THE PERFORMANCE OF LEARNING ALGORITHMS ON REDUCED DATA SETS

Kittisak Kerdprasop and Nittaya Kerdprasop

School of Computer Engineering
Suranaree University of Technology
Nakorn Ratchasima 30000, THAILAND
kerdpras@ccs.sut.ac.th, nittaya@ccs.sut.ac.th

Abstract

Knowledge discovery is the process of extracting useful and previously unknown information from the very large data set. But extracting knowledge from a large data set is computationally inefficient. Using a sample from the original data can speed up the data mining process, but this is only acceptable if it does not reduce the quality of the induced information. We thus investigate the behavior of learning algorithms on different sampling sizes to decide which sample is sufficiently similar to the original data. We observe the accuracy of the induced rules extracted from training samples of decreasing sizes and use these results to determine when a sample is sufficiently small, yet maintain the acceptable accuracy rate. We evaluate random and stratified sampling methods on data from the UCI repository with three learning algorithms.

Key Words: data mining, data reduction, sampling, accuracy

1. Introduction

Data mining (also known as knowledge discovery in databases, or KDD) is the process of applying specific learning algorithm to extract interesting and useful knowledge from data [1]. Typical data mining applications extract knowledge from databases ranging from small to moderate in size. When a data set is very large, mining process may take a very long time. Moreover, some mining algorithms may not be scalable on huge amounts of data. To handle large data sets, data reduction is one important step prior to applying the mining algorithms.

Data reduction can be achieved by reducing the number of instances and/or reducing dimensions of those instances. Our study focuses on instance reduction via the technique of sampling. Mining on reduced data set is obviously more efficient in terms of mining time than on the original data set. However, if the sample is too small, some useful knowledge may be overlooked or learning accuracy may be reduced. Our paper addresses the question of sufficient sample sizes that perform closely to the original data set, as well as the improved mining time.

We compare the performance of three different learning algorithms in terms of accuracy (or success rate) and learning time for various sampling sizes. Then conclude with the preferring samples. The rest of the paper is organized as follows. The next section describes various sampling methods. Section 3 discusses the algorithms chosen to run the data sets. Sections 4 and 5 explain the experimental setup and the results, respectively. Section 6 concludes the paper.

2. Sampling Methods

Sampling is used as a data reduction technique because it allows a large data set to be represented by a much smaller subset of the data. Basic methods of sampling commonly used are random sampling, systematic sampling, and stratified sampling [2].

Suppose that a large data set contains N instances. Random sampling selects n instances ($n < N$) at a random choice. The probability of drawing any instance in the data set is $1/N$, that is, all instances are equally likely. This is the case of random sampling without replacement. If the sampling is done with replacement, an instance has a chance to be drawn more than once.

The systematic sampling method draws n instances from the data set by their fixed stepping positions. This sampling method draws the first instance at a random position. Then iteratively draws subsequent instances at the next k position, when k is a stepping size.

Stratified sampling method first divides the data set into mutually disjoint subsets called *strata*. Then draws samples from each stratum independently by applying the simple random sampling technique. The three sampling methods are illustrated in Figure 2.1.

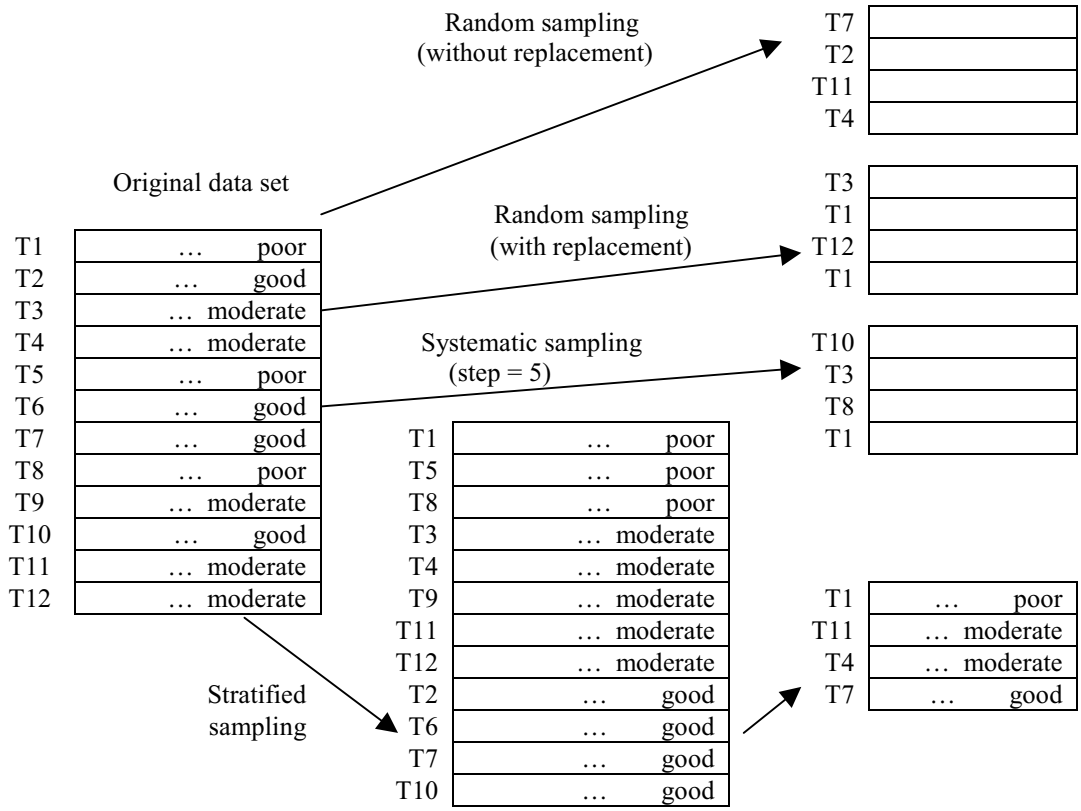


Figure 2.1 Different Sampling Methods to Draw 4 Samples.

3. Learning Algorithms

The three learning algorithms selected to perform the supervised learning task are OneR, naïve Bayes, and J48. OneR is a simple algorithm proposed by Holt [3]. OneR induces classification rules based on the value of a single attribute. We choose OneR to be a base algorithm for comparing the predictive accuracy with other sophisticated algorithms. It is shown that we can get reasonably accurate decision rules by simply looking at one attribute, as opposed to a more sophisticated top-down decision-tree induction algorithms such as C4.5 [4]. The average accuracy of OneR for the data sets tested by Holte [3] is just 5.7% lower than that of C4.5.

Naïve Bayes classification algorithm [5] is based on Bayes theorem of posterior probability. Given the instance, the algorithm computes conditional probabilities of the classes and picks the class with the highest posterior. Naïve-Bayes classification assumes that attributes are independent. The probabilities for nominal attributes are estimated by counts, while continuous attributes are estimated by assuming a normal distribution for each attribute and class. Unknown attributes are simply skipped. Experimental studies [5, 6] suggest that naïve Bayes tends to learn more rapidly than most induction algorithms. We, therefore, choose this algorithm to be a benchmark on comparing the rate of learning.

J48 algorithm [7] is an implementation of the C4.5 decision tree learner [4]. The algorithm uses the greedy technique to induce decision trees for classification. A decision-tree model is built by analyzing training data and the model is used to classify unseen data. An information-theoretic measure is used to select the attribute tested for each nonleaf node of the tree. Decision tree induction is an algorithm that normally learn a high accuracy set of rules. We thus choose the algorithm to compare with others on the basis of accuracy rate.

4. Experimental Methodology

The aim of our experiments is to study the performance of learning algorithms on the reduced data sets of various sizes. We choose the chess data set from the UCI Repository [8]. The data set is sampled using two different sampling methods: random sampling (without replacement) and stratified sampling. We skip the systematic method because in our preliminary experiments it gives a set of data that performs very close to that of the random method.

For each sampling method, a data set is drawn for six different sample sizes: 25%, 17%, 10%, 5%, 1% and 0.1% of the original data set. Then run the three learning algorithms on each sample three times and average the result. The learning algorithm is also run on the original data set (sampling size = 100%) to observe

the accuracy and the learning time. These two criteria will be used as a benchmark to compare against those obtained from the various samples.

The experiments are performed on the WEKA (Waikato Environment for Knowledge Analysis) system [9]. WEKA system is an open-source Java-based machine learning environment that provides tools and algorithms to be used as a data-mining workbench.

In our experiments, we partition the original data set into two mutually disjoint sets: a training set and a test set. The training set is used to train the learning algorithm, and the induced decision rules are tested on the test set. The test set contains 281 instances. Sampling for different sizes is done on the remaining 27,775 instances.

5. Results

Table 5.1 shows the results of running three different algorithms on the seven sampling sizes (100%, 25%, 17%, 10%, 5%, 1%, and 0.1%). Each sample is drawn using the normal random and the stratified random sampling methods. For each run the number of correctly classified rules is observed and reported as the accuracy of the learned model (shown in columns 4, 6, and 8). The learning time (or time to build model) is also investigated and displayed in columns 5, 7, and 9.

To clarify the comparison of accuracy and learning time, the bar graphs are shown in Figure 5.1. The upper graph shows the high accuracy of the decision-tree induction algorithm (J48). Despite the impressive accuracy rate, the accuracy curve is unstable on the reducing data sets, whereas the OneR and naïve Bayes show the stable accuracy on the data sets reduced down to 1% of the original data set. It turns out as we expect that the accuracy rate of OneR is the lowest among the three algorithms. Moreover, at the sampling sizes 5% and 1% the accuracy of naïve Bayes and J48 are insignificantly difference.

The lower graph in Figure 5.1 shows the improved learning time of the algorithms on the reducing data sets. Since the J48 algorithm consumes far more leaning time than the other two algorithms, we have to plot their logarithmic time values. The learning time of J48 drops drastically when we reduce the data set to the 25% sample size. It employs the learning time 85% lower than the time used by the original data set to trade with the 19% increase in the error rate. The OneR and naïve Bayes algorithms run in time almost linear with the number of instances; that is, the differences in learning time on reduced data sets at different sizes are not very significance.

Table 5.1 Experimental Result of Accuracy Estimation for each Reduced Data Set and the Learning Time.

Sampling Sizes	Number of Instances	Sampling Methods	Learning Algorithms					
			OneR		Naïve Bayes		J48	
			Accuracy	Time (seconds)	Accuracy	Time (seconds)	Accuracy	Time (seconds)
100%	27,775	No Sampling	24.62%	0.93	31.31%	1.1	69.75%	95.02
25%	6,944	Random	21.11%	0.2567	30.13%	0.35	50.41%	14.08
		Stratify	22.77%	0.2767	31.07%	0.257	51.00%	13.39
17%	4,630	Random	21.94%	0.2333	30.63%	0.127	46.02%	6.537
		Stratify	22.77%	0.13	31.19%	0.2	45.90%	6.717
10%	2,778	Random	21.94%	0.0767	29.89%	0.11	42.11%	2.977
		Stratify	21.94%	0.0567	29.06%	0.07	38.91%	3.08
5%	1,389	Random	22.77%	0.0533	27.99%	0.037	34.63%	0.717
		Stratify	21.94%	0.0533	29.77%	0.06	35.23%	0.753
1%	278	Random	20.043%	0.02	26.57%	0.037	25.26%	0.07
		Stratify	19.093%	0	27.01%	0	27.75%	0.113
0.1%	28	Random	9.4833%	0	19.45%	0	13.04%	0
		Stratify	13.633%	0	19.81%	0	14.11%	0

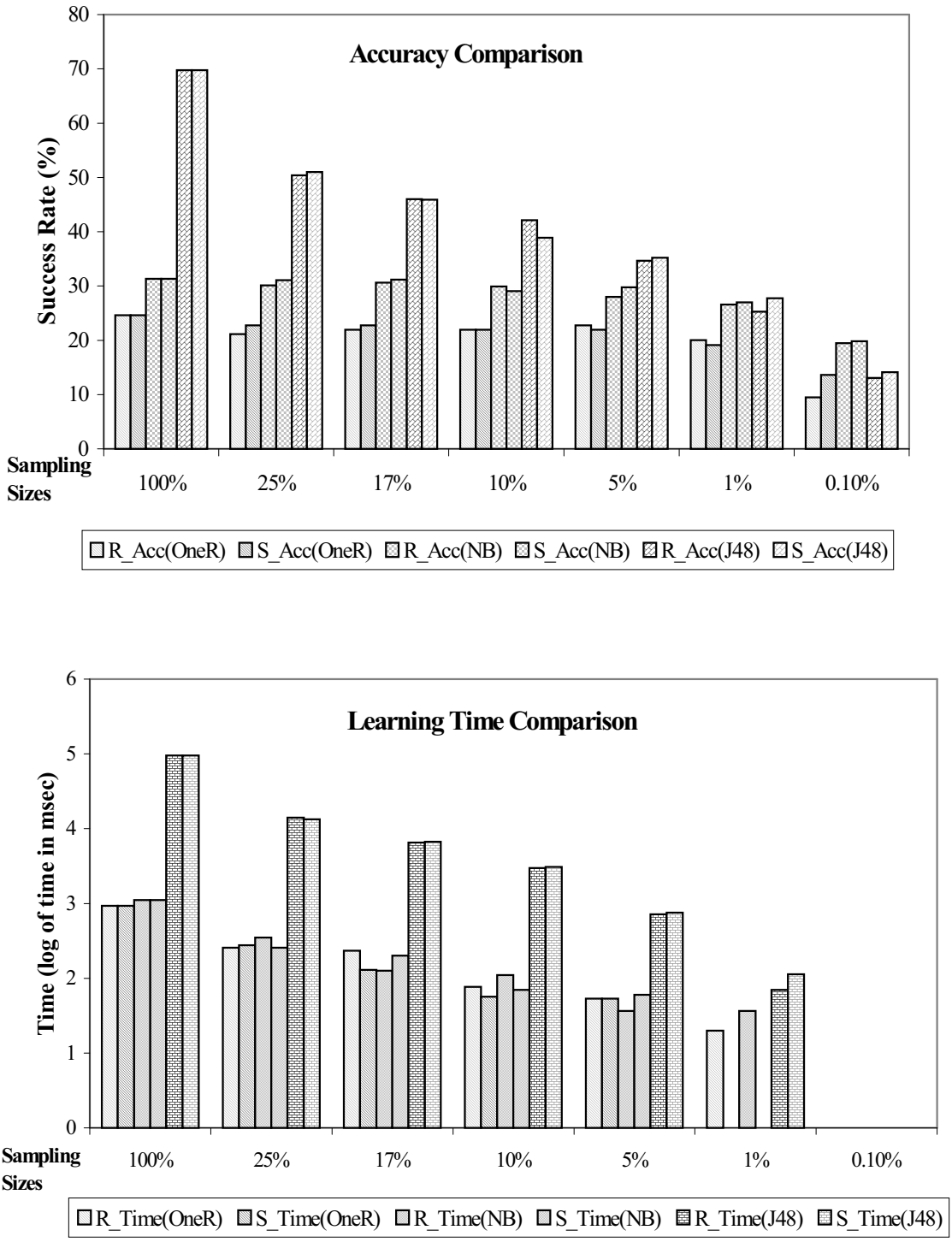
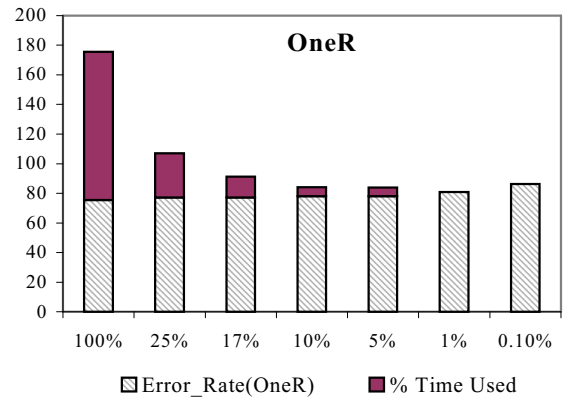
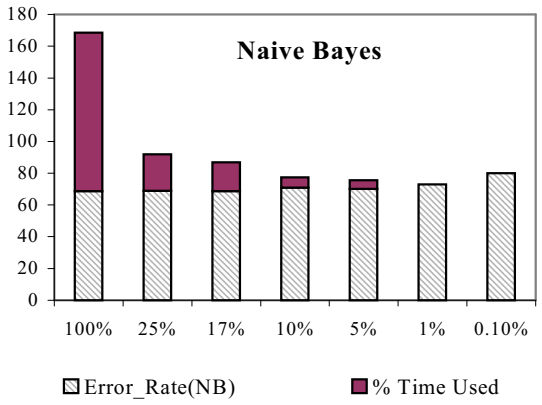


Figure 5.1 The Comparisons of Accuracy and Learning Time of Algorithms on each Sample Size. R_Acc is the accuracy (or success rate) on the random sampling (without replacement) data set. S_Acc is the accuracy (or success rate) on the stratified sampling data set. R_Time is the learning time on a data set obtained from random sampling. S_Time is the learning time on a stratified random data set. Learning time of all the 0.10% sampling data is 0 second.

Sampling	Error Rate	% Time Usage
100%	75.38	100
25%	77.23	29.75
17%	77.23	13.98
10%	78.06	6.09
5%	78.06	5.73
1%	80.91	0
0.10%	86.37	0



Sampling	Error Rate	% Time Usage
100%	68.69	100
25%	68.93	23.12
17%	68.81	18.02
10%	70.94	6.31
5%	70.23	5.41
1%	72.99	0
0.10%	80.19	0



Sampling	Error Rate	% Time Usage
100%	30.25	100
25%	49	14.09
17%	54.1	7.07
10%	61.09	3.24
5%	64.77	0.79
1%	72.25	0.12
0.10%	85.89	0

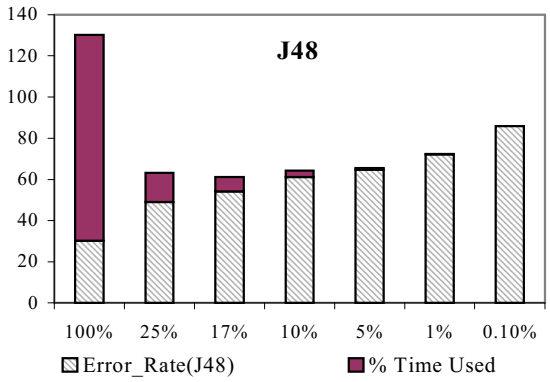


Figure 5.2 The Tradeoff in Decreasing Learning Time and Increasing Error Rate.
The varied sampling sizes are plotted on the horizontal axis. Vertical axis shows the percentage of error rate (of the model on predicting the unseen data) versus percentage of time usage on building model.

Figure 5.2 shows the accuracy-learning time tradeoff point to guide the decision of which sampling size should be the optimal choice. OneR and naïve Bayes have shown the characteristic of fast learning algorithms. They need samples around 1 to 10% to achieve the high accuracy. This results also agree with the studies of Langley et al. [5, 6] regarding the fast learning rate of naïve Bayes classifier. J48, on the other hand, needs almost 100% of the samples to reach the highest accuracy rate. However, in the situation that resources are limited and the data set is too large, the sampling sizes of 25% down to 10% should give the acceptable model in terms of accuracy.

6. Conclusion

As data sets grow to the point where the amounts are typically measured in the unit of gigabytes, mining data sets of this size is an arduous and impractical task. Using a sample from the data set can speed up the data mining process. But sampling involves a decision about a tradeoff in lower the accuracy to obtain the improved and practical running time of a data mining algorithm. We perform the experiments to explore the behavior of three different algorithms on various grains of data set. OneR and naïve Bayes algorithms show the almost stable accuracy rate. This result reflects the fast-learning property of OneR and naïve Bayes. This property, however, does not hold in the decision-tree induction algorithm.

We also suggest the range of sampling sizes for each type of learning algorithms. OneR and naïve Bayes require sample around 1 to 10% to reach their high accuracy rate. For the decision-tree induction algorithm, the sampling size of 10 to 25% should give the acceptable accurate model. Nevertheless, because of the diversity on the data domains, we encourage further study toward each data group.

References

[1] U. Fayyad, G. Piatetsky-Shapiro, & P. Smyth, From data mining to knowledge discovery in databases, *AI Magazine*, 1996, 37-54.

[2] K. Josien, G. Wang, T.W. Liao, E. Triantaphyllou, & M.C. Liu, An evaluation of sampling methods for data mining with fuzzy c-means, In Dan Braha (Ed.), *Data Mining for Design and Manufacturing*, Chapter 15 (Kluwer Academic, 2001) 351-365.

[3] R.C. Holt, Very simple classification rules perform well on most commonly used datasets, *Machine Learning, 11*, 1993, 63-90.

[4] J. R. Quinlan, *C4.5: Programs for machine learning* (Morgan Kaufmann, 1993).

[5] P. Langley, W. Iba, & K. Thompson, An analysis of Bayesian classifiers, *Proceedings of the 10th National Conference on Artificial Intelligence*, 1992, 223-228.

[6] P. Langley & S. Saga, Induction of selective Bayesian classifiers, *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, 1994, 399-406.

[7] I.H. Witten & E. Frank, *Data mining: Practical machine learning tools and techniques with Java implementations* (Morgan Kaufmann, 2000).

[8] C. L. Blake & C. J. Merz, UCI Repository of machine learning databases, *University of California, Irvine*, Department of Information and Computer Science, 1998. [http://www.ics.uci.edu/~mlearn/ML Repository.html].

[9] WEKA (Waikato Environment for Knowledge Analysis), University of Waikato, Department of Computer Science, New Zealand. [http://www.cs. waikato.ac.nz/~ml].