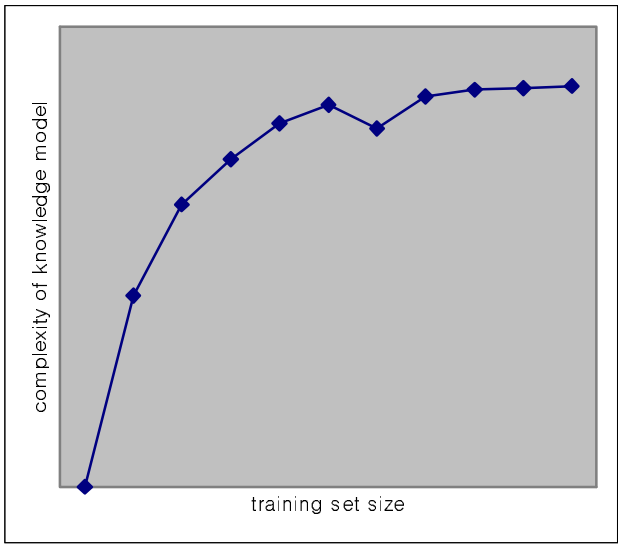

Fig. 2 Learning curve in model complexity for some induction algorithm B

3.1.3 Occam's razor

According to Domingos Occam's original razor can be defined in two forms for data mining domain [21]. The first razor prefers a simpler knowledge model on the condition that the two knowledge models have the

same error rates for unseen cases. The second razor prefers a simpler knowledge model on the condition that the two knowledge models have the same error rates for training examples.

Because training examples cover only some part of data space, it is highly possible that there are many unseen cases. So, preferring a simpler knowledge model that has the same error rate with a more complex knowledge model is not recommended. For example, if we use 10-fold cross validation method to train a knowledge model with relatively small sized target data set, choosing a simpler knowledge model among several candidates does not guarantee the smallest error rate in the future than choosing a more complex knowledge model, because 10-fold cross validation method has the tendency of Occam's second razor. So, in order to be close to Occam's first razor we should ensure a large test set data as long as it is possible.

3.1.4 Suggested method

Because most target data data sets for data mining contain lots of data, it is highly possible that arithmetic sampling alone cannot be used efficiently.

Because we know that overfitted decision trees do not perform well in prediction tasks, we should give appropriate parameter values for pruning [22] and avoid large decision trees if possible. And, moreover, because the size of decision trees has the tendency of dependency on the size of training data, it is important to do random sampling with appropriate sample size. But, because we have only limited number of data and the data should be divided into two parts, training and testing, it is not easy to determine an appropriate size of samples that is the best for the target data set. So, we resort to repeated sampling with various sizes to find the best one. We do the sampling until the sample size is less than the half of the target data set, because we assume that we have some large target data set and we want to have enough test data also. The following is a brief description of the procedure of the method.

INPUT: a data set for data mining,
k: the number of random sampling for each sample size,
s: initial sample size.
OUTPUT: a proper sample size s.
Do while s < | target data set | / 2
Do for i = 1 **to** k /* generate k decision trees for each loop*/
Do random sampling of size s;
Generate a decision tree;
End for;
a := the average (1-error rate) of decision trees;
A := A ∪ {a}; /* A: set of a values */
i := (the average (1-error rate) of decision trees of previous step) – (the average (1-error rate) of decision trees); /* average improvement rate */
I := I ∪ {i}; /* I: set of i values */
d := (maximum of (1- error rate) among the generated decision trees) - (minimum of (1-error rate) among the generated decision trees);
/* d stands for the fluctuation of (1-error rate) values in the generated decision trees */
D := D ∪ {d}; /* D: set of d values */
If s >= mid_limit **Then**
s := s + sample_size_increment; /* arithmetic sampling */
Else
s := s × 2; continue; /* while loop, geometric sampling */
End if
End while;

In the algorithm we use both of arithmetic sampling and geometric sampling to detect some critical sample size that can produce smaller decision trees in reasonable error rates. At initial state we use geometric sampling, because sample size is relatively small, and we switch to geometric sampling when we reach some appropriate point. So, we double the sample size until the size reaches some point, mid_limit, then we increment the sample size with some fixed value, because doubling the sample size can exhaust the training data soon.

Even though we do random sampling, because we may have some sampling bias and sampling errors, the generated tree may have a variety of tree sizes. So, in order to get rid of the effect of variety in tree size, we generate seven decision trees for each sample size. Then we average the sizes of the generated decision trees for each sample size, and this average decision tree size with improvement value and fluctuation value in accuracy is used to determine a proper sample size. By selecting a sample size that generates smaller decision trees in average with satisfactory error rates, we can have better decision tree in predictability in future cases.

4 Experimentation

Experiments were run using two data sets in UCI machine learning repository [18], 'census-income' and 'adult' to see the effect of the method.

In 'census-income' the number of instances for training is 199,523 in size of 99MB data file. Class probabilities for label -50000 and 50000+ are 93.8% and 6.2% respectively. The data set was selected because it is relatively very large and contains lots of values. The total number of attributes is 42. Among them eight attributes are continuous attributes. The values in continuous attributes are converted to nominal values with entropy-based discretization method, because the method showed the best result according to the experiments in [23].

In 'adult' data set the number of instances is 48,842. Class probabilities for label '>50K' is 24.78% and label '<=50K' is 75.22%. The total number of attributes is 15. Among them six attributes are continuous attributes. The data set was selected because it is relatively large and is a refined data set of 'census-income' so that we can check the performance of our sampling method more realistically.

We used CART and C4.5 to generate decision trees from various sample sizes, because the two decision tree algorithms are widely accepted to become de facto standards.

4.1 Experimentation of 'census-income' data set

The following table 1 shows average tree sizes and error rates depending on various sample sizes for CART. For each sample size seven random samples have been selected and seven decision trees have been generated for the experiment. The initial sample size for training is 2,000 and the size of samples is doubled as the while loop runs. The given mid_limit value for sample size is 16,000 and the sample size of 8,000 is increased after the mid_limit. The rest of data set after sampling is used for testing.

In the table, the fifth column, improvement(%), means the percentage of improvement in accuracy compared to the trees of previous sample size, and the last column represents the difference of maximum and minimum values of accuracy among the decision trees in the given sample size.

Table 1. Decision tree by CART with various sample sizes

Samp. Size	Tree size	Average Accuracy	Improve -ment(%)	Diff. of max & min of accuracy (%)
2,000	8	93.97	-	0.74
4,000	10	94.29	0.32	0.81
8,000	18	94.55	0.26	0.35
16,000	24	94.94	0.39	0.29
24,000	54	94.95	0.01	0.21
32,000	22	95.01	0.06	0.30
40,000	42	95.10	0.09	0.22
48,000	59	95.13	0.03	0.20
56,000	57	95.20	0.07	0.16
64,000	48	95.22	0.02	0.18

If we look at table 1, the last line has slightly better accuracy of 0.21% than that of the sixth line which has the sample size of 32,000. But we may perfer the sample size of 32,000 to the sample size of 64,000, because the size of the decision tree is almost half so that the trees from sample size of 64,000 have higher possibility of overfitting. Fig. 3 shows the trend of average tree size as the sample size grows, and Fig. 4 shows the trend of average accuracy as the sample size grows.

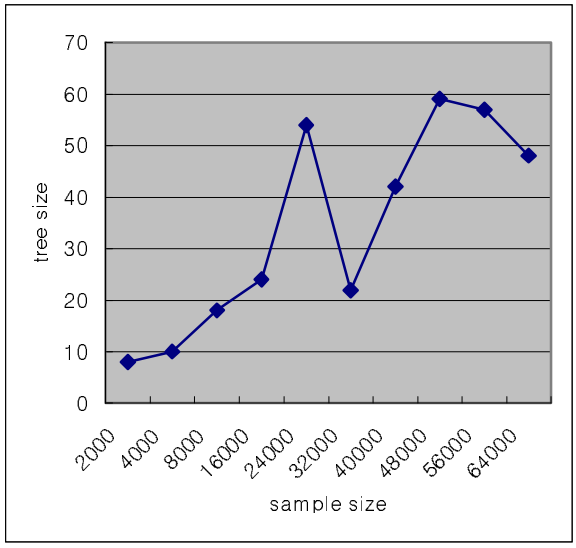

Fig. 3 The trend of average tree size as sample size grows (CART)

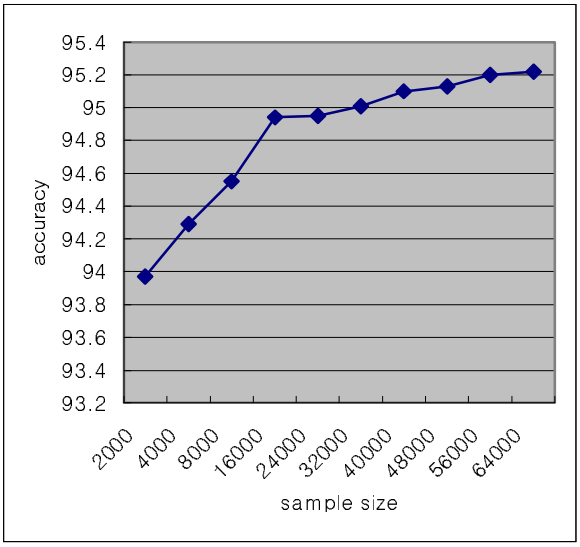

Fig. 4 The trend of average accuracy as sample size grows (CART)

Note also in table 1 that the difference of maximum and minimum values of accuracy among the decision trees in the sample size of 40,000 is 0.14% so that some good decision trees of the sample size are as good as the decision trees with the sample size of 64,000. Table 2 shows the details of each individual sample for the two sample sizes of 40,000 and 64,000. Accuracies in bold characters show the best and worst ones.

Table 2. Decision trees by CART with two different sample sizes

Sample size	40,000		64,000	
	Tree size	Accuracy (%)	Tree size	Accuracy (%)
	31	95.0139	61	**95.1219**
	33	95.1493	51	95.2266
	39	**95.2070**	33	95.2584
	23	95.0603	49	95.1956
	27	**94.9901**	31	95.2399
	99	95.1054	41	**95.2982**
	43	95.1531	69	95.2163
average	42	95.0970	48	95.2224

Experiments with C4.5 and the same sample sets were also conducted and resulta are summarized in table 3.

Table 3. Decision tree by C4.5 with various sample sizes

Samp. size	Tree size	Average accuracy (%)	Improve -ment(%)	Diff. of max & min of accuracy (%)
2,000	25	94.04	-	0.19
4,000	55	94.58	0.54	0.32
8,000	67	94.62	0.04	0.35
16,000	123	94.78	0.16	0.16
24,000	246	94.87	0.09	0.18
32,000	326	94.95	0.08	0.28
40,000	343	95.08	0.13	0.14
48,000	432	95.04	-0.04	0.28
56,000	467	95.08	0.04	0.17
64,000	490	95.14	0.06	0.16

If we look at table 3, the last line has slightly better accuracy of 0.06% than the seventh line which has the sample size of 40,000. But we may not choose the sample size of 64,000, because the size of the decision tree is almost 1.5 times larger so that the trees have higher possibility of overfitting. Fig. 5 shows the trend of average tree size as the sample size grows, and Fig. 6 shows the trend of average accuracy as the sample size grows.

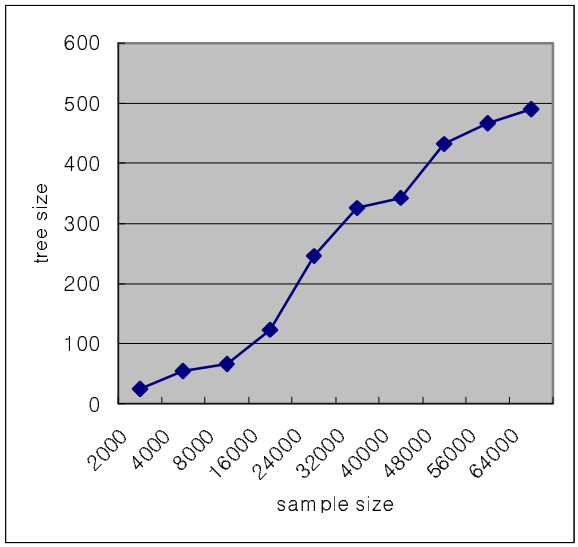

Fig. 5 The trend of average tree size as sample size grows (C4.5)

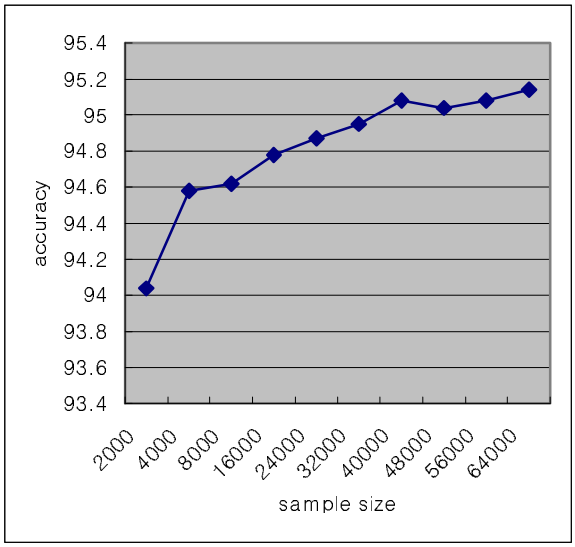

Fig. 6 The trend of average accuracy as sample size grows (C4.5)

Note also in table 3 that the difference of maximum and minimum values of accuracy among the decision trees in the sample size of 40,000 is 0.14% so that some good decision trees of the sample size are as good as the decision trees with the sample size of 64,000. Table 4 shows the details of each individual sample for the sample sizes of 40,000 and 64,000. Accuracies in bold characters show the best and worst ones.

Table 4. Decision trees by C4.5 with two different sample sizes

Sample size	40,000		64,000	
	Tree size	Accuracy (%)	Tree size	Accuracy (%)
	208	95.0973	344	95.1049
	387	95.0653	277	**95.0702**
	432	**95.1480**	564	95.1573
	423	95.1129	568	95.1632
	272	95.0985	542	95.1285
	387	**94.9713**	502	**95.2340**
	289	95.0571	635	95.1484
average	343	95.0786	490	95.1438

4.2 Experimentation of 'adult' data set

For 'adult' data set the initial sample size for training is 400 and the size of samples is doubled as the while loop runs. The given mid_limit value for sample size is 6,400 and the sample size of 3,200 is increased after the mid_limit. The rest of data set after sampling is used for testing. Table 5 summerizes the results of the experiment.

Table 5. Decision tree by CART with various sample sizes

Samp. Size	Tree size	Average Accuracy (%)	Improve -ment(%)	Diff. of max & min of accuracy (%)
400	12.7	82.3144	-	4.7727
800	13.6	83.1540	0.8396	3.0619
1,600	19.6	84.2112	1.0572	0.9018
3,200	35.3	84.7096	0.4984	1.5293
6,400	56.1	85.3451	0.6355	1.0010
9,600	55.9	85.8173	0.4722	0.4383
12,800	63.3	85.9145	0.0972	0.3325

If we look at table 5, the last line has slightly better accuracy of 0.1% than that of the sixth line which has the sample size of 9,600. But we may perfer the sample size of 9,600 to the sample size of 12,800, because the size of the decision tree is about 13% smaller so that the trees from sample size of 12,800 have higher possibility of overfitting. Fig. 8 shows the trend of average tree size as the sample size grows, and Fig. 9 shows the trend of average accuracy as the sample size grows.

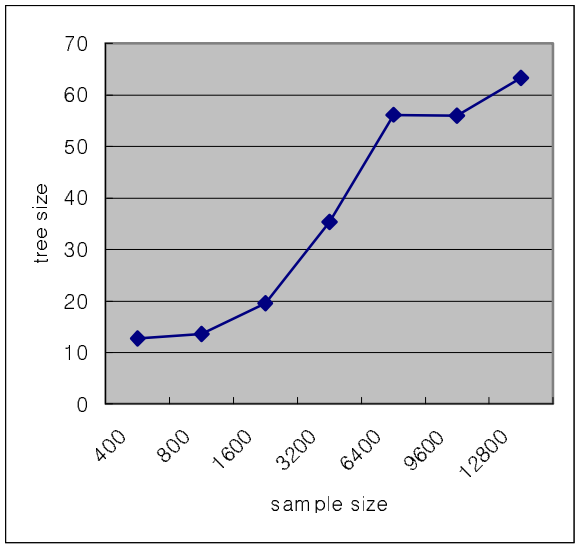

Fig. 7 The trend of tree size as sample size grows (CART)

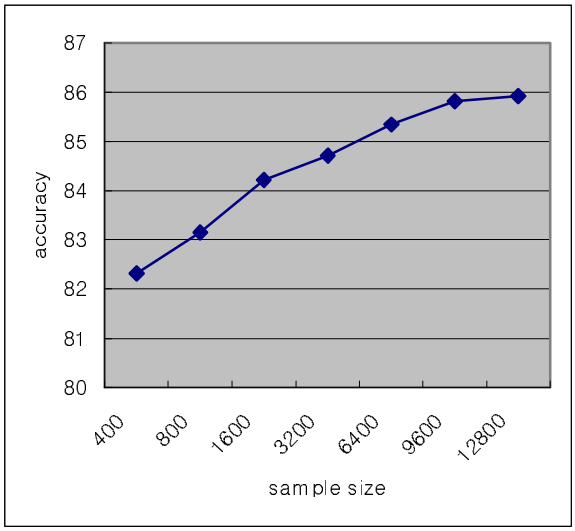

Fig. 8 The trend of accuracy as sample size grows (CART)

Note also in table 5 that the difference of maximum and minimum values of accuracy among the decision trees in the sample size of 9,600 is 0.4383% so that some good decision trees of the sample size are as good as the decision trees with the sample size of 12,800. Table 6 shows the details of each individual sample for the last two sample sizes. Accuracies in bold characters show the best and worst ones.

Table 6. Decision trees by CART with two different sample sizes

Sample size	9,600		12,800	
	Tree size	Accuracy (%)	Tree size	Accuracy (%)
	35	**85.5971**	43	85.8403
	69	85.7398	59	86.0831
	49	86.0226	71	85.8169
	53	85.7194	129	**86.0850**
	37	85.8188	65	85.8581
	59	85.7882	35	85.9556
	89	**86.0354**	41	**85.7625**
average	55.9	85.8173	63.3	85.9145

Experiments with C4.5 and the same sample sets were also conducted and resulta are summarized in table 7.

Table 7. Decision tree by C4.5 with various sample sizes

Samp. size	Tree size	Average accuracy (%)	Improve -ment(%)	Diff. of max & min of accuracy (%)
400	29.6	82.2810	-	4.1142
800	51.7	83.5558	1.2748	3.8987
1,600	83.1	83.8515	0.2957	2.2438
3,200	129.1	84.6739	0.8224	0.8413
6,400	187.6	85.1001	0.4262	0.6102
9,600	325.3	85.5443	0.4442	0.4179
12,800	424.3	85.5576	0.0133	0.5090

If we look at table 7, the last line has slightly better accuracy of 0.0133% than the sixth line which has the sample size of 9,600. But we may not choose the sample size of 12,800, because the size of the decision tree is almost 1.3 times larger so that the trees have higher possibility of overfitting. Fig. 9 shows the trend of average tree size as the sample size grows, and Fig. 10 shows the trend of average accuracy as the sample size grows.

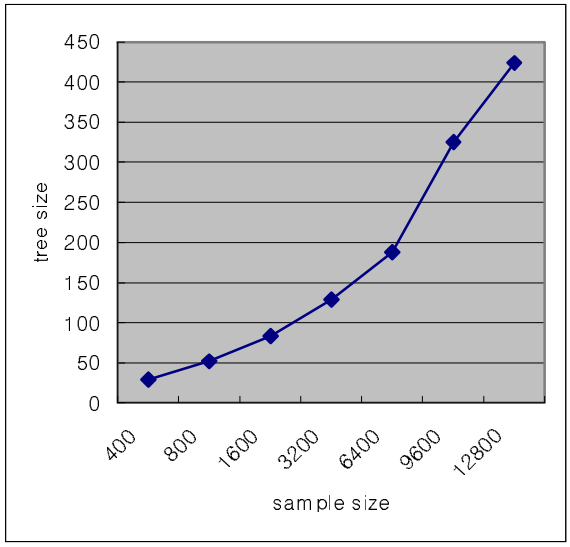

Fig. 9The trend of tree size as sample size grows (C4.5)

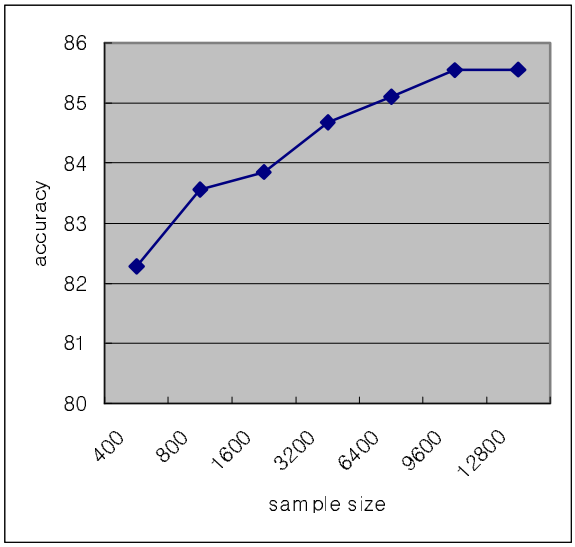

Fig. 10 The trend of accuracy as sample size grows (C4.5)

Note also in table 7 that the difference of maximum and minimum values of accuracy among the decision trees in the sample size of 9,600 is 0.4179% so that some good decision trees of the sample size are as good as the decision trees with the sample size of 12,800. Table 8 shows the details of each individual sample for the last two sample sizes. Accuracies in bold characters show the best and worst ones.

Table 8. Decision trees by C4.5 with two different sample sizes

Sample size	9,600		12,800	
	Tree size	Accuracy (%)	Tree size	Accuracy (%)
	290	**85.3448**	490	85.5582
	418	85.3601	353	85.7317
	264	**85.7627**	410	**85.2314**
	363	85.5232	400	85.6323
	480	85.6608	379	85.5013
	166	85.4646	400	85.5077
	296	85.6939	538	**85.7404**
average	325.3	85.5443	63.3	85.5576

5 Conclusions

Decision trees are widely accepted for data mining and machine learning tasks so that it is known that decision trees are one of the most successful data mining tools. But, decision trees may not always be the best data mining method due to the fact that they are built based on some greedy algorithms for limited data set. As a tree is being built, each branch starts having less number of training examples, so that the reliability of each lower branch becomes worse than the upper branches, therefore overfitting problem can happen. An overfitted trees may lead to unnecessary tests of attributes and may not represent knowledge model that are best for the domain.

Because the target data sets in data mining tasks contain a lot of data, random sampling has been considered a standard method to cope with large data sets that are common in data mining task. But, simple random sampling might not generate perfect samples that are good for the used data mining algorithms. Moreover, the task of determining a proper sample size is arbitrary so that the reliability of the generated data mining models may not be good enough to be trusted.

We propose a repeated sampling method with various sample sizes to decide the best size of random samples for decision tree algorithms. We consider the principle of Occam's razor that prefers simpler decision trees, if the candidate decision trees have similar performances. Experiments with a real world data sets and two representative decision tree algorithms, CART and C4.5 showed very promising results.

References:

[1] C. Huang, Y. Lin, C. Lin, Implementation of Classifiers for Choosing Insurance Policy using Decision Trees: a Case Study, *WSEAS Transactions on Computers*, Vol. 7, Issue 10, 2008, pp. 1679-1689.

[2] K. Miloslava, K. Jir I, J. Pavel, Application of Decision Trees in Problem of Air Quality Control in the Czech Republic locality, *WSEAS Transactions on Computers*, Vol. 7, Issue 10, 2008, pp. 1166-1175.

[3] D. Kaur, H. Pulugurta, Comparative Analysis of Fuzzy Decision Tree and Logistic Regression Methods for Pavement Treatment Prediction, *WSEAS Transactions on Computers*, Vol. 5, Issue 6, 2008, pp. 979-988.

[4] T. Wang, H. Lee, Constructing a Fuzzy Decision Tree by Integrating Fuzzy Sets and Entropy, *WSEAS Transactions on Computers*, Vol. 8, Issue 3, 2006, pp. 1547-1552.

[5] P. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 2006.
[6] W.G. Cochran, *Sampling Techniques*, 2nd ed., Wiley, 1997.
[7] G. Paul, Occam's Razor and a Non-syntactic Measure of Decision Tree Complexity, *AAAI 2004*, pp.962-963.
[8] P.M. Murphy, M.J. Pazzani, Exploring the Decision Forest: An Empirical Investigation of Occam's Razor in Decision Tree Induction, *Computational Learning Theory and Natural Learning Systems*, R. Greiner, T. Petsche, S.J. Hanson, ed., MIT press, 1997, pp.171-188.
[9] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, Inc., 1993.
[10] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth International Group, 1984.
[11] M. Mehta, R. Agrawal, and J. Rissanen, SLIQ : A Fast Scalable Classifier for Data Mining, *EDBT'96*, Avignon, France , 1996.
[12] J. Shafer, R. Agrawal, and M. Mehta., SPRINT : A Scalable Parallel Classifier for Data Mining, *Proc. 1996 Int. Conf. Very Large Data Bases*, Bombay, India, Sept. 1996, pp. 544-555.
[13] R. Rastogi, K. Shim, PUBLIC : A Decision Tree Classifier that Integrates Building and Pruning, *Data Mining and Knowledge Discovery*, vol. 4, no. 4, Kluwer International, 2002, pp. 315-344.
[14] J. Gehrke, R. Ramakrishnan, and V. Ganti, Rainforest: A Framework for Fast Decision Tree Construction of Large Datasets, *Proc. 1998 Int. Conf. Very Large Data Bases*, New York, NY, August 1998, pp. 416-427.
[15] J. Catlett, *Megainduction: Machine Learning on Very Large Databases*, PhD thesis, University of Sydney, Australia, 1991.
[16] SAS, *Decision Tree Modeling Course Notes*, SAS Publishing, 2002.
[17] G. John, P. Langley, Static Versus Dynamic Sampling for Data Mining, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996, pp. 367-370.
[18] D. Newman, *UCI KDD Archive* [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science, 2005.
[19] F. Provost, D. Jensen, T. Oates, Efficient Progressive Sampling, *Poceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, San Diego*, CA: ACM SIGKDD, 1999, pp. 23-32.
[20] J. Morgan, R. Daugherty, A. Hilchie, B. Carey, Sample Size and Modeling Accuracy of Decision Tree based Data Mining Tools, *Academy of Information and Management Sciences Journal*, Vol. 6, No. 2, 2003, pp. 77-92.
[21] P. Domingos, Occam's Two Razors: the Sharp and the Blunt, *Poceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1998, pp. 37-43.
[22] R.J. Lewis, An Introduction to Classification and Regression Tree (CART) Analysis, http:// www.saem.org/download/lewis1.pdf.
[23] H. Liu, F. Hussain, C.L. Tan, M. Dash, Discretization: An Enabling Technique, *Data Mining and Knowledge Discovery*, Vol. 6, 2002, pp. 393-423.