# Efficiently Determine the Starting Sample Size for Progressive Sampling

Baohua Gu [*] Bing Liu [†] Feifang Hu [‡] Huan Liu [§]

**Abstract**

Given a large data set and a classification learning algorithm, Progressive Sampling (PS) uses increasingly larger random samples to learn until model accuracy no longer improves. It is shown that the technique is remarkably efficient compared to using the entire data. However, how to set the starting sample size for PS is still an open problem. We show that an improper starting sample size can still make PS expensive in computation due to running the learning algorithm on a large number of instances (of a sequence of random samples before achieving convergence) and excessive database scans to fetch the sample data. Using a suitable starting sample size can further improve the efficiency of PS. In this paper, we present a statistical approach which is able to efficiently find such a size. We call it the *Statistical Optimal Sample Size* (SOSS), in the sense that a sample of this size sufficiently resembles the entire data. We introduce an information-based measure of this resemblance (Sample Quality) to define the SOSS and show that it can be efficiently obtained in one scan of the data. We prove that learning on a sample of SOSS will produce model accuracy that asymptotically approaches the highest achievable accuracy on the entire data. Empirical results on a number of large data sets from the UCIKDD repository show that SOSS is a suitable starting size for Progressive Sampling.

**Keywords:** *data mining, sampling, learning curve, optimal sample size.*

## 1 Introduction

Classification is an important data mining (DM) task. It is often solved by the decision tree approach [13]. However, given a very large data set, directly running a tree-building algorithm on the whole data may require too much computation resource and lead to an overly complex tree. A natural way to overcome these problems is to do sampling [7, 3]. One of the recent works towards improving tree-building efficiency (both in terms of time and memory space required) by sampling is Boat by Gehrke et al [6]. [6] builds an initial decision tree using a small sample and then refines it via bootstrap to produce exactly the same tree as that would be produced using the entire data. Boat is able to achieve excellent efficiency in tree construction. However, due to the large size of the data, the produced tree can be very complex and large, which makes it hard for human understanding. It has been observed that the tree size often linearly increases with the size of training data, and additional complexity in the tree results in no significant increase in model accuracy [9]. Another recent work on improving the efficiency of tree building for large data sets is Progressive Sampling (PS for short) proposed by Provost et al [11]. By means of a learning curve (see an example learning curve in Figure 1) which depicts the relationship between sample size and

[*]Email: gubh@comp.nus.edu.sg; School of Computing, National University of Singapore
[†]Email: liub@comp.nus.edu.sg; School of Computing, National University of Singapore
[‡]Email: stahuff@nus.edu.sg; Department of Statistics & Applied Probability, National University of Singapore
[§]Email: hliu@asu.edu; Department of Computer Science and Engineering, Arizona State University

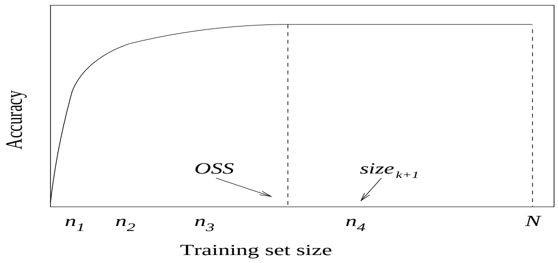



Figure 1: Learning Curves and Progressive Samples

model accuracy, PS searches for the optimal model accuracy (the highest achievable on the entire data) by feeding a learning algorithm with progressively larger samples. Assuming a well-behaved learning curve, it will stop at a size equal to or slightly larger than the optimal sample size ($OSS$ for short) corresponding to the optimal model accuracy. [11] shows that PS is more efficient than using the entire data. It also avoids loading the entire data into memory, and can produce a less complex tree or model (if the real $OSS$ is far less than the total data size). In this paper, we restrict our attention to PS and aim to improve it further by finding a good starting sample size for it.

Obviously, the efficiency of PS gets to the highest when the starting sample size is equal to the $OSS$; and the smaller the difference between the two, the higher the efficiency. The following analysis shows how much benefit can be gained from a proper starting sample size. Suppose we use the simple geometric sampling schedule suggested by [11]. We denote the sample size sequence by $\{n_0, a*n_0, a^2*n_0, a^3*n_0, ...\}$, where $n_0 > 0$ is the starting sample size, $a > 1$ is the increment ratio. If the convergence is detected at the $(k+1)$th sample (its size being $size_{k+1} = a^k * n_0 \approx OSS$), then the total size of the previous $k$ samples will be $size_{1..k} = n_0 * (1 + a + a^2 + ... + a^{k-1}) = n_0 * (a^k - 1)/(a - 1) = (size_{k+1} - n_0)/(a - 1)$. Obviously, the extra learning on the $size_{1..k}$ number of instances could be significantly reduced if $n_0$ is set near to the $OSS$. Moreover, as $k = \log_a \frac{S_{k+1}}{n_0} \approx \log_a \frac{OSS}{n_0}$, if $n_0$ is much less than $OSS$, then $k$, the number of samples needed before convergence, will be large. Note that generating a random sample from a single-table database typically requires scanning the entire table once [12]. Thus a large $k$ will result in considerably high disk I/O cost. Therefore setting a good starting sample size can further improve the efficiency of PS by cutting the two kinds of costs.

In this paper, we find such a size via a statistical approach. The intuition is that a sample with the $OSS$ should sufficiently resemble its mother data (the entire data set). We implement this intuition via three steps. First, we define an information-based measure of the resemblance (we call *Sample Quality*). Based on this measure, we then define a *Statistical Optimal Sample Size* ($SOSS$ for short). We prove that learning on a sample of the $SOSS$ will produce a model accuracy that is sufficiently close to that of the $OSS$. We show that the $SOSS$ can be efficiently determined in one scan of the mother data. Our experiments on a number of UCIKDD [1] data sets show that our approach is effective.

The remainder of this paper is organized as follows. Below we first discuss the related work. In Section 2, we introduce the measure of sample quality and its calculation. In Section 3, we define SOSS, prove an useful theorem and show its calculation. Experimental results are given in Section 4. Section 5 concludes the paper.

### 1.1 Related Work

Besides being used in classification, sampling has also been applied to other data mining tasks. For example, Zaki et al [14] set the sample size using Chernoff bounds and find that sampling can speed up mining of association rules. However, the size bounds are found too conservative in practice. Bradley et al scale up an existing clustering algorithm to large databases via repeatedly updating current model with (random) samples [2]. In the database community, sampling is also widely studied. For example, Ganti et al [5] introduce self-tuning samples that incrementally maintain a sample, in which the probability of a tuple being selected is proportional to the frequency with which it is required to answer queries (exactly). However, in both [2] and [5], how to decide a proper sample size is not mentioned.

The work that we find most similar to ours is that given in [4], where Ganti et al introduce a measure to quantify the difference between two data sets in terms of the models built by a given data mining algorithm. Our measure is different as it is based on statistical information divergence of the data sets. Although [4] also addresses the issue of building models using random samples and shows that bigger sample sizes produce better models, it does not study how to determine a proper sample size.

Our method can be a useful complement to the existing techniques. The proposed $SOSS$ also has a good property: it only depends on the data set and is independent of the algorithm. Although we present it for classification in this paper, this property may make it applicable to other DM techniques, which will be studied in our future research.

## 2 Sample Quality Measure

### 2.1 Underlying Information Theory

A sample with the $OSS$ should inherit the "property" of its mother data as much as possible. This property can be intuitively interpreted as information. In this work, we make use of Kullback's information measure [8], which is generally called the *divergence* or *deviation*, as it depends on two probability distributions and describes the divergence between the two. Below we briefly describe its related definitions and conclusions.

Suppose $x$ is a specific value (i.e., an observation) of a generic variable $X$, and $H_i$ is the hypothesis that $X$ is from a statistical population with generalized probability densities (under a probability measure $\lambda$) $f_i(x)$, $i = 1, 2$, then the information divergence is defined by $J(1,2) = \int (f_1(x) - f_2(x)) log\frac{f_1(x)}{f_2(x)} d\lambda(x)$. According to [8], $J(1,2)$ is a measure of the divergence between hypotheses $H_1$ and $H_2$, and is a measure of the difficulty of discriminating between them. Specifically, for two multinomial populations with $c$ values ($c$ categories), if $p_{ij}$ is the probability of occurrence of the $j$th value in population $i$ ($i = 1, 2$, and $j = 1, 2, ..., c$), then the information divergence is $J(1,2) = \sum_{j=1}^{c} (p_{1j} - p_{2j}) log\frac{p_{1j}}{p_{2j}}$.

The information divergence has a good limiting property described in Theorem 1 below (see [8] for its proof), based on which we will prove an important theorem about $SOSS$ in Section 3.

**Theorem 1** *Given a probability density function $f(x)$ and a series of probability density functions $\{f_n(x)\}$, where $n \to +\infty$, denote the information divergence from $f_n(x)$ to $f(x)$ as $J(f_n(x), f(x))$, we have, if $J(f_n(x), f(x)) \to 0$, then $f_n(x)/f(x) \to 1[\lambda]$, uniformly. Here $[\lambda]$ means that the limitation and the fraction hold in a probability measure $\lambda$.*

### 2.2 Definition and Calculation

We borrow the concept of information divergence between two populations and define our sample quality measure below. The idea is to measure the dissimilarity of a sample from its mother data by calculating the information divergence between them.

**Definition 1** *Given a large data set $D$ (with $r$ attributes) and its sample $S$, denote the information divergence between them on attribute $k$ as $J_k(S, D)$, ($k = 1, 2, ..., r$), then the sample quality of $S$ is $Q(S) = exp(-J)$, where the averaged information divergence $J = \frac{1}{r}\sum_{k=1}^{r} J_k(S, D)$.*

In calculating $J_k(S, D)$, we treat a categorical (nominal) attribute as a multinomial population. For a continuous (numerical) attribute, we build its histogram (e.g., using a simple discretization), and then treat it also as a multinomial population by taking each bin as a categorical value and the bin size as its frequency. According to [8], the information divergence $J > 0$, therefore $0 < Q \leq 1$, where $Q = 1$ means that no information divergence exists between $S$ and $D$. The larger the information divergence $J$, the smaller the sample quality $Q$; and vice versa. For numerical attributes, if we have prior knowledge about their distributions, further improvement on the calculation of $J$ can be achieved by directly applying them to the information divergence.

The calculation of $Q$ is straightforward. In one scan of the data set, both the occurrences of categorical values and the frequencies of numerical values that fall in the bins can be incrementally gathered. Therefore the time complexity of calculating sample quality is $O(N)$ ($N$ is the total number of instances or records of the mother data), while the space complexity is $O(r * v)$ ($v$ is the largest number of distinct values or bins of each attribute). Note that a random sample can be obtained also in one scan. Thus we can calculate a sample's quality while generating it.

# 3 Statistical Optimal Sample Size

## 3.1 Definition

From the definition of sample quality, we can observe that the larger the sample size, the higher the sample quality. This is because as sample size increases, a sample will have more in common with its mother data, therefore the information divergence between the two will decrease. We define the *Statistical Optimal Sample Size* ($SOSS$) as follows:

**Definition 2** *Given a large data set $D$, its $SOSS$ is the size at which its sample quality is sufficiently close to 1.*

Clearly, the $SOSS$ only depends on the data set $D$, while the $OSS$ depends on both the data set and the learning algorithm. Therefore, the $SOSS$ is not necessarily the $OSS$. However, by the following theorem, we can see that their corresponding model accuracies can be very close.

Given an learning algorithm $L$ and a sufficiently large data set $D$ with probability density function $f_D(x)$, we take $L$ as an operator mapping $f_D(x)$ to a real number (namely the model accuracy), i.e., $L : f_D(x) \rightarrow Acc^*$, where $Acc^*$ is the maximum accuracy obtained on $D$. Assume that a random sample of $OSS$ has the probability density function $f_{oss}(x)$, and a random sample of $SOSS$ has $f_{soss}(x)$. Let the model accuracies on the two samples be $Acc_{oss}$ and $Acc_{soss}$ respectively. We have the following theorem.

**Theorem 2** *If a random sample $S$ of $D$ has a probability density function $f_S(x)$ and $L$ satisfies that $f_S(x)/f_D(x) \rightarrow 1[\lambda] \Longrightarrow |L(f_S(x)) - L(f_D(x))| \rightarrow 0$, then $Acc_{soss} \rightarrow Acc_{oss}$.*

**Proof** (Sketch): Suppose we have a series of $n$ random samples of $D$ with incrementally larger sample sizes. Denote the size of the $i$-th sample as $\{S_i\}$, and the corresponding sample quality of the sample with $Q(S_i)$, $i = 1, 2, ..., n$. According to the definition of $SOSS$, when $S_i \rightarrow SOSS$, $Q(S_i) \rightarrow 1$, i.e., the information divergence of $S_i$ from $D$ is $J(S_i, D) \rightarrow 0$. Applying Theorem 1, we have, $f_{S_i}(x)/f(x) \rightarrow 1$, therefore, $|L(f_{S_i}(x)) - L(f(x))| \rightarrow 0$. In an asymptotic sense, $L(f_{S_i}(x)) \rightarrow Acc_{soss}$ and $L(f(x)) \rightarrow Acc_{oss}$, that is, $Acc_{soss} \rightarrow Acc_{oss}$. #

The premise of the theorem means that if two data sets are quite similar in their probability distributions, then the learning algorithm should produce quite close model accuracy on them. This is reasonable for a typical learning algorithm. Based on this theorem, we can search for the $SOSS$ by measuring sample quality instead of directly searching for the $OSS$ by running an expensive learning algorithm. We can also expect that the two are close in terms of model accuracy.

### 3.2 Calculation

To calculate the $SOSS$ of $D$, we can set $n$ sample sizes $S_i$ spanning the range of $[1, N]$ and compute the corresponding qualities $Q_i$ $(i = 1, 2, ..., n)$. We then draw a sample quality curve (relationship between sample size and sample quality) using these $(S_i, Q_i)$ points. The $SOSS$ is estimated using the curve. To be efficient, we can calculate all samples' qualities at the same time in one sequential scan of $D$ by using the idea of Binomial Sampling [10]. That is, upon reading in each instance or data record, an random number $x$ uniformly distributed on $[0.0, 1.0)$ is generated. If $x < S_i/N$, then corresponding statistics (by counting a categorical value or binning a numerical value) are gathered for the $i$-th sample. We describe the procedure using the pseudo algorithm below.

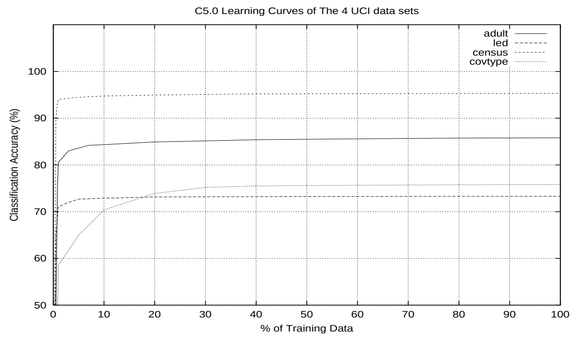
Pseudo Algorithm $SOSS$
**input**: a large data set $D$ of size $N$, $n$ sample sizes $\{S_i | i = 1, 2, ..., n\}$;
**output**: $n$ pairs of $(S_i, Q_i)$;
**begin**
1. for each instance $k$ in $D$ $(k \in [1, N])$:
    update corresponding statistics for $D$;
    for each sample $i$:
      $r \leftarrow UniformRand(0.0, 1.0)$;
      if $(r < \frac{S_i}{N})$, then update corresponding statistics for sample $i$;
2. for each sample $i$: calculate its $Q_i$ and output $(S_i, Q_i)$;
**end**

It is easy to see that its run-time complexity and the memory needed are $n$ times that of computing sample quality for a single sample. With these $(S_i, Q_i)$ points in hand, we draw the quality curve and decide the $SOSS$ in the following way: starting from the first point, we do a linear regression on every $l$ consecutive points (we set $l = 5$ in our experiment), if the 95% confidence interval of the slope of the regressed line includes zero, then the size of the middle point is the $SOSS$.

## 4 Experimental Results

We evaluate our measure on four large UCIKDD data sets: *adult*, *led*, *census*, and *covtype*. The number of instances in training data/testing data are, 36k/12.8k for *adult*, 100k/50k for *led*, 199.5k/99.8k for *census*, and 400k/181k for *covtype*. The classification algorithm we use is C5.0, the latest improvement of C4.5 [13]. We first draw the learning curve for each data set by running C5.0 on a number of sample sizes from 1% to 100% of all training data. All accuracies are tested against corresponding testing data. Then we draw the quality curve for each data set by calculating multiple sample qualities in one scan. We set the bin number to be 20 for discretizing numerical attributes. We use 50 sample sizes equally covering $[1, N]$. The learning curves and quality curves (averaged on 10 runs for the four data sets) are shown in Figure 2. We decide the $OSS$ as the size corresponding to $99.5\% * Acc^*$, where $Acc^*$ is the model accuracy on all instances. The $SOSS$ is determined by the method in Section 3. The resulting $OSS$ and $SOSS$ with their corresponding tree sizes ($S_{tree}$) and tested accuracies ($Acc$) are listed in Table 1. We can see that for *led* and *census*, the $SOSS$ is equal to or very close to the $OSS$ [1]. If starting from this size, PS will reach the optimal model accuracy at once. For the *adult* and *covtype* data sets, the $SOSS$ is about half of the $OSS$, which means only one more iteration is needed for PS (if the size increment ratio $a = 2$) starting with the $SOSS$. In all data sets, the resulting tree sizes with both the $SOSS$ and the $OSS$ are

[1] It happens that the $SOSS$ is slightly larger than the $OSS$ for *census*. This may be due to the way we decide the $OSS$. Intuitively, the $SOSS$ could not be far larger than the $OSS$, unless the learning algorithm might produce very different learning accuracy on two similar distributions.



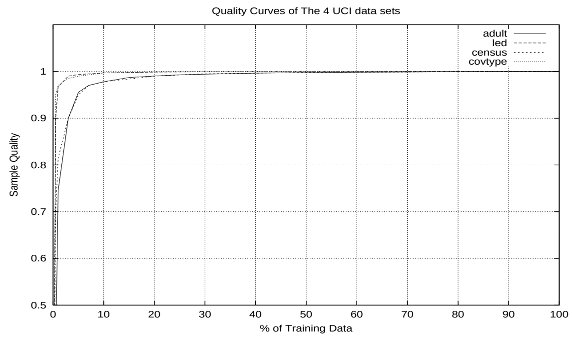




Figure 2: C5.0 Learning Curves (left) and Quality Curves (right)

Table 1: Results of the 4 Data Sets

| data set | $N$ | $S_{tree}$ | $Acc^*$ | $oss$ | $S_{tree}$ | $Acc$ | $soss$ | $S_{tree}$ | $Acc$ | $t_{ps+soss}$ ($t_{ps}/t_N$) |
|---|---|---|---|---|---|---|---|---|---|---|
| adult | 36k | 344 | 86.2% | 14k | 170 | 85.8% | 8k | 127 | 85.6% | 18s (20s/22s) |
| led | 100k | 3896 | 73.3% | 10k | 480 | 72.9% | 10k | 480 | 72.9% | 14s (38s/41s) |
| census | 199.5k | 848 | 95.3% | 30k | 91 | 94.8% | 35k | 108 | 94.9% | 94s (152s/176s) |
| covtype | 400k | 10970 | 75.8% | 140k | 4402 | 75.4% | 70k | 2417 | 73.0% | 668s (766s/1330s) |

significantly smaller than those with the full size, while their accuracy are very close. We also empirically compare the execution time used for PS starting with the found $SOSS$ ($t_{ps+soss}$) (including the time for finding the $SOSS$) to that without the $SOSS$ ($t_{ps}$) [2]. The execution time of directly learning on the entire data ($t_N$) is also given. They are all in the last column of Table 1. We can see that the $SOSS$ indeed speeds up PS in all the data sets. All these results support our expectation that the $SOSS$ can be used as a good starting sample size for PS.

## 5 Concluding Remarks

[11] shows that Progressive Sampling is more efficient than learning on the entire data. In this paper, we showed that a proper starting sample size can further improve its efficiency. With the intuition that a sample "must" sufficiently resemble its mother data in order to produce a good model, we proposed a technique to find a suitable starting sample size by measuring the similarity of a sample and the mother data. An information-based sample quality or similarity measure was introduced. Based on this measure, we defined the $SOSS$, and proved that asymptotically, the $SOSS$ will achieve model accuracy very close to that of the $OSS$. This claim was supported by our experimental results on UCIKDD data sets. Furthermore, the $SOSS$ can also be efficiently determined in one scan of the mother data and is independent of learning algorithms. All these clearly suggest that the proposed $SOSS$ can be used to start a Progressive Sampling. It can save many runs on unnecessary small samples, which are too dissimilar to the entire data.

In the proposed sample quality measure, we did not consider dependency (or interaction) among the attributes for computational reason. Although our results are rather good, we shall in our future work examine whether including this factor would do even better. Another related issue is how the proposed

[2] We use the geometric size scheme for PS as suggested in [11]. The sample size starts from 1% of the total data and doubles in the next iteration ($a = 2$) and so on. All experiments run on SUN Sparc 450.

measure and the *SOSS* will be affected by data skewness. We will address it in the future as well.


## Acknowledgment

The authors wish to thank all anonymous referees for their valuable comments on the earlier version of the paper.


## References


[1] S.D. Bay. The UCI KDD Archive [http://kdd.ics.uci.edu], 1999.

[2] P.S. Bradley, U. Fayyad, and C. Reina. Scaling cluster algorithms to large databases. In *Proceedings of KDD'98*, 1998.

[3] 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*, 1996.

[4] V. Ganti, J. Gehrke, R. Ramakrishnan, and W.Y. Loh. A framework for measuring changes in data characteristics. In *Proceedings of PODS'99*, 1999.

[5] V. Ganti, M.L. Lee, and R. Ramakrishnan. Icicles: Self-tuning samples for approximate query answering. In *Proceedings of VLDB'00*, 2000.

[6] J. Gehrke, V. Ganti, R. Ramakrishnan, and W.Y. Loh. Boat—optimistic decision tree construction. In *Proceedings of ACM SIGMOD'99*, 1999.

[7] J. Kivinen and H. Mannila. The power of sampling in knowledge discovery. In *Proceedings of ACM SIGMOD/PODS'94*, 1994.

[8] S. Kullback. *Information Theory and Statistics.* John Wiley & Sons, Inc, New York, 1959.

[9] T. Oates and D. Jensen. Large datasets lead to overly complex models: an explanation and a solution. In *Proceedings of KDD'98*, 1998.

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

[11] F. Provost, D. Jensen, and T. Oates. Efficient progressive sampling. In *Proceedings of KDD'99.* AAAI/MIT Press, 1999.

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

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

[14] 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.