Sampling and Its Application in Data Mining: A Survey

Baohua Gu[*], Feifang Hu[†], Huan Liu[‡]
National University of Singapore, Singapore

Abstract

Large data sets are becoming obstacles for efficient data mining. Sampling, as a well-established technique in statistics, is desired to play its role in overcoming the obstacles. Statistical community has provided plenty of sampling strategies which are generally believed also applicable in data mining. However, since data mining community has different starting-points and requirements from statistics community, some of these strategies may need to be reexamined when applied to data mining and it is also desirable to invent novel strategies for specific data mining tasks on specific data. This paper summarizes basic ideas and general considerations of sampling and categorizes sampling strategies existing in statistics so as to obtain potentially useful sampling strategies for data mining. Then the state-of-the-art ways of applying sampling in data mining are reviewed. By analyzing the strategies used in different data mining tasks and relating them to their precedents in statistics, we show that how traditional strategies are directly or indirectly applied. We discuss general considerations and research issues of sampling in data mining. We show that these issues are either usually not considered in statistics or not well-studied yet but essential to data mining. We believe extensive studies on sampling will contribute more to data mining, especially when dealing with large data sets.

Keywords: *sampling, data mining.*

1 Introduction

When studying the characteristics of a population, generally there are two approaches. The first one is to study every unit of the population, and such a process is statistically called a **census**. However, if the population is very large, this approach will be time-consuming or expensive thus often infeasible. In practice, therefore, as the other approach, we study the characteristics of the population by examining only a part of it. Such a technique is known as **sampling**, which is one of the most popular statistical techniques.

The theory of sampling, developed during the past decades, provides us with various scientific tools of drawing samples and making valid inferences about an interested population [22]. Although theoretically a complete census is more desirable, sampling can in effect demonstrate remarkable advantages such as reduced cost, reduced time, greater scope and greater accuracy over a complete census [9]. Therefore, sampling has been widely applied in almost every domain of the real world.

Data mining, also called knowledge discovery in databases (KDD), is a fast developing field of computer science and technology. It aims at discovering useful information from

[*]School of Computing, gubaohua@comp.nus.edu.sg
[†]Department of Statistics and Applied Probability, stahuff@math.nus.edu.sg
[‡]School of Computing, hliu@asu.edu

large collections of data. The discovered knowledge can be rules describing properties of the data, frequently occurring patterns, clusterings of the objects in the database, etc. Current technology makes it fairly easy to collect a large amount of data, but conducting analysis on these fast growing data tends to be slow and expensive [35]. A data set can be of large volume or high dimensional or both such that it will be either practically unnecessary or prohibitively expensive to mine on all of it. A natural and simple alternative way is to mine on a sample instead of all. This may work in that it directly reduces computational cost caused by large volume of data, but it may also return wrong results in that only partial data is used. Hopefully, sampling methodology, with its profound base on statistics, should be theoretically and empirically useful to improve performance while mitigating computational cost in data mining [15].

Statisticians have invented plenty of sampling strategies. It is generally believed that these still apply to data mining. However, due to different start-points and requirements of data mining and statistics, some of them need to be reinspected and novel strategies are also expected. In this paper, with data mining in focus, we first examine basic concepts underlying many sampling strategies and the rich reservoir of sampling strategies, then we survey the literature on how these sampling strategies are used to aid data mining. Our object is to obtain a better understanding of sampling from the views of both statistics and data mining, i.e., what sampling can do for data mining, how data mining demands new development in sampling. We also discuss general considerations and potential research issues for sampling in data mining, which are essential to data mining but not well studied so far. We believe that further research on these issues will facilitate data mining on large data sets.

The remainder of this report is organized as follows. In Section 2, we summarize basic terminology, basic ideas and general considerations of sampling in statistics. In Section 3, we categorize traditional statistical sampling strategies along some dimensions. In Section 4, we summarize sampling applications in data mining. We then discuss important research issues with respect to efficient sampling for data mining in Section 5. We propose general considerations when applying sampling to data mining in Section 6. We make conclusion in Section 7.

2 Traditional Statistical Sampling

2.1 Terminology

To acquire information about an unknown population, we can perform sampling on it. Statistically, a **population** is a set of **elements** about which we want to study. The population can be finite (or infinite), in the meaning that it contains a finite (or infinite) number of elements. **Sampling** is to select a subset, so called a **sample**, of a studied population and estimate some interested characteristics about the population, which usually are not available to us or hard (e.g., highly expensive or time consuming) to obtain.

Before selecting a sample, we need to do two things. First, the population must be divided into parts, which are called **sampling units** and must cover the whole of the population without overlapping, in the sense that every element in the population belongs to one and only one unit. The construction of a list of sampling units is called a **sampling frame**. Second, we design a scheme to select elements in the sample, i.e., a mechanism that can determine whether an element will be accepted or rejected as a member of a sample. The number of elements to be selected in the sample, called **sample size**, usually also need to be considered within a scheme. Such a scheme will be called **sampling strategy** thereafter in this paper, but it may have other names elsewhere perhaps with slightly different meaning, for example, **sampling method**.

After the sample is obtained, it is used to provide some statistical informations of the population, which are often called **estimators** and expressed in constructed **statistics**, which are functions of the elements in a sample. The estimation procedure usually needs techniques of **statistical inference**. It is necessary to select a sample carefully to reach higher accuracy of such estimators but with less complexity and less cost. To get good estimations with the sample, it is also necessary to pick out the relevant aspects of the data, to decide whether to use auxiliary information in estimation, and to choose the form of an estimator. The accuracy or confidence associated with an estimator is usually expressed with a **confidence interval**. Statisticians design various sampling strategies for various purposes. Together with each sampling strategy, they also provide many estimation methods, often in forms of theorems or formulae for various characteristics of a population, for example, the mean (average) value of certain characteristic of a population.

2.2 Underlying Assumptions

There are some basic assumptions or ideas underlying sampling.

1. The interested characteristics of a population are usually not available to us or hard to get, whereas the interested characteristics of its sample are much easier to obtain. This is the basic motivation that we need sampling.
2. A sample is always a subset selected from a population. However, an arbitrary selection of a subset from a population is hardly accepted as a "sampling strategy" by statisticians. In other words, an acceptable sampling strategy should be well-designed to give better estimation than an under-designed selection. This is why sampling is called a technique.
3. With a careful design of sampling and using a suitable estimation approach, one can obtain estimates that are "unbiased" for population quantities, such as mean value or total value of a target characteristic of the population, without relying on any assumptions about the population itself. This is why sampling can often work.
4. If the sample size were expanded until all units of the population were included in the sample, then the population characteristic of interest would be known exactly. Any uncertainty in estimates obtained by sampling thus stems from the fact that only part of the population is observed. This is why there are errors inherent to sampling.
5. With the population characteristics remain fixed, the estimate of them depends on which sample is selected. If for every possible sample the estimate is quite close to the true value of the population characteristic, then there is little uncertainty associated with the sampling strategy; such a strategy is considered desirable. If on the other hand the value of the estimate varies greatly from one possible sample to another, then uncertainty is associated with the strategy. This is why different sampling strategies may produce different estimations about the same population and why we should adopt a suitable one for a specified purpose.

2.3 General Considerations

2.3.1 Sampling Unit

Before sampling, one need to decide what is a unit of a specific population to make samples. This may not be an easy task. With many populations of people and institutions, it is straightforward to identify the type of units to be samples and to conceive a list or frame of the units in the population. However, in practice, it is sometimes difficult to obtain a list that corresponds exactly to the population of interest. In some other populations, it is not so clear what the units should be. Of course, one is free to choose alternative sizes and shapes of units, but such choices may affect the cost of the survey and the precision of estimators. Although the sampling units may be hard to decide, basic sampling design considerations regarding random sampling, stratified sampling, and other designs, and estimation results on design-independent estimation, ratio estimation, and other strategies still apply [55].

2.3.2 Sample Size

A suitable sample size depends on cost, accuracy and some other considerations. Generally, a sample size can be determined so that the estimates of the true value of a population does not differ by more than a stated margin error in more than δ of the cases. By setting up a probability inequality: $P(|e - e_0| \geq \epsilon) \leq \delta$, we solve for the sample size n for a given value ϵ and δ, where e stands for an estimate from the sample, which is generally a function of the sample size n, e_0 stands for the true value of the population, ϵ is called *confidence limit*, and $1 - \delta$ is called *confidence level*.

However, e_0 is usually unknown either. In this case, a practical way to determine the required sample size can be done as follow [52]: in the first step, we select a small preliminary sample of size m. Observations made on the units selected in this sample, will be used to estimate e_0 involved in the above expression. After replacing the e_0 by the estimate just obtained from the preliminary sample, the equation is then solved for n, the required sample size. If $n \geq m$, then $n - m$ additional units are selected in the final sample. If $n \leq m$, then no more units are selected and preliminary sample is taken as the final sample.

2.3.3 Unbiased Estimator, Sampling Variance and Confidence Intervals

The ultimate objective of any sampling is to make inferences about a population of interest. For this end, estimators are constructed based on a sample to evaluate certain unknown features of the population. The goodness of any estimate made from a sample depends both on the sampling strategy by which the sample is obtained and on the method by which the estimate is calculated from the sample data. A good estimator is expected to have two main properties. One is known as *unbiasedness*. The other is small *sampling variance*. An estimator e is called to be *unbiased* for the actual but unknown value e_0, if the expected value of e is equal to e_0, i.e., $E(e) = e_0$. Thus unbiasedness of an estimator ensures that on the average, e will take a value equal to e_0, although for most of the samples, the values taken by e will be more or less than e_0. Sampling variance is defined to measure the divergence of the estimator e from its expected value e_0, i.e., $V(e) = E[e - E(e)]^2$. With small sampling variance, we are ensured that the estimators differ little from sample to sample as well as from the true value e_0 . To enhance the usefulness of estimation, we can also calculate a specific region $(e_0 - \epsilon, e_0 + \epsilon)$, known as *confidence interval*, in which the true value of interested parameter e_0 lies with a probability $1 - \delta$ (i.e., the *confidence level*), where the δ satisfies $P(|e - e_0| \geq \epsilon) = \delta$.

2.3.4 Sampling and Non-sampling Errors

In sampling theory, it is usually assumed that the variable of interest is measured on every unit in the sample without error, so that errors in the estimates occur only because just part of the population is included in the sample. Such errors are called **sampling errors**. But in real world situations, **non-sampling errors** may also arise due to defective sampling procedures, ambiguity in definitions, faulty measurement techniques and so forth. Sampling error is inherent and unavoidable in any and every sampling scheme. This error, however, has the favorable characteristic of being controllable through the size and design of the sample, and shall theoretically become nonexistent in case of complete enumeration. In many situations, the decrease is inversely proportional to the square root of sample size [52]. Unlike sampling errors, the non-sampling errors are likely to increase with increase in sample size. It is quite possible that non-sampling errors in a complete enumeration survey are greater than both the sampling and non-sampling errors taken together in a sample survey. One should, therefore, be careful in evaluating and checking the processing of the sample data right from its collection to its analysis in order to minimize the occurrence of non-sampling errors.

2.3.5 Sampling Strategy Implementation

Sampling strategy refers to a plan to form a sample. By implementation of a sampling strategy, we mean a procedure of selecting a sample according to it. In practice, this may not always be an easy task. One general procedure, which applies to any arbitrary sampling strategy, is to list all the samples and the corresponding probabilities, and then to select one of these samples by a mechanism which guarantees the probability of selection. Once the listing is complete, this task is usually accomplished with the help of a table of random numbers generated by a cumulative total strategy [9]. At this stage, if necessary, the stated probabilities are reduced to rational fractions of a sufficient degree of accuracy.

Operationally, however, such a procedure becomes very inconvenient in most cases. For example, we want to take 3 elements from a set Q of 20 elements, clearly there are 1140 (the number of combinations of any 3 elements in the 20 elements) potential samples each with equal probability in such a design. A complete listing of these samples would itself be prohibitive. On the other hand, since any sample is, in effect, an ordered collection of some units of the population, it would be desirable to have some sort of mechanism which would generate the units, one by one, of different samples (according to certain chance rules) such that the overall chance of selecting a sample S would coincide with the given probability. In other words, given a sampling strategy, we would like to discover an underlying scheme of sampling that would dictate the operational step involved in generating the units of all samples one by one in a way that would eventually result in the given sampling design. One of such operable mechanisms, named *unit drawing chance mechanism*, was introduced in [22].

3 Categories of Sampling Strategies

There have been many sampling strategies developed by statisticians. Each sampling strategy has its own characters. Some sampling strategies are applied to **general purposes**, while some others are only used for a **specific domain**. Some strategies treat each unit of a population with **equal probability**, while some with **varying probability**. Some sampling can be done in **one stage**, while some must be done in **multi-stage**. Many sampling are based on **probability theory**, while a few do not rely on **probability theory**. Some sampling procedures work in a **non-adaptive** way (i.e, its scheme including sample size be determined before sampling and does not change during sampling), while some others use **adaptive** strategy by changing its scheme according to the result obtained so far from the sampling. Each sampling strategy possesses its advantages and disadvantages. Appropriate categorization of all these sampling strategies may help to grasp their basic ideas. In this section, we categorize these sampling strategies by the above-mentioned dimensions.

Sampling strategies in this paper are only briefly described due to space limit. Details can be found in many statistical textbooks that discuss sampling strategies. Some of them are listed here [39, 11, 61, 54, 27, 22, 14, 55, 53, 51, 56, 52].

3.1 General-purpose vs. Domain-specific

3.1.1 General-purpose sampling strategies

This group of sampling strategies refers to those most commonly and widely used strategies, including random sampling, stratified sampling, cluster sampling, systematic sampling, double sampling, network sampling, and inverse sampling.

Simple Random Sampling

This is a sampling strategy by which every population unit has the same chance of being selected in the sample. It has two variants: random sampling without replacement and random sampling with replacement.

Random sampling without replacement is the most popular sampling design in which n distinct units are selected from the N units in the population in such a way that every possible combination of n units is equally likely to be selected. Other sampling strategies may give each unit equal probability of being included in the sample, but only with this strategy does each possible sample of n units have the same probability. The sample may be obtained through n selections in which at each step every unit of the population not already selected has an equal chance to be selected. The selections are more commonly made by using a random number table or a computer "pseudo random number" generator. The most prominent advantage of using random sampling is that it does not introduce any bias in selecting the sample. Another advantage is that it is easy to operate.

Random sampling with replacement is a sampling design in which a sample of n units selected from a population of N units by such a procedure of a series of draws that at any draw, all N units of the population are given an equal chance of being draw, no matter how often they already been drawn, i.e., any of the units may be selected more than once. One practical advantage of sampling with replacement is that, in some situations, it is an important convenience not to have to determine whether any unit in the population is included more than once. However, for a given sample size, simple random sampling with replacement is inherently less efficient than simple random sampling without replacement [55].

Both the two strategies are widely accepted, however, they may not apply when the objective is to sample some rare units in a population, because a random sample of the population may not include any information about the objective units.

Stratified Sampling

Stratified sampling strategy forms an interesting and useful subclass of sampling strategy. Such a strategy combines the advantages of purposive (non-probabilistic) and probabilistic sampling together. To make a stratified sample, we divide the population Q of N units arbitrarily or purposively into L ($L \geq 2$) non-overlapping sub-populations or strata. Then smaller samples could be selected from these different strata independently of one another. At last the total sample is formed by combining all the small samples. The strata are generally purposively formed by utilizing available relevant auxiliary information. In order to obtain high precision of estimates, the units within a stratum are required to be as similar as possible whereas differ as much as possible between strata. Then, even though one stratum may differ markedly from another, a stratified sample with the desired number of units from each stratum in the population will tend to be "representative" of the population as a whole. A commonly used design is called *Stratified Random Sampling* in which the design within each stratum is *simple random sampling*.

The main advantages that stratified sampling over unstratified sampling are: (1) a better cross section of the population is represented in the sample thus it could be an alternative way apart from increasing the sample size to reach higher precision of estimates and in fact it normally provides more efficient estimates than unstratified sampling; (2) it enables effective utilization of the available auxiliary information, because the stratification makes it possible to use different sampling strategies in different strata; (3) by grouping extreme values in a separate stratum, it can reduce the variability within other strata thus it can deal with "outliers". Its main disadvantage is that sometimes it is difficult in term of time and cost to construct ideal strata [52].

Cluster Sampling

In many cases, there is no reliable list of all elements in the population available, or it would be prohibitively expensive to construct such a list. In other cases, even when such a list is available, it would not be an economic choice to use elements as a sampling unit. All these cases point us to another choice of a larger sampling unit, **cluster sampling**. In cluster sampling, the population is firstly separated into mutually exclusive sub-populations called "clusters". Then a sample (often a random sample) of these clusters is chosen. All units in the selected clusters are selected to form the sample. Unlike stratified sampling, the units in each cluster are desired to be as much heterogeneous as possible and all clusters are similar to each other to obtain estimators of low variance. An ideal cluster contains the full diversity of the population and hence is "representative".

The formation of clusters varies in different applications. The cluster may refer to naturally occurring groups such as individuals in an organization; clusters may also be formed artificially by grouping together units that are neighboring or can be accessed together conveniently. When the total geographical area of the population is subdivided into smaller areas and a sample of areas is taken, the sampling design is known as *area sampling*. Also the population can be partitioned into primary units, each primary unit being composed of secondary units. Whenever a primary unit is included in the sample, the values of interested attributes of every secondary unit within it are observed. In cluster sampling, a primary unit consists of a cluster of secondary units, usually in close proximity to each other.

The advantage of cluster sampling is that it is often less costly to sample a collection of clusters than to sample an equal number of units from the population. However, it may be hard to ensure the "representativeness" of the sample since it almost inevitably introduces biasness when forming clusters.

Systematic Sampling

One strategy of selecting n units from a population of N units is to select first a random number between 1 and k and then select the unit with this serial number and every kth unit afterwards. This procedure is known as "systematic sampling" and ensures that each unit has the same chance of being included in the sample, the constant k is known as the "sampling interval" and is generally taken as the integer nearest to N/n, the inverse of the sampling fraction. There are also several variants of systematic sampling [27].

This strategy has two main advantages. First, it is operationally convenient. Second, systematic samples are well spread over the population, thus there is less risk that any large contiguous part of the population will be left unrepresented. Therefore, this strategy can sometimes provide more precise estimates than a simple random sampling and a stratified random sampling.

The main disadvantages of the strategy are: first, variance estimators cannot be obtained from a single systematic sample; second, a bad arrangement of the units (such as periodicity of the units in the population) may produce a very inefficient even misleading sample.

Double Sampling

Double sampling, also called two-phase sampling (i.e., sampling is done in two phases), refers to a design in which initially a sample of units is selected from a large first-phase sample for obtaining auxiliary information only, and then a second-phase sample is selected in which the studied variable is observed in addition to the auxiliary information. The second-phase sample is often selected as a subsample of the first.

The purpose of double sampling is to obtain better estimators by using the relationship between auxiliary variables and the studied variable. In double sampling for stratification, the units in the initial sample are classified into strata. The second-phase sample is then selected by stratified sampling from the initial sample. An important special use of this strategy is adjustment for non-response in surveys. This technique is profitable only if the gain in precision from estimates or stratification is more than the loss in precision due to the reduction in the size of the second-phase also the final sample [9]. When the sampling procedure is completed in three or more phases, it is termed *multi-phase sampling*.

Network Sampling

In network sampling (sometimes called *multiplicity sampling*), a simple random selection or stratified random selection of units is made, and all units which are linked to any of the selected units are also included in the sample. Defining a network to be a set of observation units with a given linkage pattern, a network may be linked with more than one selection unit, and a single selected unit may be linked with more than one network. If the population of selection units is stratified, a network may also intersect more than one stratum [55]. The network sampling design is first used not to increase efficiency, just because it unavoidably arises in some sampling situations (e.g., an individual has records at more than one organization). Later researchers on the subject recognize its potential for giving lower variance estimates than conventional procedures and for increasing the "output" of a survey.

Inverse Sampling

In practice, an attribute under consideration may arise with rare frequency. In such cases, even a large sample may not give enough information to estimate the attribute. A proper sampling strategy for such situations is known as *inverse sampling*, in which sampling is continued until some specified conditions are satisfied, which often depend on the results of the sampling, have been fulfilled. For instance, when the proportion of the number of units possessing an attribute A is very small, an estimate of A obtained from even a very large (simple random) sample may not be precise. With inverse sampling for an attribute, the sample size n is not fixed in advance, but sampling is continued until a predetermined number n' possessing the attribute is reached. The procedure is called inverse sampling for proportions [53]. By this strategy, we can obtain enough information about rare-occurring attribute by setting a suitable n'. However, it may also be difficult to know the exact n' beforehand, i.e., how much is enough for estimating the unknown attribute.

3.1.2 Domain-specific sampling strategies

Sampling strategies in this category are highly related to their respective domains, i.e., they are purposively designed for certain specific domain. Generally, these strategies with specific sample scheme and estimators may not be applied in other domains. However, their ideas may be useful to other domains.

Distance Sampling

Distance sampling is used to estimate density or abundance of biological populations. *Line transect* and *point transect* are two main approaches to estimation of density. In distance sampling, a set of randomly placed lines or points is established and distances are measured to those objects detected by traveling along the lines or surveying the points.

Unbiased estimates of density can be made from these distance data if certain assumptions are met [3].

Spatial Sampling

In geological studies it is frequently desired to predict the amount of ore or fossil fuel that will be found at a site. The prediction may be based on values observed at other sites in the region, and these other sites may be irregularly spaced in the region. In spatial sampling, one can use the information in a covariance function to determine what selection of sample sites will give the best prediction, that is, what sampling scheme is most effective. Given that observations have been observed at a sample of n sites, and one wish to predict what will be found at another site, the covariance function can be computed by using these observations [55] and [27].

Capture-Recapture Sampling

Capture-recapture sampling is used to estimate the total number of individuals in a population. An initial sample is obtained and the individuals in that sample are marked or identified. A second sample is independently obtained, and it is noted how many of the individuals in that sample are marked before. If the second sample is representative of the population as a whole, then the sample proportion of marked individuals should be about the same as the population proportion of marked individuals. From this relationship, the total number of individuals in the population can be estimated. Capture-recapture strategies have been used to estimate the abundance of animal population (such as bird, insect) and elusive human populations (such as the homeless)[27].

Line-Intercept Sampling

Line-intercept sampling is also used to estimate elusive populations such as rare animals. A sample is formed by the way that a sample of lines is selected in a study area and whenever an object of the population is intersected by one or more of the sample lines, a variable of interest associated with that object is recorded. With the design, a large bush has a higher probability of inclusion in the sample than does a small bush. Unbiased estimation of the population quantities depends on the ways to determine these probabilities [55].

Composite Sampling

Composite sampling is used in ecological studies. A sample is formed by taking a number of individual samples and then physically mixing them to form a composite sample. The goal is to obtain the desired information in the original samples but at reduced cost or effort. It can be used for detection of a certain trait, compliance, reduction of variance in the presence of inexact analytical procedures and so on [27].

Panel Sampling

This strategy may be used in social surveys. Sampling inquires are often carried out at successive intervals of time on a continued basis covering the same population. When the same sample of respondents in a mail, personal interview is selected to provide information on more than one occasion, the sampling plan is termed panel sampling [53].

Monte Carlo Strategies

These series of strategies are successfully used in Bayesian Inference. The aims of Monte Carlo strategies are to solve one or both of the following problems: (1) to generate samples from a given probability distribution $P(x)$; (2) to estimate expectations of functions under this distribution. Common Monte Carlo strategies are: Importance Sampling, Rejection Sampling, Metropolis strategy, and Gibbs Sampling.

Importance Sampling is basically not for generating samples from $P(x)$ but for estimating the expectation of a function. *Rejection Sampling* generate a sample of the complicated object distribution by using a simpler proposal distribution $Q(x)$. It randomly generates v first, and then decides if v is accepted into the sample set by comparing two values related to $P(v)$ and $Q(v)$. *Metropolis strategy* makes use of a proposal density $Q(x)$ which depend on the current state $x(t)$. By computing a function value b of $P(x)$ and $Q(x)$ at each new state x', it can make a decision by comparing b with 1. If $b \geq 1$, then the new state x' is accepted. Otherwise, the new state is accepted with probability b. *Gibbs Sampling* is a strategy for sampling from distributions over at least two dimensions. It can be viewed as a Metropolis strategy in which the proposal distribution Q is defined in terms of the conditional distributions of the joint distribution $P(x)$. It is assumed that $P(x)$ is too complex to draw samples from directly, its conditional distributions are tractable to work with [34, 17].

Shannon Sampling

Shannon Sampling is named after the famous Shannon sampling theory and used in signal process. Given a sampling rate, Shannon's theorem gives the cutoff frequency of the pre-filter h. The original signal $x(t)$ changes into band-limited signal by computing $x(t) * h$ and then is transformed into $x(n)$ by a discretization process in which impulse train is made. After the Shannon sampling, the original signal continuous $x(t)$ is sampled as discrete $x(n)$ which can be passed through an ideal low-pass filter and then can be reconstructed after the transfer [23].

3.2 Probability vs. Non-probability

Scientifically, a survey is called a "probability sampling" when the strategy of its selection is based on the theory of probability. In order to have a randomly selected sample, for example, you must set up a procedure that assures that the different units in your population have equal probabilities of being chosen. However, sampling can also be done without the theory of probability, i.e., **non-probability sampling**.

The difference between non-probabilistic and probabilistic sampling is that non-probabilistic sampling does not involve selections according to probability but probabilistic sampling does. It does not necessarily mean that non-probabilistic samples are not representative of the population. But it does mean that non-probabilistic samples do not depend upon the rationale of probability theory. With a probabilistic sample, we know that the probabilities that we have represent the population well. We are able to estimate confidence intervals for the statistic. With non-probabilistic samples, we may or may not represent the population well, and it will often be hard for us to know how well we have done so. In general, researchers prefer probabilistic strategies to non-probabilistic ones, and consider them to be more accurate and rigorous. However, in social researches there may be circumstances where it is not feasible, practical or theoretically sensible to do probability sampling. In these cases, a wide range of non-probabilistic alternatives listed below may work better [58].

Accidental Sampling

Also called *Haphazard* or *Convenience Sampling*, this strategy refers to making a sample by accidental encountering or selection. For instance, interviewing a man on the street like interviews conducted frequently by television news programs to get a quick (although non-representative) reading of public opinion. In clinical practice, we might use clients who are available to us as our sample. In many research contexts, we sample simply by asking for volunteers. Clearly, the problem with all of these types of samples is that we have no evidence that they are representative of the populations we are interested in generalizing to – and in many cases we would clearly suspect that they are not.

Purposive Sampling

In purposive sampling, we sample with a purpose in mind. We usually would have one or more specific predefined groups we are seeking. For instance, have you ever run into people in a mall or on the street who are carrying a clipboard and who are stopping various people and asking if they could interview them? Purposive sampling can be very useful for situations where you need to reach a targeted sample quickly and where sampling for proportionality is not the primary concern. With a purposive sample, you are likely to get the opinions of your target population, but you are also likely to overweight subgroups in your population that are more readily accessible. All of the strategies that follow can be considered sub-categories of purposive sampling strategies.

Modal Instance Sampling

In statistics, a *mode* is a value most frequently occurring in a distribution. In sampling, when we do a modal instance sample, we are sampling the most frequent cases, or the typical cases. Basically, modal instance sampling is only sensible for informal sampling contexts.

Expert Sampling

Expert sampling involves the assembling of a sample of persons with known or demonstrable experience and expertise in some area. Often, we convene such a sample under the auspices of a panel of experts. The advantage of doing this is that we are not out on our own trying to defend our decisions – we have some acknowledged experts to back us. The disadvantage is that even the experts can be, and often are, wrong.

Quota Sampling

In quota sampling, we select people non-randomly according to some fixed quota. There are two types of quota sampling: proportional and non proportional. In proportional quota sampling we want to represent the major characteristics of the population by sampling a proportional amount of each. Non-proportional quota sampling is a bit less restrictive. In this strategy, we specify the minimum number of sampled units we want in each category. Here, we are not concerned with having numbers that match the proportions in the population. Instead, we simply want to have enough to assure that we will be able to talk about even small groups in the population. This strategy is the non-probabilistic analogue of stratified random sampling in that it is typically used to assure that smaller groups are adequately represented in our sample.

Heterogeneity Sampling

We sample for heterogeneity when we want to include all opinions or views, and we aren't concerned about representing these views proportionately. Another term for this is sampling for diversity. In effect, what we would like to be sampled is not people, but ideas. We imagine that there is a universe of all possible ideas relevant to some topic and that we want to sample this population, not the population of people who have the ideas. Clearly, in order to get all of the ideas, and especially the "outlier" or unusual ones, we have to include a broad and diverse range of participants. Heterogeneity sampling is, in this sense, almost the opposite of *modal instance sampling*.

Snowball Sampling

Snowball Sampling can be seen as the counterpart of network sampling in probability sampling. In snowball sampling, we begin by identifying someone who meets the criteria for inclusion in our study. We then ask them to recommend others who they may know who also meet the criteria. Although this strategy would hardly lead to representative samples, there are times when it may be the best strategy available. Snowball sampling is especially useful when we are trying to reach populations that are inaccessible or hard to find.

As a short summary, we can group these nonprobability sampling methods into two broad types: accidental or purposive. Most of sampling strategies just mentioned are purposive in nature because we usually handle the sampling problem with a specific plan in mind, while the distinctions among them just lie in the different plan.

3.3 One-stage vs. Multi-stage

The strategies we have covered so far, such as random, stratified, systematic and cluster sampling, are the simplest sampling strategies that can be done in one-stage. In many real applications, we would encounter situations that are considerably too complicated to do sampling in one stage. We can combine the one-stage strategies in a variety of useful ways that help us address our sampling needs in the most efficient and effective manner possible. When we combine sampling strategies, we call this **multi-stage sampling** or sometimes **sub-sampling**. In multi-stage sampling, the sample is selected in stages. In other words, the population is divided into a number of first-stage (or primary) units, which are sampled; then the selected first-stage units are sub-divided into a number of smaller second-stage (or secondary) units, which are again sampled: the process is continued until the ultimate sampling units are reached.

Multi-stage sampling differs from the multi-phase sampling mentioned earlier in the sense that in the former, a sampling frame of the next stage units is necessary only for the sample units selected at that stage, whereas the latter requires a complete sampling frame of the ultimate sampling units. Also, the sampling unit in the case of multi-phase sampling remains the same at each phase of sampling, whereas it changes in the case of multi-stage sampling.

Multi-stage sampling is used in a number of situations: (1) Sampling frames may not be available for all the ultimate observational units in the population, and it is extremely laborious and expensive to prepare such a complete frame. (2) Even when suitable sampling frames for the ultimate units are available for the population, a multi-stage sampling plan may be more convenient than a one-stage sampling. (3) It can reduce response errors and improve sampling efficiency by reducing the intra-class correlation coefficient observed in natural sampling units, such as households or villages.

Some commonly used multi-stage sampling strategies are: (1) *Multi-stage Simple Random Sampling*, which makes simple random sampling at each stage; (2) *Multi-stage Varying Probability Sampling*, in which varying probability sampling is used at some stage; and (3) *Stratified Multi-stage Sampling*, in which a population is at first divided into some strata, then sampling is conducted in stages separately within each stratum.

3.4 Equal probability vs. Varying probability

With varying (or unequal) probabilities, different units in the population have different probabilities of being included in the sample. The differing inclusion probabilities may result from some inherent features of the sampling procedure, or they may be deliberately imposed in order to obtain better estimates by including "more important" units with higher probability. Such a strategy can be useful, for example, when units vary considerably in size, simple random sampling does not seem to be an appropriate procedure, since it does not take into account the possible importance of the size of the unit. Under such circumstances, selection of units with unequal probabilities may provide more efficient estimators than equal probability sampling. When the units are selected with probability proportional to a given measure of size, this type of sampling is known as *probability proportional to size sampling* (PPS). It may appear that PPS sampling would provide biased estimators as the larger units are over-represented in the sample. This would be so, for example, if the sample mean is used as an estimator of the population mean. However, if the sample observations are appropriately weighted at the estimation stage, taking into consideration their probabilities of selection, it is possible to obtain unbiased estimators. Therefore, the PPS sampling is expected to be more efficient than simple random sampling when size measure x is approximately proportional to the study variable y, and the line of regression of y on x passes through, or nearly through, the origin.

3.5 Adaptive vs. Non-adaptive

For many sampling strategies, the selection procedure does not depend in any way on observations made during the sampling, so the sample size can be set before sampling. But in some sampling situations, a researcher may feel inclined to make decisions during the sampling, based on what has been observed so far, as to which units or how many units to observe next. Adaptive sampling refers to sampling strategy in which the procedure for selecting sites or units to be included in the sample may depend on values of the variable of interest observed during the sampling.

The primary purpose of adaptive sampling is to take advantage of population characteristics to obtain more precise estimates of a population than using non-adaptive strategies. Adaptive sampling strategies offer the potential to give large increase in efficiency for some populations. However, adaptive selection procedures may introduce biases into conventional estimators, so unbiased estimators under the adaptive designs are needed.

Whether an adaptive sampling is more efficient or less efficient than a non-adaptive sampling such as simple random sampling depends on the type of population being sampled. For example, it is indicated in [55] that if the within-network variance of the population is sufficiently high, the adaptive cluster sampling listed below will be more efficient than simple random sampling.

Adaptive Cluster Sampling

Adaptive cluster sampling refers to strategies in which an initial set of units is selected by some probability sampling procedure, and, whenever the variable of interest of a selected unit satisfies a given condition, additional units in the neighborhood of that unit are added to the sample.

Systematic and Strip Adaptive Cluster Sampling

This strategy is basically a variant of the "adaptive cluster sampling". It differentiates itself from the latter in the formation of the initial sample. In an adaptive cluster sampling, an initial sample is selected with primary units and subsequent additions to the sample are made with secondary units. If each initial primary unit consists of a set of units evenly spaced in some arrangement throughout the population, the initial sample will be termed a "systematic" initial sample; if each initial primary unit consists of a row of units arranged in a straight line, the initial primary units will be called "strips". These kinds of adaptive cluster sampling are called "systematic adaptive cluster sampling" or "strip adaptive cluster sampling".

Stratified Adaptive Cluster Sampling

In stratified adaptive cluster sampling, an initial stratified sample is selected from a

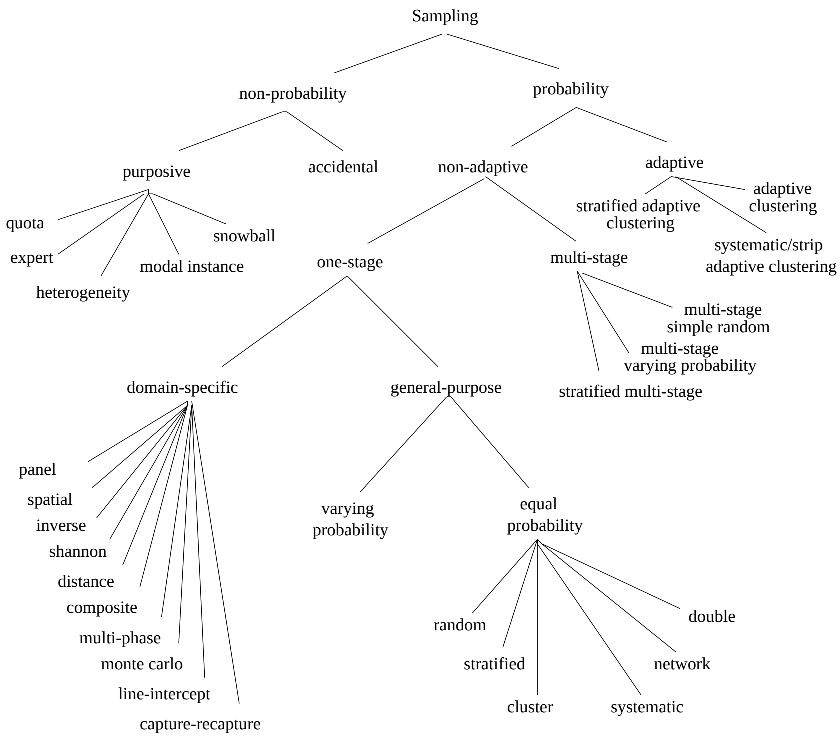

Figure 1: A Categorization Tree of Sampling Strategies

population and, whenever the value of the variable of interest for any unit is observed to satisfy a specified condition, additional units from the neighborhood of that unit are added to the sample. Still more units may be added to the sample if in turn any of the subsequently added units satisfies the condition. This strategy is important, from a practical point of view, because for many populations prior information exists on which an initial stratification can be based and yet the exact distribution or patterns of concentration of the population cannot be predicted.

3.6 Summary of sampling strategies

So far, we have described many sampling strategies, which we will call thereafter "traditional statistical sampling strategies", to differentiate them from those used in data mining to be introduced later. Although they are not exhaustive, the sampling strategies described in this paper, from basic to special, one-stage to multi-stage, probability to non-probability, adaptive to non-adaptive, constitute representatives of existing sampling strategies. To illustrate the relationship among these sampling strategies, we draw a decision-tree-like "sampling tree" in Figure 1. In the figure, each leaf node is a sampling strategy, each path from the root to a leaf shows a procedure to classify a given sampling strategy, while each internal node on such a path shows one criterion used to classify the sampling strategy.

Our categorization of sampling strategies is not the only one even by the standard used in the paper. The main motivation for this categorization is to provide a big picture and help users to view the relationships among these sampling strategies, and to offer some rough guideline to use them.

Although these sampling strategies may look very different, they have one thing in common, i.e., they all produce a subset of a target population. The differences among them lie in different procedures by which the subset is produced, and sometimes by what kind of accessory information is used when forming the sample. Our "sampling tree" can show their differences and relations.

Sometimes it may be difficult to choose a suitable sampling strategy because it depends on many factors such as research objective, permitted cost, time, and so on. Our categorization and the "sampling tree" can be used as a rough guideline to choose feasible sampling strategies from many candidate strategies. For example, random sampling due to its ease and unbiasedness, should be in the first choices. If we just want this kind of ease, systematic sampling may also be a good choice. If we care less about precision of estimates as computed in probability, we can choose non-probability strategies. If sampling in one stage involves many factors which may be complex and lead to high cost or long sampling time and sampling in stages may help, multi-stage sampling or multi-phase sampling should be considered. If the research is very domain-specific, then besides common sampling strategies, existing sampling strategies for that specific domain may work better.

Finally, it should be pointed out that having a sample in hand is not the end of a complete sampling procedure. To get estimations of the interested population is the very final objective of sampling. Due to length limitation, we deliberately omit them. We wish this would not give readers wrong impression that only a sampling strategy is important to sampling.

4 Sampling Techniques in Data Mining

As a fast developing field, data mining aims to extract significant patterns or interesting rules from data. Common tasks of data mining usually include **association rules**, **classification**, and **clustering**. These tasks are generally conducted on large data sets in which potential useful knowledge is hidden. A data set is called large if it has either a large number of instances or a large number of features, or both. A large size of instances will increase the time for fitting a possible model in model space, while a large size of features will increase the space of models. Large data sets can be great obstacles to data mining algorithms in that the process of pattern search and model fitting often requires many passes over the data, but it is often infeasible to load the whole data sets into physical memory. As large data sets are more and more common, the data mining community also challenges itself to develop data mining algorithm that can cope with large data sets [46]. However, a simple way of alleviating the problems caused by voluminous data (i.e., the number of instances) is to use sampling, i.e., to pick a subset of a large data set.

Two facts support the applicability of sampling in data mining. One is that some data mining algorithms may not use all data in a data set all the time during their runnings [12]. The other is that mining on part of data may produce comparable results as on the whole data [63, 57]. They also account for two basic ways of using sampling in data mining. One is embedding sampling into data mining algorithms, while the other is running sampling separately from data mining. However, using sampling may have a problem: the results could be wrong with a small probability, although it can give quite good approximation to the true results with a large probability. This is because just working on a subset of the whole data may distort the inherent functionally dependencies among attributes, which will be very complicated and elusive in high-dimension problems [35].

Once we decide to work on a sample instead of the whole data set, the next step is to choose the sample, where sampling strategies can play their roles. It is obvious that different sampling strategies will result in different samples from the same data set, while not all these samples will be suitable for a given data mining task. Therefore it is necessary to study how to apply sampling strategies to various data mining tasks. In fact, many sampling strategies have been employed or proposed by data mining community. In this section, we summarize the state-of-the-art of sampling strategies in data mining from two dimensions, subfields of data mining and sampling strategies, and provide a profile of sampling applications in data mining.

4.1 Sampling in Different Fields of Data Mining

The use of sampling in data mining is pervasive. Our summarization is limited within common data mining tasks: *Association Rules*, *Classification*, *Clustering*, and those used in scale-up of data mining algorithms. Because data mining often works with databases, we first review how is sampling used in Database Management Systems (DBMS).

4.1.1 Sampling for DBMS

In DBMS, one area where sampling is frequently employed is query optimization. Many researchers use sampling to estimate the size of query results [19, 18], the size of projections and joins [7], and the selectivity [42]. In [32], the problem of estimating the size of a database query is tackled by using adaptive sampling. Random sampling has been also used as a standard technique for constructing (approximate) histograms for query optimization [6]. Some research works try to make random sampling to be a standard operation in database management systems [42, 43, 41].

4.1.2 Sampling for Mining Association Rules

The task of mining association rules is usually performed in transactional or relational databases, to derive a set of strong association rules in the form of "$A_1 \wedge ... \wedge A_m \Rightarrow B_1 \wedge ... \wedge B_n$", where A_i (for $i \in 1, ..., m$) and B_j (for $j \in 1, ..., n$) are sets of attribute values, from the relevant data sets in a database [8]. Usually this task may require repeated scans through database, therefore it can result in huge amount of processing when working on a very large database. Many efficient algorithms can significantly improve the performance in both efficiency and accuracy for mining association rules [8]. However, sampling can be a direct and easy approach to improving efficiency when accuracy is not the sole concern.

The application of sampling to mining association rules is suggested by Toivonen [57]. The idea is to find frequent sets via sampling. Using a random sample to get approximate results, they work out bounds for sample sizes, given the desired accuracy of the results. They show that exact frequencies can be found efficiently, by analyzing first a random sample then the whole database. This approach requires only one full pass over the database, and two passes in the worst case.

Random Sampling of transactions in databases has also been empirically proven by other researchers [63, 28] to be an effective strategy for finding association rules. It is shown that sampling has some advantages here: (1) it is able to speed up the mining process by more than an order of magnitude in reducing I/O costs and drastically shrinking the number of transaction to be considered; (2) it makes it possible to load the sampled database in main memory; and (3) sampling can accurately represent the data patterns in the database with high confidence.

4.1.3 Sampling for Classification

Classification is a learning task that learns patterns or function approximations from raw data with existing classes, thus establishes classification rules whereby we can classify a new observation into one of the existing classes. Usually there are three kinds of approaches to classification: decision trees (e.g., ID3 [47] and C4.5 [48]), neural networks [33], and statistical approaches (e.g. discriminant analysis [38]). Sampling are found to be used in some of them. For example, "windowing" used in ID3 is in itself a variant of sampling (we will discuss it more later). Catlett [5] compares four strategies of extracting subsets (i.e., samples) to facilitate classification on large data sets. The four strategies include one random sampling and three non-random techniques, namely "compacting duplicates", "improved windowing" and "stratification". Compacting duplicates is shown being applicable in domains where identical training instances appear several times: such duplicates can be compacted into a single instance. Improved windowing, which chooses the best of several initial windows, is shown that it sometimes raises accuracy, but has problems in noisy domains. Stratification techniques are shown being applicable when the class values are not uniformly distributed in the training set: a stratified sample is taken, aiming to get the distribution as even as possible.

As an arising subfield of classification, text classification or text categorization is to assign free texts to predefined categories. Text classifiers are variants of normal classification tools, which can automatically or semi-automatically classify useful patterns in text. Unlike in normal classification tasks where the number of class labels usually is less than one hundred, classes or categories can be thousands in documents where text classifiers are running. Here, sampling strategies are important for both effectiveness and efficiency. That is, a training set should be sufficient enough for high accuracy of text classification while not too large for efficient computation. Sampling strategies such as "uncertainty sampling" and "relevance sampling" are successfully used [29, 31]. These two sampling strategies will be discussed more in the next section.

4.1.4 Sampling for Clustering

Given a data set, the objective of clustering is to find a convenient and valid organization of the data, not to establish rules for separating future data into classes [24]. Clustering sometimes is referred as unsupervised classification problem because it does not require a teacher to label each observation(element, data item, instance, or case). Sampling can serve different purposes in clustering. Some clustering algorithms need sampling to do initialization of clustering. For example, in [36], hierarchical agglomerative clustering (HAC) is used on a sample of the original training set to provide a good set of initial parameters for the use of the EM algorithm. One problem of such an initialization is that some clustering algorithms are sensitive to choice of samples [2]. Therefore, how to choose a suitable sampling strategy is an important issue.

When working on large data sets, sampling can also help to reduce the space of searching models (i.e., clusters). In order to obtain the best representatives of clusters, different sampling strategies can be used based on the nature of the data. Some clustering algorithms start from a random sample of the data set and build models from it. An iterative scheme using random sampling is given in [12], which exploits the fact that using a constructed model to classify the data scales linearly with the number of data points to be classified. Some clustering algorithms adopt non-random sampling when random samples would neglect small clusters. As a traditional non-random sampling strategy, stratified sampling is often useful [12]. Novel sampling strategies are being proposed to enhance efficient clustering. For example, *Density Biased Sampling* [44] performs under-sample in dense regions and over-sample in sparse regions.

4.1.5 Sampling for Scaling-Up Data Mining Algorithms

Scaling-up aims to make established data mining algorithms be capable to deal with large data sets efficiently. This could be very difficult sometimes due to the constraint that data mining algorithms has been initially designed for small data sets. One intuitively possible way is to use the idea of divide and conquer: splitting a large data set into small un-overlapping parts on which established algorithms can run without any changes, then merging the results from all small parts into the final result. Notice that this is almost equivalent to shift the difficulty to the merging step which may require very complicated processing to get correct models. So the overall complexity may not reduce. Another way is via sampling, i.e., running the established algorithms on samples instead of the whole data. It is supported by the fact that not all data in the database are of equal important to the model being constructed [2]. Therefore, if representative data can be obtained, then it may be possible to facilitate the merging step to get a correct model. This is why and where sampling strategies can help in clustering.

For example, to scale clustering algorithms to large databases, a general framework is proposed by Bradley and et al [2]. The idea is to iterate over random samples of the database and merge information computed over previous samples with information computed from the current sample. In this way, at most one scan of the database is required. Catlett [4] studies the active and tactical use of sampling to reduce complexity as learning algorithms process large data sets. He finds that by looking at subsets of examples (called peepholes) when searching for good split values for numeric attributes, the run-time of decision tree learners can be reduced substantially without sacrificing accuracy. As an alternative to sampling strategies that fix sample size beforehand, Domingo and et al [10] propose adaptive sampling methods for scaling up data mining algorithms, in which the sample size is not fixed a priori.

Despite these successful applications of sampling, the usefulness of sampling for scaling up data mining algorithms is inherently problematic. Just as Provost and Kolluri [46] suggest that, as the most common approach to cope with the feasibility of learning from very large data sets, sampling is a degenerate form of a data partitioning in a learning system. The question of scalability asks whether the algorithm can process large data sets efficiently, while building from them the best possible models. Therefore, if using sampling produces models with lower accuracy than otherwise, its usefulness for scaling up is in question. On the other hand, if using sampling produces equivalent (or better) models, then sampling is an effective scaling mechanism. Whether sampling will be effective depends on how dramatically the rate of increase in learning accuracy slows.

4.2 Existing sampling strategies in Data Mining

In this section, we review existing sampling in data mining from the view of the sampling strategies and try to connect them with traditional statistical sampling strategies introduced in Section 3.

Random Sampling

This strategy may be the most common strategy and in many cases it will work. Many algorithms initialize parameters by evaluating a random sample of data [36]. It has also been successfully used in query size estimation in databases [41], where sampling may assist the user by providing approximate answers. Such information can be used for statistical analysis of databases, where approximate answers would suffice. It may also be used to estimate selectivities or intermediate result sizes for query optimization. Research works on sampling from databases have been surveyed by Olken and et al [42, 43].

A random sample can be made by using a sorted list of random numbers, which is done off-line. However, sometimes on-line generation of a random sample is more desirable to obtain higher performance. Some work has focused on on-line sequential random sampling algorithms [1]. One efficient algorithm for sequential random sampling without replacement is presented in [60]. Kivinen and Mannila [26] give theoretical bounds for the sample size that is required for establishing (with a given maximum error probability) the truth of association rules that have been discovered from the sample.

Stratified Sampling

In some cases when random sampling does not work, a stratified sample is required and works better. In the SKICAT application [13], for example, uniform random sampling would simply defeat the entire purpose of clustering. When one is searching for new clusters, it is clear that random sampling should be avoided in that members of a minority class could completely disappear from any small (or not so small) sample. Catlett [4, 5] studies a variety of procedures for sampling instances from a large data set and compares empirically the results of using the different techniques. In particular, he compares the random sampling and stratified sampling. He concludes that stratified sampling is applicable when the class values are not uniformly distributed in the training sets, where examples of the minority classes are selected with a greater frequency in order to even out the distribution.

Windowing

Windowing is proposed by Quinlan [47] for the ID3 decision tree learning algorithm. Its goal is to reduce the complexity of a learning problem by identifying an appropriate subset of the original data, from which a decision tree of sufficient quality can be learned. For this purpose, it maintains a subset of the available data, the so-called **window**, which is used as the training set for the learning algorithm. The window is initialized with a small random sample of the available data, and the learning algorithm induces a tree from this sample. This tree is then tested on the remaining examples. If the quality of the learned theory is not sufficient, the window is adjusted, usually by adding more examples from the training data, and a new tree is learned. This process is repeated until a tree of sufficient quality has been found. Contrary to random sampling, which will retain the distribution of examples of the original training set in the sample, windowing purposefully tries to skew this example distribution by adding only examples that are misclassified by the current tree. Furnkranz [16] proposed an improvement to traditional windowing which they called "integrative windowing". They aim to solve a problem that will happen frequently when using windowing with a rule learning algorithm: good rules have to be discovered again and again in subsequent iterations of the windowing procedure.

Windowing is basically embedding sampling into the data mining algorithm and this reminds us of traditional "adaptive sampling" because it makes use of learning results of current sample to decide when to stop sampling and which units should be added to renew the sample. There are at least three motivations for studying windowing techniques: (1) Memory limitations: Almost all learning algorithms still require to have all training examples and all background knowledge in main memory. Although memory has become cheaper and bigger, data accumulates much faster and there certainly are datasets too big to fit into the main memory of conventional computer systems. (2) Efficiency gain: learning time usually increases (most often super-linearly) with the complexity of a learning problem. Reducing this complexity may be necessary to make a learning problem tractable. (3) Accuracy gain: It has been observed that windowing may also lead to an increase in predictive accuracy. A possible explanation for this phenomenon is that learning from a subset of examples may often result in a less over-fitting theory.

Progressive Sampling and Dynamic Sampling

Unlike traditional usage of random sampling on a training set that takes no advantage of knowledge of learning algorithm (called "static sampling"), John and Langley [25] present a different approach (called "dynamic sampling") that successively expands the learning window until an extrapolation of the learning curve (i.e., learning accuracy over sample size) does no longer promise significant gains for further enlargements of the window. In terms of running-time, the authors note that this technique can in general only gain efficiency for incremental algorithms.

In the recent work of Provost and et al [45], they suggest "progressive sampling". The basic idea is when sample size increases the learning accuracy will follow the "learning curve" so that there should be a convergence point from which the learning accuracy will no longer increase greatly no matter how large the sample size be. By using this strategy, one can find the optimal size of a sample from which a learning algorithm can reach a given accuracy of learning. They conclude that, in a wide variety of realistic circumstances, progressive sampling is preferable to analyzing all instances from a database, even when optimal sample size being known in advance. Because in these two strategies, the final sample size is decided during sampling, it can be viewed as a counterpart of "inverse sampling" as mentioned in Section 3.

Uncertainty Sampling and Relevance Sampling

In text classification [62], random sampling will usually not be effective. For example, if only 1 phrase in 1000 phrases belongs to a desired class, and only 500 phrases can be labeled, then a random sample will usually contain 500 negative examples and no positive ones. This will not help train a classifier to distinguish positive from negative examples. Uncertainty sampling and relevance sampling are two of a number of possible active example selection approaches to choose training data. Both are far superior to random sampling when only a fraction of a data set can be labeled for training classification rules, due to resource limitation. In a sense, these two strategies are much like "purposive sampling" we mentioned in Section 3.

Relevance sampling uses feedback, thus it is a kind of nonrandom sampling. In effect, users are asked to label those texts that the current classifier considers very likely to be class members. This approach, is a reasonable strategy in a text retrieval context, where the user is more interested in seeing relevant texts than in the effectiveness of the final classifier produced. Relevance feedback has also been proposed for finding examples of unusual word senses. However, relevance feedback has many problems as an approach to sampling. It works more poorly as the classifier improves, and is susceptible to selecting redundant examples. Relevance sampling is also a sequential approach, since the labeling of earlier examples influences the selection of later ones.

Uncertainty sampling, an alternative sequential approach, is motivated by results in computational learning theory. It is an iterative process of manual labeling of examples, a classifier is built from those examples, and the classifier is used to select new examples whose class membership is unclear. In the paper of Lewis and et al [31, 30, 29], they showed how a probabilistic classifier can be trained using uncertainty sampling. They used uncertainty sampling as the strategy of training on misclassified instances. The difference is that when data is not labeled one must use the classifier itself to guess which examples are being misclassified. Note that the initial classifier plays an important role, since without it there may be a long period of random sampling before examples of a low frequency class are stumbled upon. Uncertainty sampling has been proved by experiments to have an advantage over relevance sampling and random sampling.

Resampling

Resampling means that after initial estimators are made from a former sample, new units are added to the sample to revise the estimators. Windowing uses resampling by adding only misclassified examples. But doing so is likely to increase the noise level inside the window. To avoid this, in the work of Furnkranz etal [16], they form a set of candidates containing all examples that are not yet in the window or covered by insignificant rules, plus all uncovered positive examples. The algorithm then selects a number of these candidate examples and adds them to the window. It sticks to adding uncovered positive examples only, because after more and more rules have been discovered, the proportion of positive examples in the remaining training set will considerably decrease, so that the chances of randomly picking a positive example from the set of all uncovered examples would decrease, which in turn might slow down the learner. Although adding only positive uncovered examples may increase the chances of learning over-generalized rules, these will be discovered by the second part of some given criterion and appropriate counter-examples will eventually be added to the window. Again, we can see the basic idea of "purposive sampling" here.

Sub-Sampling

This strategy is used for increasing the accuracy of a classifier. The basic idea is to train classifiers on multiple subsamples of the data and combine their predictions, usually by voting. Such approaches include **bagging** (acronym for **b**ootstrap **agg**regat**ing**) [16], where all examples are sub-samples with equal probabilities, and **boosting** [49], where examples that have been misclassified in previous iterations are more likely to be selected in the next iterations. Interestingly, some have noted that these techniques rely on unstable base learners, while they conjecture that the better performance of windowing with rule learning algorithms is due to moving stability. While the main focus of these works is to improve the accuracy of a given learning algorithm, it would be interesting to evaluate bagging and boosting techniques along their run-time and memory requirements as well.

Partitioning, as another approach of this sub-sampling strategy [16], splits the example space into segments of equal size and combines the rules learned on each partition. This technique produces promising results in noisy domains, but has substantially decreased learning accuracy in non-noisy domains. Besides, the technique seems to be tailored to a specific learning algorithm and not applicable to a wider group of rule learning algorithms. Here sub-sampling uses a similar idea as "multi-stage cluster sampling" in the sense that some sub-samples are made and worked on in steps to form a sample.

Similarity-Driven Sampling

In the work of Thomas Reinartz [50], he proposes a novel similarity-driven sampling approach with two preparation steps, sorting and stratification, and reuses an improved variant of leader clustering. Experimental results show that this strategy often outperforms statistical sampling techniques in terms of error rates using smaller samples. Leader clustering simultaneously generates partitions of data and selects leaders within each cluster by a single pass through the data. Following up leader clustering for data reduction by selecting leaders as representative prototypes without explicitly creating corresponding clusters, he designs a strategy, called "leader sampling", that works as follows. It selects the 1st instance as the 1st "prototype" of the 1st cluster. For each of following instances, leader sampling considers similarities between this instance and already selected prototypes. If the similarity between the instance and the most similar prototype exceeds a pre-defined threshold δ, which means that the two are not similar, it proceeds with the next instance. Otherwise, leader sampling adds the current instance to the sample. Thus the procedure of sampling is adaptively adjusted based on learned information from the current sample. Again, we see the idea of "adaptive sampling", and the concept of similarity here is much like the "linkage" of "network sampling".

Density Biased Sampling

As uniform sampling often tends to miss small classes or clusters, Palmer and Faloutsos [44] propose a novel sampling to probabilistically under-sample dense regions and over-sample sparse regions. A weighted sample is used to preserve the densities of the original data. It will naturally includes uniform sampling as a special case. A memory efficient algorithm is also proposed that approximates density biased sampling using only a single scan of the data. Experiments show that this method scales linearly and out-performs uniform samples when clustering realistic data sets. This idea combines that of "varying probability sampling" and "inverse sampling" in statistics to treat rare classes more carefully than ordinary classes.

4.3 Features of Sampling in Data Mining

So far, we introduced a number of sampling strategies used in various data mining tasks. We summarize them in Table 2 by data mining tasks where they have been used. Basically they are used in two ways. One is being embedded with data mining algorithms, the other is running separately before data mining. We notice that: (1) many but not all statistical sampling strategies introduced in Section 3 are used in data mining; (2) different strategies are chosen mainly according to different targets of data mining tasks and different nature of data sets; (3) new sampling strategies are being introduced for new applications such as text classification, boosting, and so on.

Table 1: Summary of sampling strategies in data mining

Sampling Strategies	Data Mining Tasks			
	Classification	Clustering	Association Rules	Scale-up
Random Sampling	$\surd$	$\surd$	$\surd$	$\surd$
Stratified Sampling	$\surd$			
Windowing	$\surd$			
Progressive Sampling	$\surd$	$\surd$	$\surd$	
Dynamic Sampling	$\surd$	$\surd$	$\surd$	
Uncertainty Sampling	$\surd$			
Relevance Sampling	$\surd$			
Sub-sampling	$\surd$			
Similarity Driven Sampling	$\surd$			
Density Biased Sampling	$\surd$	$\surd$		
Resampling	$\surd$			

There are some differences between sampling in data mining and that of traditional sampling. First, their start points are different. In statistics, little data about a population is available and sampling is often the only way to get information about the population. However, for data mining, large amounts of data are available and what we worry about is how to use them. In other words, data mining researchers can assume that the population is ready for use, while statistician usually assume the population is unknown. Second, their targets are different. The motivation of sampling in data mining are usually: (1) to overcome the limitation of memory, (2) to reduce the complexity of learning algorithm, or (3) to accelerate the learning speed without losing much of accuracy, while in statistics the objective is to construct some statistics to get estimates of the population. Third, different operations will be conducted on obtained samples. In data mining, data mining algorithms will run on the sample instead of its mother data to get knowledge, while in statistics, sampled data are used to get estimators about interested characteristics of an unknown population.

Although many traditional sampling strategies exists, not all of them are being used in data mining. The above mentioned differences accounts for this. Besides, some sampling strategies such as *random sampling* can be wholly used, from their ideas to its implementations, while others such as *adaptive sampling* are only used with their ideas. This is mainly due to the special requirements of data mining. Data mining puts a great emphasis on its automatic running with human intervention as few as possible. This requires a sampling strategy should possess a mechanism that is well formated and programmable. Random sampling has an intrinsic mechanism that can be easily implemented on computer. This is one of the main reasons that random sampling is popular in data mining. However, many other traditional sampling strategies do not possess such a mechanism so that are not easy to program them. This may explain why they are not so popular as random sampling and also why many non-traditional sampling strategies are invented for data mining by borrowing the ideas of traditional sampling strategies. However, this does not mean that these traditional sampling strategies are not useful to data mining but really mean that we need to find ways to use them for data mining.

There can be two ways to apply them to data mining. One is to modify their statistical implementations into computerized implementations. This can be a huge work because it is subject to changing from general considerations in statistics to those in data mining. We have introduced general considerations in statistics and will introduce those in data mining later. The other way is to borrow their idea to invent new sampling strategies which work in data mining context. This seems to be easier than the former because an idea of a sampling strategies is usually easy to be considered within the design of a data mining algorithm. However, this may involve partially or sometimes wholly redesigning the data mining algorithm.

4.4 Procedures and Recategorization of Sampling

Generally speaking, sampling, in the data mining context, is composed of three basic stages, which are:

1. a *selection procedure to decide which instance is selected in the sample,*
2. *a stopping criterion to decide when to stop selection,*
3. *an evaluation procedure to check if the obtained sample is good or not.*

The whole procedure of generating a sample data set from a population data set is shown in Figure 2.

Input: population data set D;
  Repeat *Selection Procedure*;
  Until *Stopping Criterion*;
  Do *evaluation procedure*;
Output: sample data set S ($S \subset D$);

Figure 2: **Generic Sampling Procedure**

In the selection procedure, instances (data items) in a given data set (i.e., the population) are checked one by one, starting from the first. There can be many variations in this step and most statistical sampling designs can be used. Generally, this step can use (1) non-probabilistic, (2) equal-probabilistic, (3) unequal-probabilistic schemes, involving all strategies mentioned in Section 3. The stopping criterion can be set by (1) reaching a given sample size, or (2) satisfying a condition where the sample size is not set beforehand. The condition referred in the second criterion can have many forms, e.g., the number of instances with some specified values. The evaluation procedure corresponds to the estimation procedure in a statistics context. The obtained sample is evaluated by a user-specified strategies, which can be sometimes statistical hypothesis tests, to check if the sample is of the desired one so that we can apply data mining algorithms. Due to different data mining tasks and different data mining algorithms, an evaluation strategy should be highly related to a specified task or algorithm. Although, in our opinion, this step is a reasonable step of a sampling procedure, it is often omitted in applications. Later we will discuss the issue of sample quality and argue that by checking this quality we can evaluate if a sample is good enough to be a representative of its parent data set.

5 Research Issues of Sampling for Data Mining

Kivinen and Mannila [26] show the power of sampling for data mining theoretically. Their work basically answers the question of when sampling will be effective but does not answer the question of which sampling strategy will work better. However, to make full use of sampling in data mining, we believe, there is still a lot of work to do. We list some of them below, most of which are usually not within traditional statistical considerations but essential to sampling in data mining:

- Implementing common sampling strategies on common data structures.
- Embedding sampling in data mining algorithms.
- Suitable sampling strategies for specific data mining tasks.
- Suitable sampling strategies for specific data sets.
- Novel sampling strategies for data mining.

A sampling strategy may have different implementations on different data structures. Data mining deals with data that usually stores in physical external memory in various logical data structures. As the efficiency of access to these data is a key issue to efficient data mining and this efficiency depends largely on data structures, it is necessary to implement common sampling strategies in efficient ways for common data structures. There has been some works addressing this issue. For example, Olken [41] studies random sampling from databases (e.g., $B+$ tree, Hash files). However, we find that there is little research on this issue for many other sampling strategies. Random sampling is very useful but this does not mean that it is omnipotent. Other sampling strategies also have the potential to play their roles in data mining. Therefore, it is necessary to study efficient implementation of common sampling strategies on common data structures.

A sampling strategy may play a better role if embedded in the procedure of a data mining algorithm than otherwise. Although the procedure of sampling can be done outside the procedure of data mining, there are cases that binding the two can produce effects of "$1+1 > 2$". As we have seen in Section 4, some sampling strategies such as "windowing" [48] have been successfully embedded in data mining algorithms. This will be not easy and may sometimes need modification of a data mining algorithm, but this could make the data mining algorithm work better. Therefore, it is desirable to study the possibility of binding sampling within a data mining algorithm.

For a specific data mining task and a specific data set, one sampling strategy may work better than others in terms of accuracy or efficiency. Therefore, it is necessary to study how different sampling strategies are in a specific data mining task given a specific data set, in order to provide users a set of guideline for them to make decisions on which context it will be more suitable to use which sampling strategy. So far, there is little literature on this.

For some data mining tasks, existing sampling strategies may not work well. So it is hoped to invent new sampling strategies for them. "Density Biased Sampling" [44] provides a good sample for this end.

6 general considerations of Sampling in Data Mining

6.1 Efficiency Of Sampling

Efficiency is one of the most important issues for all data mining algorithms. So is learning accuracy. The more the data, the higher accuracy but the less efficiency; the less the data, the higher efficiency but the less accuracy. Users use sampling to gain in efficiency but risk losing learning accuracy. In some cases, due to a massive volume of data, sampling is necessary to achieve expected efficiency by decreasing the amount of data, whether or not it decreases learning accuracy as well. For example, in an application of inductive learning, Fayyad and et al [13] use sampling to reduce more than three terabytes of raw data. However, if it is necessary to scan the entire data set in order to produce a sample, or if sampling is needed to execute more than one time, extra cost will incur and this may result in loss of the advantage of sampling.

Most applications and discussions of sampling assume that producing random samples efficiently from large data sets is not difficult. For most large databases, this is simply not true. To produce a random sample from even a single-table database may require scanning the entire database. This has obvious implications for claims of algorithm efficiency with sampling; for example, the asymptotic run-time complexity will be at least linear in the total number of instances. Along with a better understanding of sampling must come database operations and organizations that allow for efficient sampling [12].

Generally speaking, time of a database scan to form a sample depends on the strategy by which the sample is obtained and the data structure that we work on. For example, if the database is a normal two-way data table and can be sequentially accessed, for some sampling strategies, like simple random sampling, no more than one scan is needed. However, for some other strategies, e.g. stratified sampling, if the strata are constructed by using the values of some or all records, database scan is done at least two times, one for forming the strata and the other for forming the sample. If the database has a special structure in which records of the database cannot be sequentially or randomly accessed, the number of scanning will be more than one. Thus, implementations of different sampling strategies on varied data structures should be studied to achieve high efficiency. Olken and Rotem [42] survey random sampling strategy from various data structures such as B+ Tree, Hash File and so on. However, as we know, there is little systematic and comprehensive research on this topic in general.

6.2 Sample size

Basically, this issue is subject to two limits: one is about the memory size, the other is about the required learning accuracy. The related question is that given the two limits, how large is enough to be a good sample to support a specified data mining algorithm. Although both statistics and data mining communities address the issue of sample size, these two communities tend to give quite different answers.

Those with formal statistics training are taught that patterns extracted from small representative samples, often less than 1000 observations, can produce accurate results in practice [20]. The data mining community, which is comprised of many diverse disciplines including computer science, machine learning, artificial intelligence and databases, most likely believes that larger datasets provide significantly more information. In the report by Harris-Jones and Haines [20], they present results of a study examining the potential information gains from varying sample size on five commonly used statistical and data mining algorithms. The results show that statistical regression strategies produce stable models on relatively small datasets of a few thousand records while data mining algorithms continue to improve with datasets into hundreds of thousands of records.

Thus, for data mining, sensible answers to the questions like “How large sample is enough for learning?” and “ Is more always better?” should be that it will depend on both the nature of the sampled population and the required accuracy by the learning algorithm. Some related research work has been done in this direction. In the work of Kivinen et al [26], they show that sampling is not always an efficient strategy for discovering universal sentences. Sampling can reliably show the falsity only for sentences that are in some sense clearly false. They also give an estimate of the sufficient sample size, depending on the error measure used. Toivonen [57] and Zaki et al [63] discuss the determination of sufficient sample size for finding association rules that are no smaller than a predefined size, based on tolerances on the probability of error and the size of the error.

A different view of sufficient sample size, that of sample complexity, is provided by Valiant's theoretical framework [59, 21], which for a given hypothesis space allows the calculation of the number of examples sufficient for learning with high probability a good approximation to the “true concept” if one exists in the hypothesis space. In the work of Oates and Jansen [40], they found that small numbers of training instances suffice to build small, accurate trees, in addition to yielding a useful tree-simplification tool, frees data previously used in tree construction for other purposes. Progressive sampling [45] provides a power tool for finding the correct sample size, although it stands on the assumption that detection of convergence can be done efficiently. These findings imply, from another viewpoint, that sampling can work with small sample size, while the lower limit of this size is still an open issue.

6.3 Randomness

Randomness usually guarantees the unbiasness of a sample. Generally, shuffling (randomizing the order of instances in a data set) before sampling or learning can meet the requirement. Although randomness is desired in most cases (not all cases), in the actual process of sampling it is often quite difficult to ensure absolute randomness. The main problems may be the famous conclusion by Von Neumann, i.e., one cannot generate a real random number by computer. What we actually get from any computer program is a pseudo random number. However, from the point of applications, a pseudo random number is sufficient. The problem here is how to guarantee a sample to be selected satisfying this (pseudo) randomness, especially when sampling in special data structures. To attain this, the underlying random process of selection should be addressed. An efficient algorithm for drawing a sequential random sample of n records from a database containing N records has been put by Vitter [60]. The algorithm is online (so that CPU time can be overlapped with I/O), has a small constant memory requirement, and is easy to program. Olken and Rotem [42] survey some random sampling implementations for some common data structures. Because data mining may mine on various data structures, it is highly needed to develop algorithms to make random sample on these data structures, e.g., for semi-structured data sets.

6.4 Sample Quality

By sample quality, we mean to measure the degree of a sample to which it resembles the original data, or the representativeness of the sample. This issue is closely related to sampling strategy and sample size as well as sampling efficiency. Given a sampling strategy and a sample size, traditionally, we can describe a sample by using distribution and sampling error.

The difference between the distributions of a sample and the original data has been usually accepted as a measure of sample quality. Intuitively, we hope a sample is sufficiently similar to its parent database [25]. One approach to determining this kind of similarity is to test the hypotheses that each field in the sample comes from the same distribution as the original database. For the categorical fields, a Chi-Square hypothesis test is used to test the hypothesis that the sample and the large database come from the same distribution. For the numeric fields, a large-sample test (relying on the central limit theorem) is used to test the hypothesis that the sample and the large database have the same mean. However, this measure has its intrinsic shortcomings. First, the setting of the confidence levels will affect sample size and performance, and this kind of effect is usually not easy to control. Second, it is difficult to decide whether a sample is good enough because it ignores the data-mining algorithm.

The basic sampling view assumes that the interested variable is measured on every unit in the sample without error, so that errors in the estimates occur only because just part of the population is included in the sample. Such errors are referred to as "sampling errors". In data mining, sampling errors also occur just because only part of the whole data set is selected in the sample. As a result of this partiality, learning accuracy may be undesirable. As we mentioned above, increasing sample size may help to increase learning accuracy. Thus, we can expect to cut down the sampling errors by using larger samples. However, even if memory is large enough to allow a larger sample to be loaded in, the effect of doing so still depends on the properties of the interested data set and the learning algorithms we use.

Therefore, either we have to find more reliable testing measure to check the similarity or we should seek for other ideas to define the quality of a sample. In the work of John and Langley [25], they present a PCE (Probably Close Enough) criterion to evaluate a sampling strategy by taking into account the knowledge of learning algorithm. Actually, they abandon the use of distribution. A thorough study on the relationship of distributions between a sample and its parent may propose most powerful tools to measure the effect of sampling.

Another issue related to sample quality is which sampling strategy will give the best sample quality. It can be concluded that no sampling strategy will work for all data mining tasks. For example, random sampling will miss rare classes so that it is not suitable for some clustering task. This is also why "density biased sampling" is needed for such tasks. It is better to do an extensive study of the pros and cons of each sampling strategy for different data mining tasks. As we know, this is still an open topic in data mining.

7 Conclusion

Sampling methodology, which has a long tradition in statistics, can profitably be used to mitigate computational requirements in data mining. Also, a powerful, computationally intense procedure operating on a sample of the data may in fact provide superior accuracy than a less sophisticated one using the entire database [15]. By surveying sampling strategies, we find many sampling strategies that have been applied in different domains of applications. We categorize them according to their characteristics. We also investigate how sampling techniques have been used in data mining. We discuss general considerations when sampling in data mining and propose potential research issues on how to use sampling more effectively for data mining. We conduct experiments to show the importance of these research in data mining views.

Till now, we find with some surprise that there are relatively fewer sampling schemes used in data mining, compared with a large variety of sampling strategies which have existed for quite some time in statistics. Furthermore, due to the inherent requirement of data mining, some conclusions of traditional sampling may not apply or be desirable in data mining. Therefore, more theoretical and empirical research will be needed before we can claim a thorough understanding of sampling and its application to data mining.

The ultimate goal of studying sampling in data mining is to provide a comprehensive set of sampling techniques including strategies, algorithms and guidelines for users to determine which context requires which sampling technique. Some preliminary work has been done along the direction of the above topics. However, in order to accomplish these challenging tasks, data miners and statisticians should work more closely to overcome the barriers between sampling theory and data mining tasks. Further research on these topics will be great contributions to data mining as well as sampling itself because it would supply powerful tools for data mining especially in dealing with ever increasing large databases.

References

[1] J.H. Ahrens and U. Dieter. Sequential random sampling. *ACM Transections on Mathematical Software*, 11:157–169, 1985.

[2] P.S. Bradley, U. Fayyad, and C. Reina. Scaling cluster algorithms to large databases. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*. AAAI Press / The MIT Press, 1998.

[3] S.T. Buckland, D.R. Anderson, K.P. Burnham, and J.L. Laake. *Distance Sampling: Estimating Abundance of Biological Populations*. Chapman – Hall, Inc., 1993.

[4] J. Catlett. Megainduction: A test flight. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 596–599. Morgan Kaufmann, 1991.

[5] J. Catlett. *Megainduction: Machine Learning on Very Large Database*. PhD thesis, School of Computer Science, University of Technology, Sydney, Australia, 1991.

[6] S. Chaudhuri, R. Motwani, and V. Narasayya. Random sampling for histogram construction: How much is enough? In *Proceedings of ACM SIGMOD'98*, 1998.

[7] S. Chaudhuri, R. Motwani, and V. Narasayya. On random sampling over joins. In *Proceedings of the ACM SIGMOD'99*, pages 263–274, 1999.

[8] M.S. Chen, J.W. Han, and P.S. Yu. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8:866–883, 1996.

[9] W.G. Cochran. *Sampling Techniques*. Wiley, third edition, 1977.

[10] C. Domingo, R. Gavalda, and O. Watanabe. Adaptive sampling methods for scaling up knowledge discovery algorithms. Technical report, Dept. of Math. and Comp. Science, Tokyo Institute of Technology, 1999.

[11] S.C. Fan. *Basic Sampling Methods*. University of Singapore, 1967.

[12] U. Fayyad and P. Smyth. From massive data sets to science catalogs: Application and challenges. In *Proceedings of the Workshop on Massive Data Sets*, 1995.

[13] U. Fayyad, N. Weir, and S. Djorgovski. Skicat: A machine learning system for automated cataloging of large scale sky surveys. In *Proceedings of the 10th International Conference on Machine Learning*. Morgan Kaufmann, 1993.

[14] E.K. Foreman. *Survey Sampling Principles*. M. Dekker, 1991.

[15] J.H. Friedman. Data mining and statistics: What is the connection. *29th Symposium on the Interface between computer science and statistics*, 1997.

[16] J. Furnkranz. Integrative windowing. *Journal of Artificial Intelligence Research*, 8:129–164, 1998.

[17] D. Gamerman. *Markov Chain Monte Carlo: Stochastic Simulation For Bayesian Inference*. Chapman – Hall, 1997.

[18] P.B. Gibbons and Y. Matias. New sampling-based summary statistics for improving approximate query answers. In *Proceedings of ACM SIGMOD'98*, pages 331–342, 1998.

[19] P.J. Haas and A.N. Swami. Sequential sampling procedures for query size estimation. In *Proceedings of the ACM SIGMOD'92*, 1992.

[20] C. Harris-Jones and T.L. Haines. Sample size and misclassification: Is more always better. In *Working Paper of American Management Systems Center for Advanced Technologies*. AMS Center for Advanced Technologies, 1997.

[21] D. Haussler. Quantifiying inductive bias: Ai learning algorithms and valiant's learning fromawork. *Artificial Intelligence*, 36:177 – 221, 1988.

[22] A.S. Hedayat. *Design and Inference in Finite Population Sampling*. NY: John Wiley – Sons, Inc., 1991.

[23] J.R. Higgins. *Sampling Theory in Fourier and Signal Analysis*. Oxford Science Publications, 1996.

[24] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1988.

[25] G.H. John and P. Langley. Static versus dynamic sampling for data mining. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI / MIT Press, 1996.

[26] J. Kivinen and H. Mannila. The power of sampling in knowledge discovery. In *Proceedings of the thirteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 77–85. ACM, 1994.

[27] P.R. Krishnaiah and C.R. Rao. *Handbook of Statistics 6: Sampling*. North-Holland, 1988.

[28] S.D Lee, D.W. Cheung, and B. Kao. Is sampling useful in data mining? a case in the maintenance of discovered association rules. *Data Mining and Knowledge Discovery*, 2:233 – 262, 1998.

[29] D.D. Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. *SIGIR Forum*, 29:13–19, 1995.

[30] D.D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of Eleventh Internaltional Conference on Machine Learning (ML'94)*, pages 148–156. Morgan Kaufmann, 1994.

[31] D.D. Lewis and W.A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of Seventeenth Annual Internaltional ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. Springer-Verlag, 1994.

[32] R.J. Lipton and J.F. Naughton. Query size estimation by adaptive sampling. *Journal of Computer and System Science*, 51:18–25, 1995.

[33] H. Lu, R. Setiono, and H. Liu. Neurorule: A connectionist approach to data mining. In *Proceedings of the 21st VLDB Conference (VLDB'95)*, 1995.

[34] D.J.C. Mackay. *Introduction to Monte Carlo Methods, in Learning in Graphical Models*. Kluwer Academic Publishers, 1998.

[35] H. Mannila. Methods and problems in data mining. In *Proceedings of International Conference on Database Theory*. Springer-Verlag, 1997.

[36] M. Meila and D. Hecherman. An experimental comparison of several clustering and initialization methods. Technical report, Center for Biological and Computational Learning, MIT, 1998.

[37] C. J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1996. FTP from ics.uci.edu in the directory pub/machine-learning-databases.

[38] D. Michie, D.J. Spiegelhalter, and C.C. Taylor. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood Limited, Campus 400, Maylands Avenue, Hemel Hempstead, Hertfordshire, HP2 7EZ, England, 1994.

[39] M.N. Murthy. *Sampling: Theory and Methods.* Statistical Publish Society, 1967.

[40] T. Oates and D. Jensen. The effects of training set size on decision tree complexity. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.

[41] F. Olken. *Random Sampling from Databases.* PhD thesis, Department of Computer Science, University of California Berkeley, 1993.

[42] F. Olken and D. Rotem. Random sampling from databases: A survey. *Statistics and Computing*, 5:25 – 42, 1995.

[43] F. Olken and D. Rotem. Sampling from spatial databases. *Statistics and Computing*, 5:43 – 57, 1995.

[44] C.R. Palmer and C. Faloutsos. Density biased sampling: An improved method for data mining and clustering. In *Proceedings of the ACM SIGMOD'2000*, 2000. (To appear).

[45] F. Provost, D. Jensen, and T. Oates. Efficient progressive sampling. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining(KDD'99)*, pages 23–32. AAAI/MIT Press, 1999.

[46] F. Provost and V. Kolluri. A survey of methods for scaling up inductive algorithms. *Machine Learning*, pages 1–42, 1999.

[47] J. Quinlan. Induction of decision trees. *Machine Learning*, pages 81 – 106, 1986.

[48] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993. (http://www.cse.unsw.edu.au/ quinlan/).

[49] J.R. Quinlan. Baggin, boosting, and c4.5. *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI'96)*, 1996.

[50] T. Reinartz. Similarity-driven sampling for data mining. In *The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases(PKDD'98).* Springer, 1998.

[51] R.L. Scheaffer, W. Mendenhall, and Ott L. *Elementary Survey Sampling.* Duxbury Press, fifth edition, 1996.

[52] R. Singh. *Elements of Survey Sampling.* Kluwer Academic Publishers, 1996.

[53] R.K. Som. *Practical Sampling Techniques.* M. Dekker, second edition, 1995.

[54] A. Stuart. *Basic Ideas of Scientific Sampling.* London: Griffin, 1976.

[55] S.K. Thompson. *Sampling.* Wiley, 1992.

[56] S.K. Thompson and G. A. F. Seber. *Adaptive Sampling.* Wiley, 1996.

[57] H. Toivonen. Sampling large databases for association rules. In *Proceedings of the 22nd International Conference on Very Large Data Base (VLDB'96).* Morgan Kaufmann, 1996.

[58] W.M.K. Trochim. *Research Methods in Knowledge Base.* Cornell Custom Publishing, second edition, 1999.

[59] L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134 – 1142, 1984.

[60] J.S. Vitter. An efficient algorithm for sequential random sampling. Technical report, Brown University, 1986.

[61] T. Yamane. *Elementary Sampling Theory.* Prentice Hall, 1967.

[62] Y. Yang. Sampling strategies and learning efficiency in text categorization. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pages 88–95. AIII Press, 1996.

[63] M.J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. In *Proceedings of the 7th Worhshop on Research Issues in Data Engineering*, 1997.