An Empirical Determination of Samples for Decision Trees

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 it is not known to determine a proper sample size for data mining tasks, the task of determining proper sample sizes for decision trees that are one of the best data mining algorithms is arbitrary, and 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 with limited target data set. This paper suggests an objective approach in determining a proper sample size to generate good decision trees with respect to generated tree size and error rates. 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, because they are good in prediction tasks and their structures are easy to understand. 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 [1]. But even though decision trees are one of the most successful data mining methodologies, there is some weak point due to the fact that they are built based on greedy algorithms with limited target data sets. Because of the greedy property of decision tree generation algorithms, as a decision tree is being built, each branch becomes to have less training examples as the result of branching. Therefore, the reliability of lower branches becomes worse than upper branches due to the smaller size of training examples in the lower branches than the upper branches. In order to overcome the weak point of decision trees somewhat, pruning is performed based on some 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 [2]. 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 [3, 4] 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 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 [5] uses an entropy-based measure, and CART [6] uses a purity-based measure for splitting 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 [7], SPRINT [8], PUBLIC [9], and RainForest [10]. 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 [3, 4, 11, 12] so that overpruning was a natural consequence to generate comprehensively sized decision trees. In his Ph.D. dissertation, ‘mega induction’ for very large databases [11], 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. Another simple method to use, when no explicit post processing for pruning is applied, is to stop the tree generation if the tree size becomes larger than some maximum allowable size. But, this method has similar problem with that of overpruning.

3 The Method

Because we know that overfitted decision trees do not perform well in prediction tasks, we should give appropriate parameter values for pruning [13] 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 \cup \{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 \cup \{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 \cup \{d\}; /* D: set of d values */
  If s >= mid_limit Then
    s := s + sample_size_increment;
  Else
    s := s \times 2; continue; /* while loop */
  End if
End while;

In the algorithm 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 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 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 error rate 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 a database in UCI machine learning repository [14] called 'census-income' to see the effect of the method. 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 database 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 [15].

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. The following Table 1 and 2 show average tree sizes and error rates depending on various sample sizes for CART and C4.5 respectively. 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 for each loop. The rest of data set after sampling is used for testing.

In the table, the fifth column, improvement(%), means the percentage of improvement in error rate compared to the trees of previous sample size, and the last column represents the difference of maximum and minimum values of 1 – error rate among the decision trees in the given sample size.

### Table 1. Decision tree by CART with various sample sizes

<table>
<thead>
<tr>
<th>Samp. Size</th>
<th>Tree size</th>
<th>Average of 1-error rate(%)</th>
<th>Improve -ment(%)</th>
<th>Diff. of max &amp; min of 1-error rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,000</td>
<td>8</td>
<td>93.97</td>
<td>-</td>
<td>0.74</td>
</tr>
<tr>
<td>4,000</td>
<td>10</td>
<td>94.29</td>
<td>0.32</td>
<td>0.81</td>
</tr>
<tr>
<td>8,000</td>
<td>18</td>
<td>94.55</td>
<td>0.26</td>
<td>0.35</td>
</tr>
<tr>
<td>16,000</td>
<td>24</td>
<td>94.94</td>
<td>0.39</td>
<td>0.29</td>
</tr>
</tbody>
</table>

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 prefer the sample size of 32,000 to the sample size of 64,000, because the size of the decision tree is almost double so that the trees from sample size of 64,000 have higher possibility of overfitting. Note also that the difference of maximum and minimum values of 1 – error rate 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. Decision tree by C4.5 with various sample sizes

<table>
<thead>
<tr>
<th>Samp. size</th>
<th>Tree size</th>
<th>Average of 1-error rate(%)</th>
<th>Improve -ment(%)</th>
<th>Diff. of max &amp; min of 1-error rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,000</td>
<td>25</td>
<td>94.04</td>
<td>-</td>
<td>0.19</td>
</tr>
<tr>
<td>4,000</td>
<td>55</td>
<td>94.58</td>
<td>0.54</td>
<td>0.32</td>
</tr>
<tr>
<td>8,000</td>
<td>67</td>
<td>94.62</td>
<td>0.04</td>
<td>0.35</td>
</tr>
<tr>
<td>16,000</td>
<td>123</td>
<td>94.78</td>
<td>0.16</td>
<td>0.16</td>
</tr>
<tr>
<td>24,000</td>
<td>246</td>
<td>94.87</td>
<td>0.09</td>
<td>0.18</td>
</tr>
<tr>
<td>32,000</td>
<td>326</td>
<td>94.95</td>
<td>0.08</td>
<td>0.28</td>
</tr>
<tr>
<td>40,000</td>
<td>343</td>
<td>95.08</td>
<td>0.13</td>
<td>0.14</td>
</tr>
<tr>
<td>48,000</td>
<td>432</td>
<td>95.04</td>
<td>-0.04</td>
<td>0.28</td>
</tr>
<tr>
<td>56,000</td>
<td>467</td>
<td>95.08</td>
<td>0.04</td>
<td>0.17</td>
</tr>
<tr>
<td>64,000</td>
<td>490</td>
<td>95.14</td>
<td>0.06</td>
<td>0.16</td>
</tr>
</tbody>
</table>

If we look at table 2, 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. Note also that the difference of maximum and minimum values of 1 – error rate among the decision trees in the sample size of 32,000 is 0.30% so that some good decision trees of the sample size are as good as the decision trees with the sample size of 64,000.

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 set and two representative decision tree algorithms, CART and C4.5 showed very promising result.

References: