Database Sampling for Data Mining

Patricia E.N. Lutu
University of Pretoria, South Africa

INTRODUCTION

In data mining, sampling may be used as a technique for reducing the amount of data presented to a data mining algorithm. Other strategies for data reduction include dimension reduction, data compression, and discretisation. For sampling, the aim is to draw, from a database, a random sample, which has the same characteristics as the original database. This chapter looks at the sampling methods that are traditionally available from the area of statistics, how these methods have been adapted to database sampling in general and database sampling for data mining in particular.

BACKGROUND

Given the rate at which database/data warehouse sizes are growing, attempts at creating faster/more efficient algorithms that can process massive data sets may eventually become futile exercises. Modern database and data warehouse sizes are in the region of 10s or 100s of terabytes, and sizes continue to grow. A query issued on such a database/data warehouse could easily return several millions of records.

While the costs of data storage continue to decrease, the analysis of data continues to be hard. This is the case for even traditionally simple problems requiring aggregation, for example, the computation of a mean value for some database attribute. In the case of data mining, the computation of very sophisticated functions, on very large numbers of database records, can take several hours, or even days. For inductive algorithms, the problem of lengthy computations is compounded by the fact that many iterations are needed in order to measure the training accuracy as well as the generalization accuracy.

There is plenty of evidence to suggest that, for inductive data mining, the learning curve flattens after only a small percentage of the available data from a large data set has been processed (Catlett, 1991; Kohavi, 1996; Provost et al., 1999). The problem of overfitting (Dietterich, 1995) also dictates that the mining of massive data sets should be avoided. The sampling of databases has been studied by researchers for some time. For data mining, sampling should be used as a data reduction technique, allowing a data set to be represented by a much smaller random sample that can be processed much more efficiently.

MAIN THRUST

There are a number of key issues to be considered before obtaining a suitable random sample for a data mining task. It is essential to understand the strengths and weaknesses of each sampling method. It is also essential to understand which sampling methods are more suitable to the type of data to be processed and the data mining algorithm to be employed. For research purposes, we need to look at a variety of sampling methods used by statisticians, and attempt to adapt them to sampling for data mining.

Some Basics of Statistical Sampling Theory

In statistics, the theory of sampling, also known as statistical estimation or the representative method, deals with the study of suitable methods of selecting a representative sample of a population, in order to study or estimate values of specific characteristics of the population (Neyman, 1934). Since the characteristics being studied can only be estimated from the sample, confidence intervals are calculated to give the range of values within which the actual value will fall, with a given probability.

There are a number of sampling methods discussed in the literature, for example the book by Rao (Rao, 2000). Some methods appear to be more suited to database sampling than others. Simple random sampling (SRS), stratified random sampling, and cluster sampling are three such methods. Simple random sampling involves selecting at random elements of the population, P, to be studied. The method of selection may be either with replacement (SRSWR) or without replacement (SRSWOR). For very large populations, however, SRSWR and SRSWOR are equivalent. For simple random sampling, the probabilities of inclusion of the elements may or may not be uniform. If the probabilities are not uniform then a weighted random sample is obtained.

The second method of sampling is stratified random sampling. Here, before the samples are drawn, the population P is divided into several strata, p_1, p_2,.. p_k, and the

sample S is composed of k partial samples $s_1, s_2, .., s_k$, each drawn randomly, with replacement or not, from one of the strata. Rao (2000) discusses several methods of allocating the number of sampled elements for each stratum. Bryant et al. (1960) argue that, if the sample is allocated to the strata in proportion to the number of elements in the strata, it is virtually certain that the stratified sample estimate will have a smaller variance than a simple random sample of the same size. The stratification of a sample may be done according to one criterion. Most commonly though, there are several alternative criteria that may be used for stratification. When this is the case, the different criteria may all be employed to achieve multi-way stratification. Neyman (1934) argues that there are situations when it is very difficult to use an individual unit as the unit of sampling. For such situations, the sampling unit should be a group of elements, and each stratum should be composed of several groups. In comparison with stratified random sampling, where samples are selected from each stratum, in cluster sampling a sample of clusters is selected and observations/measurements are made on the clusters. Cluster sampling and stratification may be combined (Rao, 2000).

Database Sampling Methods

Database sampling has been practiced for many years for purposes of estimating aggregate query results, database auditing, query optimization, and, obtaining samples for further statistical processing (Olken, 1993). Static sampling (Olken, 1993) and adaptive (dynamic) sampling (Haas & Swami, 1992) are two alternatives for obtaining samples for data mining tasks. In recent years, many studies have been conducted in applying sampling to inductive and non-inductive data mining (John & Langley, 1996; Provost et al., 1999; Toivonen, 1996).

Simple Random Sampling

Simple random sampling is by far, the simplest method of sampling a database. Simple random sampling may be implemented using sequential random sampling or reservoir sampling. For sequential random sampling, the problem is to draw a random sample of size n without replacement, from a file containing N records. The simplest sequential random sampling method is due to Fan et al. (1962) and Jones (1962). An independent uniform random variate [from the uniform interval (0,1)] is generated for each record in the file to determine whether the record should be included in the sample. If m records have already been chosen from among the first t records in the file, the $(t+1)^{st}$ record is chosen with probability (RQsize/RMsize), where RQsize = (n-m) is the number of records that still need to be chosen for the sample, and RMsize = (N-t) is the number of records in the file still to be processed. This sampling method is commonly referred to as method S (Vitter, 1987).

The *reservoir sampling* method (Fan et al., 1962; Jones, 1962; Vitter, 1985, 1987) is a sequential sampling method over a finite population of database records, with an unknown population size. Olken (1993) discuss its use in sampling of database query outputs on the fly. This technique produces a sample of size S, by initially placing the first S records of the database/file/query in the reservoir. For each subsequent k^{th} database record, that record is accepted with probability S/k. If accepted, it replaces a randomly selected record in the reservoir.

Acceptance/Rejection sampling (A/R sampling) can be used to obtain *weighted samples* (Olken, 1993). For a weighted random sample, the probabilities of inclusion of the elements of the population are not uniform. For database sampling, the inclusion probability of a data record is proportional to some weight calculated from the record's attributes. Suppose that one database record r_j is to be drawn from a file of n records with the probability of inclusion being proportional to the weight w_j. This may be done by generating a uniformly distributed random integer $1 \le j \le n$ and then accepting the sampled record r_j with probability $\alpha_j = w_j / w_{max}$, where w_{max} is the maximum possible value for w_j. The acceptance test is performed by generating another uniform random variate u_j, $0 \le u_j \le 1$, and accepting r_j iff $u_j < \alpha_j$. If r_j is rejected, the process is repeated until some r_j is accepted.

Stratified Sampling

Density biased sampling (Palmer & Faloutsos, 2000) is a method that combines clustering and stratified sampling. In density biased sampling, the aim is to sample so that within each cluster points are selected uniformly, the sample is density preserving, and the sample is biased by cluster size. Density preserving in this context means that the expected sum of weights of the sampled points for each cluster is proportional to the cluster's size. Since it would be infeasible to determine the clusters apriori, groups are used instead to represent all the regions in n-dimensional space. Sampling is then done to be density preserving for each group. The groups are formed by "placing" a d-dimensional grid over the data. In the d-dimensional grid, the d dimensions of each cell are labeled either with a bin value for numeric attributes, or by a discrete value for categorical attributes. The d-dimensional grid defines the strata for multi-way stratified sampling. A one-pass algorithm is used to perform the weighted sampling, based on the reservoir algorithm.