Progressive Sampling for Association Rules Based on Sampling Error Estimation

Kun-Ta Chuang, Ming-Syan Chen, and Wen-Chieh Yang

Graduate Institute of Communication Engineering, National Taiwan University,
Intelligent Business Technology Inc., Taipei, Taiwan, ROC
doug@arbor.ee.ntu.edu.tw
mschen@cc.ee.ntu.edu.tw
slavayang@gmail.com

Abstract. We explore in this paper a progressive sampling algorithm, called *Sampling Error Estimation* (*SEE*), which aims to identify an appropriate sample size for mining association rules. *SEE* has two advantages over previous works in the literature. First, *SEE* is highly efficient because an appropriate sample size can be determined without the need of executing association rules. Second, the identified sample size of *SEE* is very accurate, meaning that association rules can be highly efficiently executed on a sample of this size to obtain a sufficiently accurate result. This is attributed to the merit of *SEE* for being able to significantly reduce the influence of randomness by examining several samples with the same size in one database scan. As validated by experiments on various real data and synthetic data, *SEE* can achieve very prominent improvement in efficiency and also the resulting accuracy over previous works.

1 Introduction

As the growth of information explodes nowadays, reducing the computational cost of data mining tasks has emerged as an important issue. Specifically, the computational cost reduction for association rule mining is elaborated upon by the research community [6]. Among research efforts to improve the efficiency of mining association rules, sampling is an important technique due to its capability of reducing the amount of analyzed data [2][5][7].

However, using sampling will inevitably result in the generation of incorrect association rules, which are not valid with respect to the entire database. In such situations, how to identify an appropriate sample size is key to the success of the sampling technique. Progressive sampling is the well-known approach to determine the appropriate sample size in the literature. Progressive sampling methods are based on the observation that when the sample size exceeds a size N_s, the model accuracy λ obtained by mining on a sample will no longer be prominently increased. The sample size N_s can therefore be suggested as the appropriate sample size. In general, the sample size N_s can be identified from the "*model accuracy curve*", which is in essence the model accuracy versus the sample size

T.B. Ho, D. Cheung, and H. Liu (Eds.): PAKDD 2005, LNAI 3518, pp. 505–515, 2005.

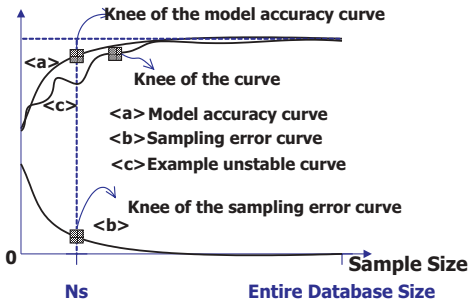


Fig. 1. The illustrated model accuracy curve and sampling error curve

[3]. Curve <a> in Figure 1 illustrates an example of the *model accuracy curve* versus the sample size. It can be observed that the model accuracy will stay in a plateau (the accuracy is no longer much improved) when the sample size exceeds N_s. Thus, the goal of progressive sampling algorithms is to efficiently estimate the *model accuracy curve* and then identify the point (N_s, λ_i) of this curve, where λ_i is the corresponding model accuracy for the sample size N_s.

However, previous progressive sampling algorithms for association rules mainly suffer from two problems. First, to measure the model accuracy of each sample size, previous progressive sampling algorithms have to resort to the execution of association rules either on samples [2] or on the entire database [3], which is, however, very costly. Second, for efficiency reasons, previous algorithms usually evaluate the model accuracy of a sample size by only executing association rules on a sample with this size, and the phenomenon of *randomness* [4] is thus not considered. *Randomness* refers to the phenomenon that mining on samples of the same size may obtain different results. In fact, *randomness* will affect the determination of the accuracy of obtained association rules for each sample size. Thus previous works will generate an unstable curve, like curve <c> in Figure 1, to estimate the *model accuracy curve*, and the resulted sample size may not be a proper choice.

To remedy these problems, we devise in this paper an innovative algorithm, referred to as *Sampling Error Estimation* (abbreviated as *SEE*), to identify the appropriate sample size without the need of executing association rule mining either on several samples or on the entire database, thus significantly improving the execution efficiency. The fundamental concept of algorithm *SEE* is to estimate the *model accuracy curve* by generating a curve of *sampling errors* versus the sample size. *Sampling errors* stem from the phenomenon that the proportion (also referred to as *support* in association rule research) of each item in the sample will deviate from its population proportion, and *sampling error* is indeed the reason for incurring incorrect association rules in the sample. In general, the smaller the *sampling error* of the sample, the higher accuracy can be obtained by mining on this sample. By calculating those *sampling errors* which will influence the obtained model accuracy, the shape of the *sampling error curve* can reflect the shape of the *model accuracy curve*. Curve <b> in Figure 1 illustrates

the *sampling error curve*. Moreover, *SEE* can greatly reduce the influence of *randomness* and correctly measure *sampling errors* of each sample size. Thus the *sampling error curve* can be employed to better estimate the *model accuracy curve*. This is attributed to the merit that *SEE* can calculate *sampling errors* of each sample size from a number of samples of this size in one database scan. As validated by experiments on various real data and synthetic data, algorithm *SEE* can achieve very prominent improvement in efficiency and also the resulting accuracy over previous works.

2 Sampling Errors for Association Rules

2.1 Descriptions of Sampling Errors

Sampling errors, referring to the phenomenon that the *support* of each item in the sample will deviate from its *support* in the entire data [4], will result in incorrectly identifying whether an item is a *frequent* item. The inference can be made from the following discussion. Suppose that we have 10,000 transactional records and 1,000 records of them contain the item {bread}. Hence the *support* of {bread} is 10%. In general, the phenomenon of *sampling errors* can be observed from the distribution of the item support in samples. When we generate a lot of samples of the same size, the sampling distribution, i.e., the *support* of one specified item among these samples, will approximately follow a *normal* distribution with *mean* equal to the support of this item in the entire database. Figure 2(a) shows an example of the distribution of the support of the item {bread} in samples. The *support* of {bread} in samples of the same size will follow a *normal* distribution with *mean* 10% because {bread} occurs 10% in the entire database. Suppose that the *minimum support* is specified as *8%* and thus the item type of {bread} is *frequent* because its support in the entire database is larger than *8%*. As a result, the shadow region in Figure 2(a) can represent the probability of incorrectly identifying {bread} as a *non-frequent* item. Moreover, the larger the shadow region, the larger probability we will obtain the incorrect item type.

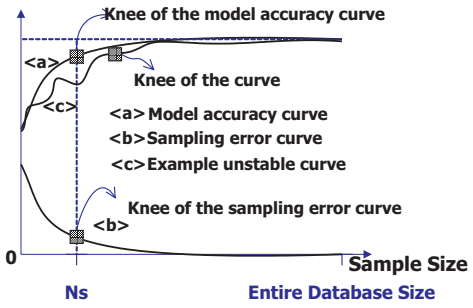


Fig. 2. The phenomenon of sampling errors

In practice, for one specified item, the probability of obtaining the incorrect item type relies on the sample size. More specifically, the variance of the item support in samples is inversely proportional to the square root of the sample size [4], and thus the influence of *sampling errors* will decrease as the sample size increases. As shown in Figure 2(b), the support of the item {bread} will have a smaller variance in a sample of 3,000 records than in a sample of 1,000 records. We can observe that the sample of 3,000 records will have a smaller error probability (the shadow region is smaller) of identifying {bread} as a *non-frequent* item. Thus for each item, the error probability of identifying its item type will decrease as the sample size increases.

In essence, mining association rules will generate a lot of itemsets. We know that as the sample size decreases, all itemsets will have the larger probability of incorrectly identifying their item types since *sampling errors* also increase as the sample size decreases. Therefore, if *sampling errors* cannot be significantly decreased when the sample size is larger than a sample size s_n, s_n can be suggested as the appropriate sample size for association rules.

In fact, such a size can be determined by generating a curve of *sampling errors* versus the sample size. In addition, the corresponding sample size at the convergence point of the curve will be suggested as the appropriate sample size. In the following, we present the method to measure *sampling errors* of each size.

2.2 Measurement of Sampling Errors

Since *sampling errors* stem from the difference between the proportion of each item in the sample and its population proportion, the *sampling error of item* i_n in the sample S, can be defined as:

Definition 1: (*Sampling error of item* i_n in the sample S)

$$SE(i_n, S) = |Sup(i_n, S) - Sup(i_n, D)|\,,$$

where $Sup(i_n, S)$ and $Sup(i_n, D)$ denote supports of item i_n in the sample S and in the entire database D, respectively.

Furthermore, for evaluating the accuracy of association rules, *sampling errors* of all itemsets are calculated because *sampling errors* of all itemsets will influence the result of association rules. Hence, *sampling errors* of the sample S can be defined as the root mean square sum of *sampling errors* of each occurred itemsets in the entire database.

In practice, using this measurement to evaluate the model accuracy of association rules will suffer from two problems. The first problem is that examining *sampling errors* of all itemsets is inefficient because the number of itemsets is huge. The second problem is that calculating *sampling errors* of several itemsets is indeed unnecessary. Consider the example shown in Figure 2(a). If the *minimum support* is specified as 20%, the error probability of identifying the item type of the item {bread} will be close to zero. Thus *sampling errors* of this item will not influence the accuracy of association rules.

To remedy the first problem, we employ the solution that only *sampling errors* of 1-itemsets will be calculated. Indeed, calculating *sampling errors* of 1-itemsets cannot completely response the model accuracy of association rules. However, it will be generally sufficient for the following reason. The property of association rules, i.e., *downward closure property* [6], shows that *sampling errors* of 1-itemsets will influence the accuracy of 2-itemsets, 3-itemset, etc. In other words, inaccurately identifying the item type of 1-itemsets will incur the error identification of item types of 2-itemsets, 3-itemsets, an so on. It can be expected that *sampling errors* of 1-itemsets will dominate the accuracy of *frequent* itemsets. Thus the model accuracy will not significantly increase when the sample size is larger than a size whose corresponding *sampling errors* of 1-itemsets will no longer significantly decrease. Therefore calculating *sampling errors* of 1-itemsets will be a good pilot to determine whether a sample size is sufficient for mining association rules.

Furthermore, to resolve the second problem, i.e., some itemsets will be irrelevant to the model accuracy if their supports are far away from the *minimum support*, we take into account the relationship between *sampling errors* and the *minimum support*. Suppose that p denotes the sample ratio, i.e. $\frac{|S|}{|D|}$, where $|S|$ and $|D|$ are sizes of S and D, respectively. Since sampling may cause changes of item supports, three distinct cases of the support change will be considered when the range of item supports (0˜1) is divided into 3 intervals, i.e., <A> $[0, p \cdot Min\_Sup]$, <B> $[p \cdot Min\_Sup, Min\_Sup]$, and <C> $[Min\_Sup, 1]$. Case (1) consists of those items whose supports change from <C> to <B> after sampling. Those items are identified as *frequent* in D but *non-frequent* in S. Case (2) consists of those items whose supports change from <B> to <C> after sampling. Those items are identified as *non-frequent* in D but *frequent* in S. In addition, case (3) consists of items identified as *frequent* in both D and S.

Note that the model accuracy is usually calculated as the combination of *recall* and *precision* [1]. In addition, *F-score* is a widely-used measurement which combines *recall* and *precision*. *F-score* of the result obtained by mining on S, is defined as $\frac{(\beta^2+1)\cdot P\cdot R}{\beta^2\cdot P+R}$, where P and R are *precision* and *recall*, respectively. β is a weighted value and it is usually set as 1 to fairly consider *precision* and *recall*. *Precision* $P(S)$ and *recall* $R(S)$ of *frequent* itemsets obtained in the sample S are defined as

$$P(S) = \frac{|FI_a(S)|}{|FI_a(S)|+|FI_b(S)|}; R(S) = \frac{|FI_a(S)|}{|FI_a(S)|+|FI_c(S)|},$$

where $FI_a(S)$ consists of itemsets which belong to case (3). $FI_b(S)$ consists of itemsets belonging to case (2) and $FI_c(S)$ consists of itemsets belonging to case (1). $|FI_a(S)|$, $|FI_b(S)|$ and $|FI_c(S)|$ are their corresponding sizes. The accuracy of the set of *frequent* itemsets obtained by mining on the sample S is thus formulated as $F(S) = \frac{2\cdot P(S)\cdot R(S)}{P(S)+R(S)}$. In essence, we can observe that only those itemsets belonging to $FI_a(S)$, $FI_b(S)$ and $FI_c(S)$ will affect the accuracy of *frequent* itemsets whereas other itemsets will not. Consequently, the *association rules-related sampling errors* of the sample S can be defined:

Definition 2: (*Association rules-related sampling errors of the sample* S). Suppose that there are M 1-itemsets, $\{a_1,a_2, ..., a_M\}$, belonging to cases (1)/(2)/(3). *Sampling errors* of the sample S which will influence the accuracy of association rules, can be defined as:

$$A\_SE(S) = \sqrt{\left(\sum_{k=1}^{M} SE(a_k, S)^2\right) / M}.$$

Furthermore, as mentioned previously, only mining on a sample is inadequate to evaluate the correct mining accuracy of the corresponding sample size due to the phenomenon of *randomness*. We use L samples of the same size to measure *sampling errors* of this size. Definition 3 follows.

Definition 3: (*Association rules-related sampling errors of sample size* $|S|$). Suppose that $A\_SE(S_k)$ denotes *association rules-related sampling errors* of the k^{th} sample of the sample size $|S|$, where $1 \leq k \leq L$. *Association rules-related sampling errors* of the sample size $|S|$ is defined as:

$$A\_SSE(|S|) = \left(\sum_{k=1}^{L} A\_SE(S_k)\right) / L.$$

Therefore, we calculate $A\_SSE(|S|)$ to measure *sampling errors* of each sample size which is given by a sampling schedule, and then the curve of $A\_SSE(|S|)$ versus the sample size can be used to estimate the curve of the model accuracy versus the sample size.

3 Algorithm SEE

3.1 Pseudocode of Algorithm SEE

Algorithm *SEE* will generate a curve of *sampling errors* versus the sample size immediately after one database scan, and then suggest the corresponding sample size at the convergence point of this curve as the appropriate sample size for association rules. To measure *sampling errors*, frequencies (or said *support count*) of each item in the entire database and in each sample will be required in *SEE*. To efficiently acquire such information, *SEE* is devised as a two phases algorithm: (1) the *database scan phase*, in which the database is scanned once and simultaneously the *frequency* of each item in the entire database and in each sample is stored. (2) The *convergence detection phase*, in which *sampling errors* of each sample size are calculated, and then the appropriate sample size is identified from the curve of *sampling errors* versus the sample size. The pseudo code of SEE is outlined below:

Algorithm SEE: SEE(D,L,$\mathbb{R}$,minSup)

$SEE[n][m]$: store sampling errors of the m^{th} sample of size s_n.

//**The database scan phase**

01. while has next transaction t_d
02. for every item i_k in t_d
03. $IList[i_k] \rightarrow freq_D++$;
04. for $n = 1$ to P
05. for $m = 1$ to L
06. if ($rv[n][m]$.next$<\frac{s_n}{|D|}$)
07. for every item i_k in t_d
08. $IList[i_k] \rightarrow freq_S[n][m]++$;

//**The convergence detection phase**

01. for $n = 1$ to P
02. for $m = 1$ to L
03. e=0;count_item=0;
04. for each item i_k in $IList$ {
05. if i_k belongs to case (1)/(2)/(3)
06. e+=$\left(\frac{IList[i_k] \rightarrow freq_S[n][m]}{s_n} - \frac{IList[i_k] \rightarrow freq_D}{|D|}\right)^2$;
07. count_item++;
08. $SEE[n][m] = \sqrt{\frac{e}{\mathbf{count\_item}}}$;
09. $A\_SSE(s_n) = \frac{\sum_{j=1}^{L} SEE[n][j]}{L}$;
10. if (s_n, $A\_SSE(s_n)$) is the convergence point
11. report s_n as the appropriate sample size; program terminated;

3.2 Complexity Analysis of Algorithm SEE

Time Complexity: Suppose that $|I|$ is the number of distinct items in the entire database D. The time complexity of *SEE* is $O(|D| \times P \times L + P \times L \times |I|)$, which is linear with respect to the entire database size $|D|$.

Space Complexity: The space complexity is $O(L \times P \times |I|)$. The major space requirement is used to store the frequency of each item in each sample, i.e., $IList[i_k] \rightarrow freq_S[n][m]$.

4 Experimental Results

We assess the quality of algorithm *SEE* in Windows XP professional platform with 512Mb memory and 1.7G P4-CPU. For comparison, the corresponding results of algorithm *RC-S* [2] are also evaluated. In the literature, *RC-S* can be deemed as the most efficient method to date. The utilized association rule mining algorithm is *Eclat* [6]. Furthermore, in all experiments, the user-specified parameter α of algorithm *RC-S* is set to one, which is the same as its default

value addressed in [2]. All necessary codes are implemented by Java and complied by Sun jdk1.4.

4.1 Methods for Comparison

To demonstrate that progressive sampling can improve the performance of mining association rules, in all experiments, the execution time of each algorithm will consist of two costs: (1) the time of executing progressive sampling algorithm to determine the appropriate sample size; (2) the time of executing association rules on a sample with the identified size. Moreover, since the scale of *sampling errors* is different from the model accuracy and the self-similarity, all curves shown in experimental results are normalized to [0,1] scale.

In our experiments, the curve "*Normalized Model Accuracy*" denotes the curve of the normalized *model accuracy* of frequent itemsets versus the sample size. Note that the *model accuracy* of a sample size is calculated as the average *F-Scores* from 20 runs with this size. In addition, the curve "*RC-S*" denotes the normalized *self-similarity curve* which is generated by algorithm *RC-S*. Note that *sampling errors* will decrease as the sample size increases, and the model accuracy will increase as the sample size increases. Thus the curve "*Inv_NSEE*" shows *the inverse sampling errors*, i.e., 1-*normalized* $A\_SSE(s_n)$, of each sample size s_n. In addition, algorithms *RC-S* and *SEE* all aim to estimate curves "*Normalized Model Accuracy*", and thus we can estimate the effectiveness of *SEE* and *RC-S* by observing the difference between "*Inv_NSEE*"/"*RC-S*" and "*Normalized Model Accuracy*". Note that the quantitative analysis of this difference can be defined as the root mean square error, $\sqrt{\frac{1}{P}\sum_{i=1}^{P}[\upsilon(s_i)-\varphi(s_i)]^2}$, where P is the number of distinct sample sizes in the sampling schedule, and $\upsilon(s_i)$ denotes the normalized model accuracy of the sample size s_i and $\varphi(s_i)$ denotes the normalized score of the sample size s_i in the curve "*Inv_NSEE*"/"*RC-S*". The root mean square error will be shown as the value *"Curve Error"* in each experiment.

Furthermore, we use an arithmetic sampling schedule with $\mathbb{R}$=$\{0.05 \times |D|, 0.1 \times |D|, ..., 0.95 \times |D|\}$. In addition, since curves "*Normalized Model Accuracy*", "*RC-S*", and "*Inv_NSEE*" are all monotonically increasing, the appropriate sample size identified in each curve will be the smallest sample size whose corresponding normalized score exceeds 0.8, meaning that we have up to 20% improvement when the sample size is larger than the identified size.

4.2 Experiments on Real Data and Synthetic Data

Experiments on the Parameter Sensitivity. In this experiment, we observe the parameter sensitivity of algorithm *SEE*, and the large real data set, POS is utilized. First, Figure 3(a) shows the sensitivity analysis of the parameter L, which is the number of samples used to evaluate the corresponding *sampling errors* of a sample size. In Figure 3(a), the y-axis represents the score distance of each sample size s_i, which is defined as $|\upsilon(s_i) - \varphi(s_i)|$, where $\upsilon(s_i)$ denotes the

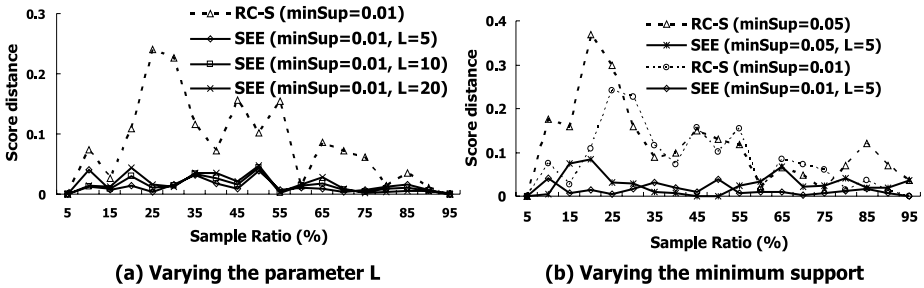


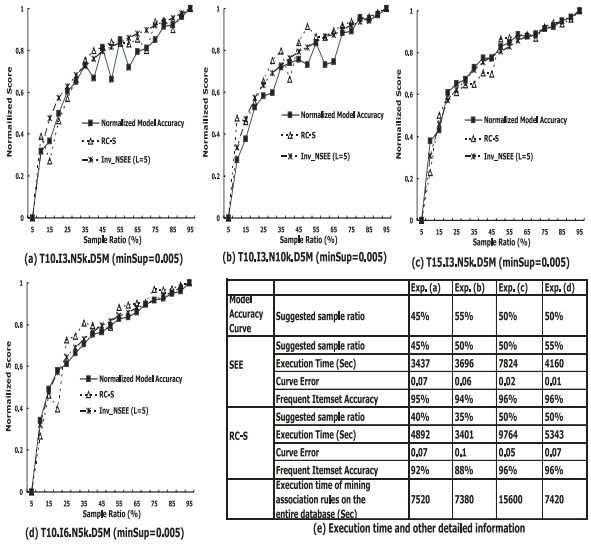
Fig. 3. The sensitivity analysis of different parameters

normalized model accuracy of the sample size s_i and $\varphi(s_i)$ denotes the normalized score of s_i in the curve "*Inv_NSEE*"/"*RC-S*". We can observe that curves *SEE* with different L are all with smaller score distances than the corresponding score distance of *RC-S*, meaning that *SEE* can more correctly estimate the variation of the *model accuracy curve*. Moreover, $L = 5$ is sufficient to obtain an accurate *sampling error curve* because the differences between $L = 5$, 10, and 20 are very small. Thus we can use acceptable memory to store frequencies of each item in 5 samples of the same size, showing the practicability of algorithm *SEE*. Furthermore, we observe the influence of the *minimum support* in Figure 3(b). Results of two different *minimum supports* are shown. We can observe that algorithm *SEE* has the smaller score distance than that of algorithm *RC-S* under different *minimum supports*. Moreover, changing the *minimum support* will not obviously influence the result of algorithm *SEE*, indicating that *SEE* is robust under different parameters of association rules.

Experiments on Synthetic Data. The observations on various synthetic data are shown in Figure 4. We generate four different synthetic data with different "the average transaction length", "the average length of maximal patterns" and "number of different items" (denoted as **T**, **I**, **N** in the name of the generated data, respectively). The number of transactions is set to 50,000,000 in all generated data to mimic the large database.

We observe that *SEE* can save a lot of time and obtain a sufficiently correct model result. On the other hand, *RC-S* may have a smaller execution time in some cases but the obtained model accuracy will be not so acceptable, showing the ability of SEE to balance the efficiency and the model accuracy.

Furthermore, the execution time of four synthetic data, which database sizes vary from 5×10^7 to 2×10^8, are shown in Figure 5. In this experiment, "the average transaction length" is set to 15, and "the average length of maximal patterns" is set to 3. In addition, "the number of different items" is set to 10^6. In this experiment, algorithms *SEE* and *RC-S* similarly suggest sample ratios 40%˜45% as the appropriate sample ratios for association rules, which can achieve a 96% model accuracy when we execute association rules on a sample of the identified sample size. In practice, we observe that the execution time of



		Exp. (a)	Exp. (b)	Exp. (c)	Exp. (d)
Model Accuracy Curve	Suggested sample ratio	45%	55%	50%	50%
SEE	Suggested sample ratio	45%	50%	50%	55%
	Execution Time (Sec)	3437	3696	7824	4160
	Curve Error	0,07	0,06	0,02	0,01
	Frequent Itemset Accuracy	95%	94%	96%	96%
RC-S	Suggested sample ratio	40%	35%	50%	50%
	Execution Time (Sec)	4892	3401	9764	5343
	Curve Error	0,07	0,1	0,05	0,07
	Frequent Itemset Accuracy	92%	88%	96%	96%
	Execution time of mining association rules on the entire database (Sec)	7520	7380	15600	7420

(e) Execution time and other detailed information

Fig. 4. Experiments on various synthetic data

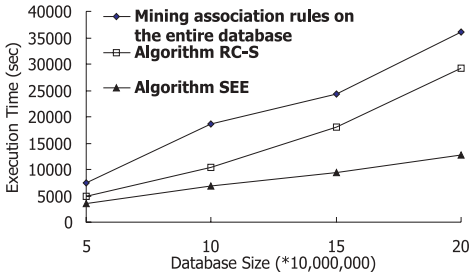


Fig. 5. The execution time on various database size

SEE is smaller than that of *RC-S* in all cases. When the database size increases, algorithm *RC-S* cannot effectively reduce the execution time because it will suffer from the need of considerable I/O operations (executing association rules on several samples). On the other hand, algorithm *SEE* only requires I/O operations of one database scan, showing high execution efficiency.

5 Conclusion

In this paper, we devise a progressive sampling algorithm, *SEE*, to identify an appropriate sample size for mining association rules with two advantages over previous works. First, *SEE* is highly efficient because the appropriate sample size can be identified without the need of executing association rules on samples

and on the entire database. Second, the identified sample size will be a proper sample size since it is determined as the corresponding sample size at the convergence point of the *sampling error curve*, which can effectively estimate the *model accuracy curve*. As shown by experiments on various real data and synthetic data, the efficiency and the effectiveness of *SEE* significantly outperform previous works.

Acknowledgments

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
2. S. Parthasarathy. Efficient progressive sampling for association rules. In *Proc. of ICDM*, 2002.
3. F. Provost, D. Jensen, and T. Oates. Efficient progressive sampling. In *Proc. of SIGKDD*, 1999.
4. R. L. Scheaffer, W. Mendenhall, and R. L. Ott. *Elementary Survey Sampling*. Duxbury Press, 1995.
5. H. Toivonen. Sampling large databases for association rules. In *Proc. of VLDB*, 1996.
6. M. J. Zaki, S. Parthasarathy, M. O., and W. Li. New algorithms for fast discovery of association rules. In *Proc. of SIGKDD*, 1997.
7. M.J. Zaki, S. Parthasarathy, Wei Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. In *Int. Workshop on Research Issues in Data Engineering*, 1997.