COMMUNICATIONS IN INFORMATION AND SYSTEMS
Vol. 2, No. 1, pp. 53-68, June 2002

003

DATA REDUCTION VIA ADAPTIVE SAMPLING*

XIAO-BAI LI†

Abstract. Data reduction is an important issue in the field of data mining. This article describes a new method for selecting a subset of data from a large dataset. A simplified chi-square criterion is proposed for measuring the goodness-of-fit between the distributions of the reduced and full data sets. Under this criterion, the data reduction problem can be formulated as a binary quadratic program and a tabu search technique is used in the search/optimization process. The procedure is adaptive in that it involves not only random sampling but also deterministic search guided by the results of the previous search steps. The method is applicable primarily to discrete data, but can be extended to continuous data as well. An experimental study that compares the proposed method with simple random sampling on a number of simulated and real world datasets has been conducted. The results of the study indicate that the distributions of the samples produced by the proposed method are significantly closer to the true distribution than those of random samples.

Keywords. Data reduction, data mining, chi-square, goodness-of-fit, tabu search, binary quadratic programming.

1. Introduction.

In recent years, we have observed an explosion of electronic data generated and collected by individuals, corporations, and government agencies. It was estimated several years ago that the amount of data in the world was doubling every twenty months [5]. By current standards, that estimate is no doubt too conservative. The widespread use of bar codes and scanning devices for commercial products, the computerization of business and government transactions, the rapid development of electronic commence over the Internet, and the advances in storage technology and database management systems have allowed us to generate and store mountains of data. This rapid growth in data and databases has created the problem of data overload. There has been an urgent need for new techniques and tools that can extract useful information and knowledge from massive volumes of data. Consequently, an emerging field, known as data mining, has flourished in the past several years [4].

Data mining is the process of discovering hidden patterns in databases. The entire process includes (loosely) three steps: (1) data preparation, which includes data collection, data cleaning, data reduction and data transformation; (2) pattern exploration, which involves developing (or using existing) algorithms and computer programs to discover the patterns of interest; and (3) implementation, in which the patterns discovered in the previous step are used to solve real world problems such as credit evaluation, fraud detection, and customer relationship management. Although it is commonly acknowledged that data preparation is often the most involved and potentially most important step in the data mining process, there have been sur-

*Received on February 25, 2002, accepted for publication on June 11, 2002.

†School of Management, JO 44, University of Texas at Dallas, 2601 N. Floyd Road, Richardson, TX 75083, E-mail: lixbob@utdallas.edu

prisingly few studies of problems in this area, as compared to the other areas. This article addresses a key issue in data preparation, namely, data size reduction, where the size refers to the number of records (rows) in a dataset. The number of attributes (columns) in a dataset is often called dimensionality. The problem of dimensionality reduction, known as feature selection, has been studied more extensively (see [11] [16]) and will not be discussed in this paper.

The presence of massive datasets can cause serious problems in an organization's decision support systems and database management systems. First, many data mining and decision support systems cannot handle a dataset with size larger than a certain limit (e.g., memory size). Second, the time spent on mining a large dataset can be prohibitive even if the data size itself is not a constraint to the systems. Third, maintaining and managing large volumes of data can be very expensive in required personnel and storage equipment. On the other hand, some researchers have argued that more effective data mining can be done by working on a reduced dataset instead of the full set, as a statistician put it: "A powerful computationally intensive procedure operating on a sub-sample of the data may in fact provide superior accuracy than a less sophisticated one using the entire data base."

A number of studies on data reduction problems have been done recently. Weiss and Indurkhya [20] and Pyle [16] presented comprehensive reviews of sampling techniques in the context of data reduction for data mining. Provost and Kolluri [15] provided an in-depth discussion of data reduction techniques for inductive machine learning. Han and Kamber [9] offered some guidelines for data reduction in general. Catlett [2] studied a variety of procedures for selecting subsets from a large dataset and compared empirically the results of using different techniques. Quinlan [17] used a sampling technique, called windowing, in his C4.5 decision tree programs to handle large datasets. Common to all of the data reduction techniques described or proposed in these studies is that they are mainly based on statistical sampling techniques, such as simple random sampling, stratified sampling or cluster sampling (see [3] for details of these sampling techniques). There have also been some data reduction methods that incorporate random sampling with adaptive procedures [8] [10] [14] [18]; but these methods were intended for use in conjunction with a specific data mining technique such as decision trees or association rule algorithms.

There is a weakness in almost all of the existing data reduction methods: they fail to recognize a key difference between statistical sampling and data reduction in the context of data mining. In statistical sampling, data is viewed as an expensive resource and it is assumed that collecting population data is practically impossible. The purpose of sampling is to allow us to draw statistical inferences about the *unknown* population from sample data. Therefore, sampling procedures must be stochastic in nature. In data reduction from large databases, data stored in the databases are so large in size that they are normally regarded as "population" data (often, they indeed represent the population). The concern of data reduction centers on getting a subset

of data that best represents the *known* "population". Therefore, techniques for data reduction do not have to be truly stochastic. In fact, since there exists a known object (the full dataset) as a reference for optimization, it should be beneficial to incorporate deterministic search mechanism in a data reduction procedure in finding better representative subsets.

In this paper, we present a new method for selecting a subset of data from a large dataset. A simplified chi-square statistic is proposed for measuring the goodness-of-fit (or closeness-of-fit) between the distributions of the reduced and full datasets. Under the simplified chi-square criterion, the data reduction problem can be formulated as a binary quadratic program. The global optimal solution to the problem is computationally intractable for large datasets and therefore a tabu search technique is used in the search/optimization process. The procedure is adaptive in nature. It begins with a random sample of the full set as an initial subset, and then repeatedly swaps a record inside the subset with another outside. A candidate swap is drawn in random but is committed only if it is not a tabu swap or it improves the value of the global objective function. The procedure is very fast and produces a significantly better subset, in terms of the closeness to the true distribution, than a simple random sample. The method is applicable primarily to discrete data, but can be extended to continuous data as well.

The rest of the paper is organized as follows. The simplified chi-square criterion is proposed and discussed in the next section. The third section describes details of the adaptive sampling procedure, including the tabu search method used. An experimental study that compares the proposed method with the simple random sampling on a number of simulated and real world datasets has been conducted and is described in Section 4. We conclude our study and discuss potential extensions in Section 5.

2. Simplified Chi-Square Criterion. Given a large dataset, our objective is to find a reduced dataset whose frequency distribution is as close to the true distribution as possible, where the true distribution refers to the frequency distribution of the original dataset. The size of the reduced set is expected to be substantially smaller than that of the full set.

A classical measure of the closeness of (or distance between) the actual and expected distributions is the chi-square goodness-of-fit statistic, given by

$$X^2 = \sum \frac{(n_i - m_i)^2}{m_i}, \tag{1}$$

where n_i is the frequency of the actual (or observed) distribution; m_i is the frequency of the expected distribution; and subscript i runs over all possible category combinations (all cells in the contingency tables). When the data values are of the continuous type, they are grouped into certain intervals before applying equation (1). The statistic X^2 follows asymptotically a χ^2 distribution with appropriate degrees of freedom.

To apply equation (1) to the data reduction problem, let N and n be the sizes of the original and reduced datasets, respectively, and call $p = n/N$ the sampling proportion. Let J be the number of attributes and $K_j\,(j = 1, \ldots, J)$ be the number of categories (intervals) in the jth attribute. Call each category combination a *pattern*. Then the total number of patterns can be as large as $\prod_{j=1}^{J} K_j$. The actual number of patterns, C, may be smaller since some of the patterns may not appear in the data. Let N_i and $n_i\,(i = 1, \ldots, C)$ be the frequencies of the ith pattern in the original and reduced datasets, respectively. Then, X^2 can be computed by

$$X^2 = \sum_{i=1}^{C} \frac{(n_i - pN_i)^2}{pN_i}, \tag{2}$$

Now, it appears that the data reduction problem described at the beginning of this section can be translated to finding a reduced dataset that minimizes the X^2 value in equation (2). However, there are two serious problems when equation (2) is applied to large datasets. First, quantity C can be a huge number for a large dataset. For instance, a dataset with 20 attributes of 5 categories each could potentially have 5^{20} different patterns. In this case, it is unlikely to compute N_i from the data under current computing capacity. Therefore, equation (2) is, in general, computationally prohibitive in a data mining context. Second, in a large dataset, the frequencies of different patterns often vary significantly. Some patterns may have hundreds or thousands of replicates; others may have very few. When the latter case is not negligible, X^2 will depart more or less from a χ^2 distribution and the degrees of freedom associated with it will be difficult to determine. Consequently, the validity of the chi-square goodness-of-fit test is in doubt. To overcome these problems, we propose the following simplified chi-square statistic:

$$\begin{aligned} X_s^2 &= \sum_{j=1}^{J} \sum_{k=1}^{K_j} \frac{(n_{jk} - pN_{jk})^2}{pN_{jk}} \\ &= \sum_{i=1}^{K} \frac{(n_i - pN_i)^2}{pN_i}, \end{aligned} \tag{3}$$

where n_{jk} is the frequency of the kth category of the jth attribute in the reduced dataset; N_{jk} is the corresponding frequency in the original set; and $K = \sum_{j=1}^{J} K_j$. Since the frequencies in equation (3) are marginal frequencies instead of joint frequencies, quantity K is generally much smaller than C, and the required memory space and the amount of computation involved are reduced substantially. For instance, the dataset with 20 attributes of 5 categories each will have only 20×5 terms in the summation.

Our question now is (1) whether X_s^2 is a reasonably good measure of the closeness of two distributions and (2) what distribution form the X_s^2 statistic has. The following theorem provides the answer to the second question.

THEOREM 2.1. *The simplified chi-square X_s^2 has a χ^2 distribution with $K - J$*

degrees of freedom; that is,

$$X_s^2 \sim \chi^2(K - J). \tag{4}$$

Proof. X_s^2 can be written as $X_s^2 = \sum_{j=1}^{J} x_j^2$, where $x_j^2 = \sum_{k=1}^{K_j} \frac{(n_{jk} - pN_{jk})^2}{pN_{jk}}$ is the sum of the squared frequency deviation for the jth attribute. Note that the frequency count of categories in each attribute has a multinomial distribution and the size of each n_{jk} or pN_{jk} is apparently large enough to apply the central limit theorem. Based on Pearson's work [13], x_j^2 follows a $\chi^2(K_j - 1)$ distribution. By the additivity property of χ^2 distributions, we have $X_s^2 \sim \chi^2(K - J)$. □

To answer the first question, let us look at the relationship between the probability distribution of the original dataset,

$$P_N(\mathbf{X}) = P_N(X_j)P_N(\mathbf{X}_{\setminus j}|X_j), \tag{5}$$

and that of the reduced set,

$$P_n(\mathbf{X}) = P_n(X_j)P_n(\mathbf{X}_{\setminus j}|X_j), \tag{6}$$

where $\mathbf{X}_{\setminus j}$ represents the set of all attributes, excluding X_j. The X^2 statistic in equation (2) is a good measure of the closeness of two distributions in that it represents the closeness of the two joint distributions, $P_N(\mathbf{X})$ and $P_n(\mathbf{X})$. (The fact that X^2 is not computationally practical and its distribution form may not be clear in large settings is irrelevant to whether it is a good measure of closeness.) The X_s^2 statistic, on the other hand, represents the closeness of the two marginal distributions, $P_N(X_j)$ and $P_n(X_j)$. An optimization procedure based on the X_s^2 criterion may not yield consistent results with those based on the X^2 criterion unless the conditional probability distribution remains relatively unchanged. However, since our proposed optimization algorithm (discussed in the next section) minimizes the differences between $P_N(X_j)$ and $P_n(X_j)$ for all j simultaneously, it is reasonable to argue that the decreasing rate of these differences will generally be greater than the rate of the changes in the differences between $P_N(\mathbf{X}_{\setminus j}|X_j)$ and $P_n(\mathbf{X}_{\setminus j}|X_j)$. Therefore, minimizing X_s^2 will, in general, causes the X^2 value to decrease. Further, for a chi-square distribution with large degrees of freedom (*df*), its cutoff point values fall into a narrow range centered at *df*. A small drop in the X^2 value caused by minimizing X_s^2 will lead to a significant drop in probability of rejecting the null hypothesis that two distributions are the same. (This fact can be observed from a χ^2 distribution table or any software package that generates the χ^2 distribution.) This is true even though X^2 departs somewhat from a χ^2 distribution. Therefore, the decrease in the X^2 value caused by minimizing X_s^2 is often statistically significant. We will see this effect in the experiments in Section 4.

3. Adaptive Sampling Procedure. Having established the simplified chi-square as the minimization criterion, we now formulate our data reduction problem in

a more rigorous form, as a quadratic optimization problem, and describe our proposed adaptive sampling procedure for data reduction.

Given a large dataset $\mathbf{X}$ of size $N \times J$ (N rows/records, J columns/attributes), our objective is to find a reduced dataset $\mathbf{x}$ of size $n \times J$ that minimizes the simplified chi-square in equation (3). Continuing the notations in the previous section, the problem can be formulated as a binary quadratic program as follows:

$$\min \; X_s^2 = \sum_{j=1}^{J} \sum_{k=1}^{K_j} \frac{(n_{jk} - pN_{jk})^2}{pN_{jk}}, \tag{7a}$$

$$\text{s.t.} \quad \sum_{r=1}^{n} y_{rjk} = n_{jk}, j = 1, \ldots, J; \; k = 1, \ldots, K_j, \tag{7b}$$

(7c)

y_{rjk} $= 1$ if the (r, j) entry value in $\mathbf{x}$ is the kth category of the jth attribute,
$= 0$ otherwise,
(this implies $\sum_{k=1}^{K_j} y_{rjk} = 1, r = 1, \ldots, n; \; j = 1, \ldots, J$).

In addition, a row vector formed by $y_{rjk}\,(j = 1, \ldots, J; \; k = 1, \ldots, K_j)$ must correspond to a row in $\mathbf{X}$. Note that $p = n/N$ and N_{jk} are known values. If there are missing values in $\mathbf{X}$, the missing item is counted as a category labeled as, say, "missing"; that is, K_j represents the number of categories in the jth attribute, including the "missing" category. Like most integer programs, there is no "slick" solution to problem (7). We have to rely on enumeration based search technique. Our basic strategy is to draw first a random sample of size n from $\mathbf{X}$, and then repeatedly swap a record inside the sample with a record outside to reduce the value of the objective function (7a). The global optimal solution to the problem is computationally intractable for large data size N. In order to find a good solution with limited iterations, we implement a tabu search technique in our optimization/searching procedure.

Tabu search is a meta-heuristic that guides a local heuristic search procedure to overcome local optimality. The local search typically involves the use of an operation called *move* to identify the neighborhood of a given solution. In order to escape the trap of local optimality, the method records recent moves in a tabu list, or tabu search memory, and forbid or penalize new moves that attempt to repeat or retrace the recorded moves. The tabu list, which has a certain size, is updated after each iteration. Therefore, a move recorded on the tabu list will stay in memory only for a certain period. Tabu search also allows a move on the tabu list to be included in the new solution if the move improves the level of *aspiration* function, which is typically related to the global optima of the objective function. (See [7] for a comprehensive treatment of tabu search.) Tabu search has been applied to various combinatorial

optimization problems, including binary quadratic programming [6] and quadratic assignment problem [19].

In the data reduction problem, after initial sampling, a move is a swap between a record inside the sample and a record outside. A major difference between this problem and other integer programming related tabu search problems is that a swap of two records in this problem involves a simultaneous change of multiple attribute values, while a move in the other problems normally involves only one attribute. Therefore, it is not appropriate to place a sequence of recent swaps into the tabu list, as most of the other tabu search applications do, because the compound effect of forbidding these swaps is rather complicated and unpredictable. In our proposed procedure, we first check if the two records to be interchanged have the same value in each of the respective attributes, and draw a new pair of records if this situation occurs. Based on the nature of our problem, we then place two restrictions on the tabu list. To explain these restrictions, let us look at the objective function (7a), which can be expressed, using the notations in the proof of Theorem 2.1, as below:

$$\min \; X_s^2 = x_1^2 + x_2^2 + \cdots + x_J^2, \tag{7a$'$}$$

where the individual summation term, $x_j^2 = \sum_{k=1}^{K_j} \frac{(n_{jk} - pN_{jk})^2}{pN_{jk}}$, represents the closeness of the marginal distribution of the jth attribute. When $x_j^2 = 0$, it indicates a perfect fit in the jth marginal distribution. When this situation occurs, we want to keep it in the subsequent iterations. Therefore, a restriction is placed such that once a x_j^2 reaches zero, any swap that changes the term to positive is forbidden (all x_j^2's are nonnegative, obviously). The second restriction is to forbid a swap that causes the smallest non-zero summation term to deteriorate (increase). However, if a swap improves the aspiration level, which is defined as the best (smallest) X_s^2 value that has been achieved up to the current iteration, the swap will be selected even if this restriction is violated. Since each summation term x_j^2 measures the gap between two marginal distributions, the purpose of these two restrictions is to avoid selecting swaps that allow gaps flow from one term to the other. In some rare occasions, it could be difficult to find a swap that does not violate the above restrictions. To prevent long trials at one iteration, a threshold number can be prespecified at the beginning of the process. If the number of trials in searching a swap exceeds this threshold, the existing tabu list is destroyed, which would allow a new swap to be selected quickly. Another parameter specified by the user is the total number of iterations, which is dependent on the sizes of the full and reduced datasets, N and n, as well as the time that the user is willing to spend on the computation. Having described the details of the procedure, we present the computation algorithm for adaptive sampling in Table 1.

TABLE 1
Adaptive Sampling Algorithm

0. Let $\mathbf{X}$ be a data matrix of size $N \times J$. Let M be the total number of iterations and set iteration counter $t = 0$. Let $\mathbf{x}_t$ be the sample data selected at the tth iteration. Let m be the total number of trials within an iteration.
1. Draw a simple random sample $\mathbf{x}_0$ of size $n \times J$ from $\mathbf{X}$. Compute X_s^2 using equation (7a′). Set $X_{\min}^2 = X_s^2$, where $X_{\min}^2$ represents the minimum objective function value. Set the best current sample $\mathbf{x}^* = \mathbf{x}_0$. Create the tabu list b as a J-dimensional vector, with jth element $b_j = x_j^2$, where x_j^2 is the jth summation term in (7a′).
2. Set $t = t + 1$. Select a swap as follows:
   a. Set $i = 0$.
      (i) Set $i = i + 1$. If $i > m$, reset each component of b to an arbitrarily large number and go to step 3; otherwise, randomly select a row u from $\mathbf{X}$ and a row v from $\mathbf{x}_{t-1}$.
      (ii) If $u_j = v_j$ for all j, go to (i); otherwise, add u to $\mathbf{x}_{t-1}$ and drop v from $\mathbf{x}_{t-1}$. Denote the resulting sample as $\tilde{\mathbf{x}}_t$.
      (iii) Update the X_s^2 (and x_j^2) value based on $\tilde{\mathbf{x}}_t$. If there is a subscript j ($j = 1, \ldots, J$) such that $x_j^2 > 0$ and $b_j = 0$, then go to (i); otherwise, go to (iv).
      (iv) Let $b_{j*} = \min_j \{b_j | b_j > 0\}$. If $x_{j*}^2 > b_{j*}$ and $X_s^2 \geq X_{\min}^2$, then go to (i); otherwise, go to step b.
   b. If $\tilde{\mathbf{x}}_t = \emptyset$, set $\mathbf{x}_t = \mathbf{x}_{t-1}$; otherwise, set $\mathbf{x}_t = \tilde{\mathbf{x}}_t$ and update the tabu list by assigning the new x_j^2 value to b_j ($j = 1, \ldots, J$), respectively. If $X_s^2 < X_{\min}^2$, set $X_{\min}^2 = X_s^2$ and $\mathbf{x}^* = \mathbf{x}_t$. Go to step 3.
3. Repeat step 2. Stop if $t > M$ or $X_{\min}^2 = 0$.

Next, we discuss the computational complexity of the proposed algorithm. The algorithm basically includes two phases: (1) select an initial random sample and compute X_s^2 (step 1); and (2) repeatedly swap a pair of records to minimize X_s^2 (steps 2 and 3). The most time-consuming operation in phase (1) is the computation of X_s^2. It involves a full scan of the entire datasets $\mathbf{X}$ and $\mathbf{x}$, respectively. During the scan, the algorithm identifies the category value of each attribute and counts the frequency of each category. The total number of searching and counting operation is $(N + n)\sum_{j=1}^{J} K_j$ or $(N + n)K$, which is of order $O(N)$ since N is dominantly larger than n and K. Therefore, the time complexity of the first phase is linear in data size N.

The second phase includes two loops. The outer loop repeats exactly M times and the inner loop repeats at most m times. In the inner loop, the algorithm selects a candidate swap, checks its tabu status, and updates the X_s^2 and tabu list values. Note

that updating X_s^2 is much faster than computing X_s^2 from scratch because it only involves adjusting the existing frequency tables, instead of scanning the entire datasets. The time complexity of this operation is of order $O(K)$. The other operations in the inner loop have the same or lower order of time complexity. The maximum number of runs for the inner loop, m, is often set as a small number (between 5 and 10) and, based on our experience, the actual number of runs is even smaller (1 or 2 in most cases). The effect of this number on computing time is thus not considered a factor. Therefore, the time complexity of the second phase is of order $O(KM)$. Apparently, the user has control over the computing time in this phase since M is set by the user. To sum up, the time complexity of the adaptive sampling algorithm is of order $O(N) + O(KM)$. Clearly, the algorithm is capable of handling large data reduction problems in a data mining context.

4. Experiments. In this section, we describe an experimental study that compares the proposed adaptive sampling with simple random sampling on a number of simulated and real world datasets. The primary objective of using simulated data is to allow us to observe empirically the relationship between the X^2 statistic in equation (2) and the proposed X_s^2 statistic, while the purpose of using real world data is to see how the adaptive sampling performs in real settings against random sampling. Therefore, the experiments for the simulated and real world data are described separately.

Four artificial datasets were generated using SAS package. A summary of these four datasets is given in Table 2 below, where $2 \times 2 \times 2$ indicates that the dataset has three attributes, each with two categories, and so on. The datasets were generated in a way such that the frequency of each pattern (category combination) in each dataset ranges between 100 and 2000. This frequency range, together with properly chosen sampling proportions, would ensure that each pattern after sampling has a sufficient number of records for a valid chi-square goodness-of-fit test. We were unable to experiment with datasets of a larger dimensionality (i.e., a larger number of attributes and/or categories) since the number of patterns increases exponentially with dimensionality, which makes the calculation of X^2 virtually impossible. We also avoided generating a dataset that has a larger number of records but a low dimensionality (which implies a high frequency for each pattern), since this kind of data is uncommon in practice.

We set the maximum number of runs for the outer and inner loops, M and m, to 500 and 10, respectively, for each dataset. In the experiments, we were able to obtain the optimal solution (of problem (7)), where $X_s^2 = 0$, for each dataset in fewer than M iterations. The results of the experiments are shown in Table 3. Because the dimensionality of the datasets is relatively low, we were able to compute the X^2 and its corresponding p-value in each dataset. Table 3 lists the X^2, X_s^2 and their corresponding p-values for the initial random sample and final sample (the result of running the adaptive algorithm on the initial sample), respectively. Since a small p-

value favors rejecting the null hypothesis that the sample has the same distribution as the full dataset does, a larger p-value (and smaller X^2 value) is desired. Also given in the table are the actual number of iterations and the CPU computing time in seconds to find the optimal solution.

TABLE 2
Summarized Descriptions of Simulated Data

Dataset Name	Number of Records	Number of Attributes & Categories	Sampling Proportion
A	5000	2×2	0.02
B	8000	$2 \times 2 \times 2$	0.02
C	13500	$3 \times 3 \times 2 \times 2$	0.05
D	24000	$2 \times 2 \times 2 \times 2 \times 2 \times 2$	0.05

TABLE 3
Results on Simulated Data

	Initial Sample					
Data	X^2	p-val	X_s^2	p-val		
A	1.48	**0.69**	1.36	0.51		
B	6.00	**0.54**	5.23	0.16		
C	32.5	**0.59**	6.67	0.35		
D	80.7	**0.07**	13.4	0.04		
	Final Sample					
Data	X^2	p-val	X_s^2	p-val	No. of Runs to Converge	CPU Time in Seconds
A	0.21	**0.98**	0	1.00	12	5
B	1.40	**0.99**	0	1.00	68	21
C	26.4	**0.85**	0	1.00	155	87
D	52.7	**0.82**	0	1.00	183	161

It is very clear from Table 3 that minimizing X_s^2 causes the X^2 value to decrease. More importantly, the improvement in the p-value for each X^2 is quite substantial, as shown in the two highlighted columns. For dataset D, the p-value for X^2 in the initial random sample is 0.07, which could lead to a decision of rejecting the null hypothesis that the sample follows the same distribution as the original data does, depending on the significance level used. In the final sample, this p-value improves to 0.82, well

above any reasonable significance level. It appears that the simple random sampling may or may not select a sample with a sufficiently large p-value; but after the sample is processed by the algorithm, the p-value tends to increase to a level very close to one.

Next, we describe the experiments on the real world data. It is very difficult, for obvious reasons, to find a real world dataset of ideal size. U. C. Irvine maintains a large collection of real world datasets [1], but those datasets are too small to be used as data reduction examples. We nevertheless selected one set from this database for our study. This dataset, initially used in [12], consists of 699 records of patients with breast cancer, each having 9 attributes of integer type representing 9 medical measurements, and a class label representing diagnostic decision. The integer values range from 1 to 10. We selected this data due to a number of considerations. First, we would like to see if the proposed algorithm could be readily applied to a dataset with integer attributes, where the number of integer values is not too large so that each integer can be treated as a category. Second, the dimensionality of this data is rather high (2×10^9 possible patterns) compared to the data size. We want to see how the proposed algorithm performs in this situation. Third, this dataset has missing values.

The second dataset was collected from a company in travel industry. The dataset contains 30384 customer records, each with 22 categorical attributes (four of them were converted from numeric attributes). These attributes include customer descriptions such as age, gender, membership status, credit card type, payment amount, the number of transactions in a certain period, and so on. Due to company confidentiality requirements, it is not possible to reveal more information about the data. The dimensionality of the dataset is reasonably large but the size is not. We collected and used this data in our study because this kind of data is quite typical in data mining applications such as credit evaluation and customer relationship management. A summary of the two real world dataset is provided in Table 4.

Since X^2 is computationally intractable for both datasets, we resort to an indirect method to check the quality of the processed samples. Both datasets have a class attribute that allows us to perform classification analysis on the data (the breast data has been used for testing various classification algorithms in numerous studies). The classification results based on the full and reduced sets can be used to indirectly evaluate the effectiveness of different sampling methods. A popular decision tree system, C4.5 [17], was used to perform the classification analysis. For each dataset, we randomly selected 20% of the records and reserved the data as a common test set for evaluating decision trees built based on different data sizes. This part of the data is not involved in the data reduction process. The remaining 80% of the data forms the full set and sampling methods were applied to this part of the data only. The sampling proportions in Table 4 are larger than those used in the simulated data because the dimensionalities of these sets are much larger and we want to avoid losing

too many patterns after sampling.

The maximum number of runs for the outer and inner loops, M and m, was set to 1000 and 10, respectively, for both datasets. The results of the experiments are given in Table 5, where the CPU time under the initial sample column indicates the time spent on step 1 of the algorithm and under the final sample column is the extra processing time on steps 2 and 3. Neither sample converges to zero in 1000 runs but the X_s^2 values in both cases drops significantly after processing. Decision trees were built based on the data in the full set, initial and final samples, respectively, and tested using the reserved test set. The results are given in Table 6, where the error rate is the test error rate and the tree size is the average number of nodes in the decision trees. The error rate is a primary measure of the quality of decision trees while the tree size measures the simplicity of decision trees. If the test error rate based on a sample is close to that based on the full set, it will be reasonable to argue that the relationships among various attributes in the full set are well captured by the sample. A similar statement does not apply to the comparison of tree size since smaller data size tends to result in smaller trees. However, if the error rates are about the same, then a smaller tree size is desirable.

TABLE 4
Summarized Descriptions of Real World Data

Dataset Name	No. of Records	No. of Attributes	Max. No. of Categories in an Attribute	Sampling Proportion
Breast	693	10	10	0.30
Travel	30384	22	8	0.20

TABLE 5
Sampling Results on Real World Data

	Initial Sample			Final Sample	
Dataset Name	X_s^2	CPU Time in Seconds		X_s^2	Extra CPU Time in Seconds
Breast	62.68	3		3.82	55
Travel	47.00	108		0.54	1025

It is evident that both samples have significantly smaller tree sizes than the full set has while keeping their error rates comparable to those of the full set. The tree size for the final sample in the travel data is substantially smaller than those of the other two. For the breast data, the final sample has the same error rate as the full set while the initial random sample does have a notable higher error rate. For the travel data, the final sample again has almost the same error rate as the full set while the initial

TABLE 6
C4.5 Classification Results on Real World Data

	Error Rate				Tree Size		
Dataset Name	Full Set	Initial Sample	Final Sample		Full Set	Initial Sample	Final Sample
Breast	0.029	0.036	0.029		32	21	22
Travel	0.313	0.321	0.312		865	315	276

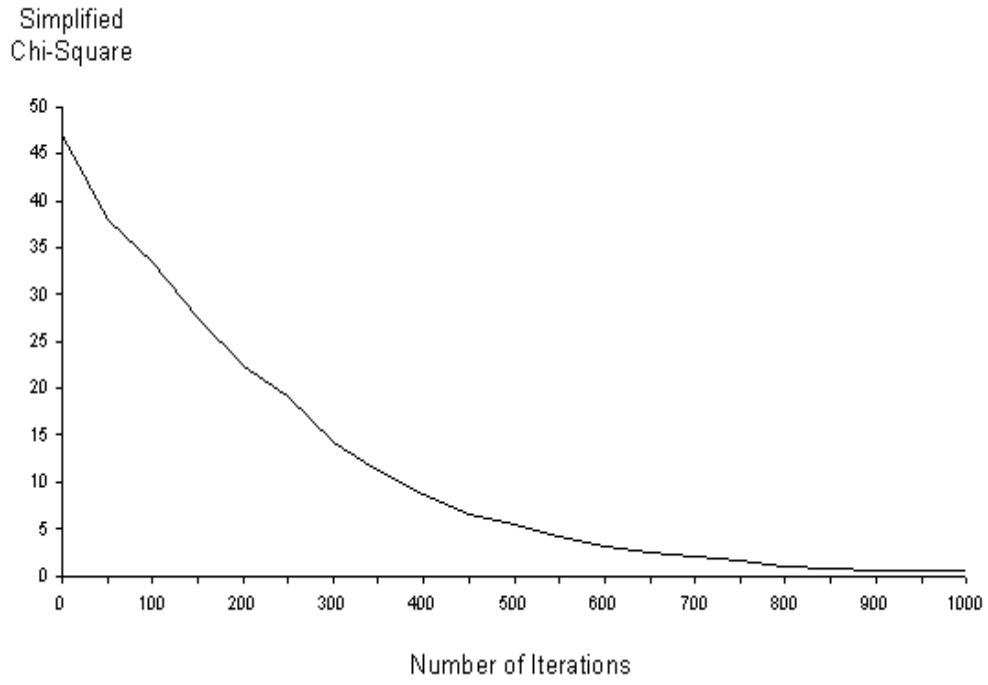

FIG. 1. *Relationship between the Simplified Chi-Square and the Number of Iterations for the Travel Data*

sample has a slightly higher error rate. The differences in error rate for this data are not significant, however. In short, the results are favorable to our proposed sampling method. However, more experiments are needed before reaching any conclusion in this aspect. In terms of computing time, it appears that the extra processing time for the travel data is somewhat long. A further investigation of the relationship between the X_s^2 value and the number of iterations, as shown in Figure 1, indicates that the X_s^2 value drops substantially in the first 300 runs, and levels off after about 500 runs. Therefore, the actual time to find a sample with a significantly smaller X_s^2 can be reduced substantially.

5. Conclusions and Extensions. We have presented a new data reduction method that applies a tabu search technique to minimize a simplified chi-square criterion in selecting a sample from a large dataset. The results of our experimental study are favorable, although more empirical studies are needed to justify the effectiveness of the proposed method.

As we stated earlier, the proposed method is applicable primarily to discrete or categorical data. It is not difficult to extend the method to continuous or numerical data, however. In fact, both of the real world datasets used in our experiments involve numeric data. The easiest way to apply the method to continuous data is to convert them to discrete values by grouping (or binning) the continuous values. With individual record ID properly indexed, the original continuous values can be retrieved for the final sample. Another approach to deal with continuous data is to use a criterion that is appropriate for continuous data, instead of the chi-square related measures. A possible criterion is the mean squared deviation, where the mean refers to the average value of a continuous attribute in the full set, and the deviation should be normalized for each attribute. This criterion seems feasible when the data is continuous over all attributes. The key is to set the criterion properly so that the selected sample will be close to the full set in terms of both the mean and variance. When the data is of a mixed type (both continuous and discrete), it will be difficult, if not impossible, to have a criterion commensurate to both continuous and discrete values. Therefore, conversion between continuous and discrete values seems inevitable.

The simplified chi-square statistic, X_s^2, is based on marginal distribution while the chi-square statistic, X^2, is based on joint distribution. In some occasions, the correlation between X_s^2 and X^2 may be weak and minimizing X_s^2 may not cause X^2 to decrease sufficiently. When this is a serious concern, a modified chi-square measure based on a higher order (higher than marginal) distribution can be considered. For example, a second order chi-square measure that involves the frequency of joint category combinations from any two attributes would generally have higher correlation with X^2 than X_s^2 does. However, the amount of computation for such kind of higher order statistic increases quickly. Therefore, whether or not this approach is practical depends on the size and dimensionality of the dataset.

REFERENCES

[1] C. Blake, E. Keogh and C. J. Merz, *UCI repository of machine Learning databases*, Dept. of Information and Computer Science, University of California, Irvine, CA, 1998, http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[2] J. Catlett, *Megainduction: Machine Learning on Very Large Databases*, Ph.D. Dissertation, School of Computer Science, University of Technology, Sydney, Australia, 1991.

[3] W. G. Cochran, *Sampling Techniques* (3rd ed), John Wiley & Sons, NY, 1977.

[4] U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, *From data mining to knowledge discovery: An overview*, Advances in Knowledge Discovery and Data Mining (U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, eds.), pp.1-34, AAAI Press / MIT

Press, Menlo Park, CA, 1996.

[5] W. J. Frawley, G. Piatetsky-Shapiro and C. J. Matheus, *Knowledge discovery in databases: An overview*, Knowledge Discovery in Databases (G. Piatetsky-Shapiro and C. J. Matheus, eds.), AAAI Press / MIT Press, Menlo Park, CA, 1991.

[6] F. Glover, G. A. Kochenberger and B. Alidaee, *Adaptive memory tabu search for binary quadratic programs*, Management Science, 44:3(1998), pp. 336–345.

[7] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic, Norwell, MA, 1997.

[8] G. John and P. Langley, *Static versus dynamic sampling for data mining*, Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD-96), pp.367-370, AAAI Press, Menlo Park, CA, 1996.

[9] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Mateo, CA, 2001.

[10] H. Liu and H. Motoda (eds.), *Instance Selection and Construction for Data Mining*, Kluwer Academic, Norwell, MA, 2001.

[11] J. Kittler, *Feature selection and extraction*, Handbook of Pattern Recognition and Image Processing (T. Y. Young and K. S. Fu, eds.), Academic Press, NY, 1986.

[12] O. L. Mangasarian, R. Setiono and W. Wolberg, *Pattern recognition via linear programming: Theory and application to medical diagnosis*, Large-Scale Numerical Optimization (T. F. Coleman and Y. Y. Li, eds.), pp.22–30, SIAM Publications, Philadelphia, PA, 1990.

[13] K. Pearson, *On a criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling*, Philos. Mag., 5:50(1900), pp.157–175.

[14] F. Provost, D. Jensen and T. Oates, *Efficient progressive sampling*, Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD-99), pp.23–32, AAAI Press, Menlo Park, CA, 1999.

[15] F. Provost and V. Kolluri, *A survey of methods for scaling up inductive algorithms*, Data Mining and Knowledge Discovery, 3:2(1999), pp.131–169.

[16] D. Pyle, *Data Preparation for Data Mining*, Morgan Kaufmann, San Mateo, CA, 1999.

[17] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.

[18] T. Reinartz, *A unifying view on instance selection*, Data Mining and Knowledge Discovery, 6:2(2002), pp.191–210.

[19] J. Skorin-Kapov, *Tabu search applied to the quadratic assignment problem*, INFORMS Journal on Computing, 2:1(1990), pp.33–45.

[20] S. M. Weiss and N. Indurkhya, *Predictive Data Mining: A Practical Guide*, Morgan Kaufmann, San Mateo, CA, 1997.