Initial Sampling for Automatic Interactive Data Exploration

Wenzhao Liu[1], Yanlei Diao[1], and Anna Liu[2]

[1]College of Information and Computer Sciences, University of Massachusetts, Amherst
[2]Department of Mathematics and Statistics, University of Massachusetts, Amherst

April 20, 2016

1 Introduction

In many real world applications, users might not know the queries to send to a database in order to retrieve data in the user-interested areas. Users can apply a trial and error method to discover the queries. However, as the data set is usually quite large, the discovery of queries will take a long time and the whole process is labor-intensive. We want to build a discovery-oriented, interactive data exploration system, that guides users to their interested data areas through interactive sample labeling process. In each iteration, the system will strategically select some sample points to present to users for feedback, as relevant or irrelevant, and finally converge to a query that is able to retrieve all the data in the user-interested area.

In this synthesis project, we mainly focus on the initial sampling problem. Initially, we don't have any input(data labels) from users regarding the area of interest and our goal is to use as few samples as possible to find at least one sample within the user-interest area. As we don't have any clue about the user interest before the first iteration, the most naive sampling method we can use is random sampling. We designed equi-width and equi-depth stratified sampling methods, and also applied them to the progressive sampling framework. In this project, we apply techniques from two areas, statistics and database systems. We theoretically analyzed the probability lower bound, that within k samples we can get at least one sample within the user-interest area, for random sampling, equi-width, equi-depth stratified sampling, and progressive sampling. We then compare the probability lower bound for these sampling methods. We implement the equi-width and equi-depth stratified sampling algorithms inside the PostgreSQL database, and test their performance(CPU time and I/O time) when we (1)change the number of tuples in the table, (2)use column tables with different number of columns, (3)change the number of dimensions in the sampling space. We also run simulations over synthetic data set to demonstrate our theoretical results for these sampling methods.

Assume we have a dataset with d dimensions, we define our data space as the minimum bounding box of our dataset inside the d-dimensional space. Figure 1 plot an example in a 2-dimensional space. The green rectangle, which is the minimum bounding box for our dataset(blue points), is our data space. We only draw a few data points in Figure 1 as an illustration, our real dataset is much larger. We only consider inside our data space, as there are no data points outside the data space. In d-dimensional space, the minimum bounding box, or our data space, is also d-dimensional. The orange areas are an example of the user interest areas. There may be multiple disjoint areas(in Figure 1, there are two areas) for the user interest areas. Each area may have irregular shape, as plotted in Figure 1. We want to draw samples from the data space, and increase the probability that at least one sample is within the user interest area, so that in the first iteration, we can get at least one positive feedback from the user. Otherwise, all samples selected will be labeled as negative, and the system cannot utilize the information to learn the user interest areas.

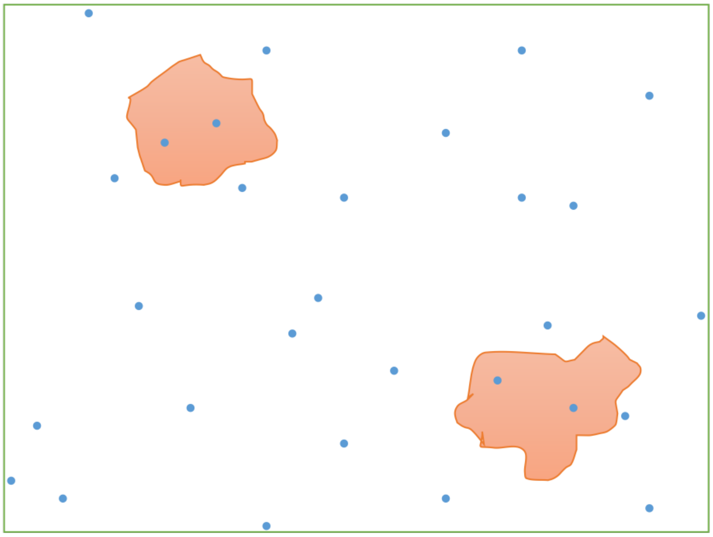


Figure 1: data space

We give the details of our sampling methods in the following section. In the analysis below, we assume the size of our data space(for example, the size of the area within the green rectangle in Figure 1) is A_t, where t means 'total', and the total number of data points within the data space is N_t. The size of the user interest area is A_i, where i means 'interest', and the number of data points within the user interest area is N_i.

2 Initial sampling methods

We present four methods below for our initial sampling task, which are random sampling, equi-width stratified sampling, equi-depth stratified sampling, and progressive sampling.

2.1 random sampling

For random sampling, suppose we know the total number of data points N_t, then we can generate k distinct random integers within $[1, N_t]$ according to the algorithm 1 below. Assume we have a column with row id from 1 to N_t for each tuple and we have an index for this column, then we can select the k tuples with row id corresponds to the k distinct random integers.

Algorithm 1 Random Sampling

```
procedure SELECT_RANDOM($k$)
    for $i$ = 1 to $k$ do
        int $r$;
        do
            $r \leftarrow rand(1, N_t)$
        while $r$ is in $samples[1...i-1]$
        $samples[i] \leftarrow r$
    return $samples[1...k]$
```

2.2 equi-width stratified sampling

In equi-width stratified sampling algorithm 2, we divide each dimension into equal-width bucket, so we will have multiple grids in the data space. Then we select one random sample from each grid. Suppose our data space is d dimensional. The minimum bounding box for our dataset in the d-dimensional space is $S = [L_1, H_1]*[L_2, H_2]*...*[L_d, H_d]$. And the length of the range in each dimension is $R_1 = H_1 - L_1, R_2 = H_2 - L_2, ..., R_d = H_d - L_d$. If we divide each dimension into c equal-width bucket, then we will get $k = c^d$ grids.

Algorithm 2 Equi-width Sampling

```
procedure SELECT_EQUIWIDTH($k$)
    $c \leftarrow \sqrt[d]{k}$
    int $i \leftarrow 1$
    for $j_1$ = 1 to $c$ do
        for $j_2$ = 1 to $c$ do
            ...
            for $j_d$ = 1 to $c$ do
                $samples[i] \leftarrow$ one random sample from data points within grid $[L_1 + (j_1 - 1) * \frac{R_1}{c}, L_1 + j_1 * \frac{R_1}{c}] * [L_2 + (j_2 - 1) * \frac{R_2}{c}, L_2 + j_2 * \frac{R_2}{c}] * ... * [L_d + (j_d - 1) * \frac{R_d}{c}, L_d + j_d * \frac{R_d}{c}]$
                $i \leftarrow i + 1$
    return $samples[1...k]$
```

We give an example for equi-width stratified sampling when $d = 2$ in Figure 2. We divide each dimension into equal width bucket(4 bucket in the example), so that we get k d-dimensional grids($k = 16$ grids in the example), and then we select one random sample from each grid.

2.3 equi-depth stratified sampling

In equi-depth stratified sampling algorithm 3, we divide the data space in a way that each grid in the data space has the same number of data points. Suppose our data space is d-dimensional. The d features are $F_1, F_2, ..., F_d$. The minimum bounding box for our dataset in the d-dimensional space is $S = [L_1, H_1] * [L_2, H_2] * ... * [L_d, H_d]$. In the first round, we sort all the data according to F_1 in ascending order, then we divide the range $[L_1, H_1]$ for F_1 into c buckets, so that the number of points within each bucket is almost the same(the different is at most 1). The boundary of each bucket can be determined by the average of two data points from each side and closest to the boundary. In the second round, we sort the data points in each

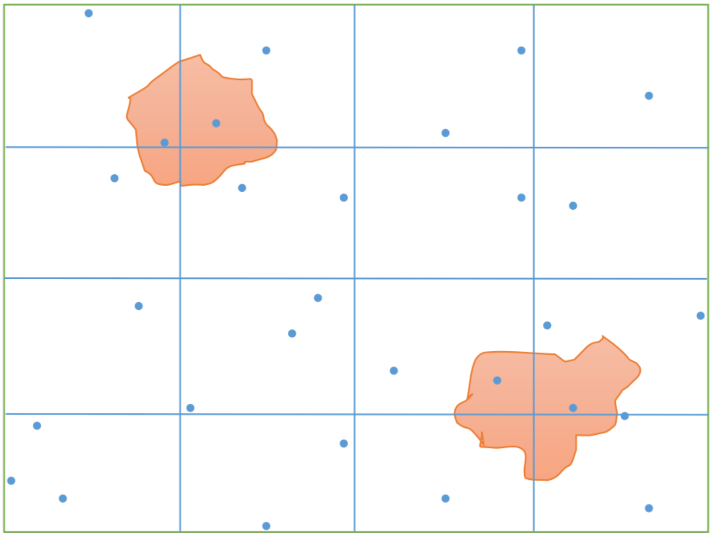


Figure 2: equi-width stratified sampling

bucket according to feature F_2, and divide each bucket into c sub-buckets, so the number of points within each sub-bucket is almost the same. We do this for d dimensions, and we know that after this process, each grid will have nearly the same number of data points.

Algorithm 3 Equi-depth Sampling

```
procedure SELECT_EQUIDEPTH($k$)
    $c \leftarrow \sqrt[d]{k}$
    $Bucket\_Set \leftarrow \{S\}$
    for $i$ = 1 to $d$ do
        $New\_Bucket\_Set \leftarrow \{\}$
        for each bucket $b$ in $Bucket\_Set$ do
            sort data points in $b$ according to feature $F_i$
            divide $b$ into $c$ sub-buckets $b_1, b_2, ..., b_c$ according to $F_i$, so that $|b_1| = |b_2| = ... = |b_c|$
            $New\_Bucket\_Set.append(b_1, b_2, ..., b_c)$
        $Bucket\_Set \leftarrow New\_Bucket\_Set$
    $j \leftarrow 1$
    for each bucket $b$ in $Bucket\_Set$ do
        $samples[j] \leftarrow$ one random sample from b
        $j \leftarrow j + 1$
    return $samples[1...k]$
```

We illustrate the algorithm of equi-depth stratified sampling through an example in Figure 3. In the example, we divide the data space into 16 grids, and each grid has nearly the same number of data points. Then we

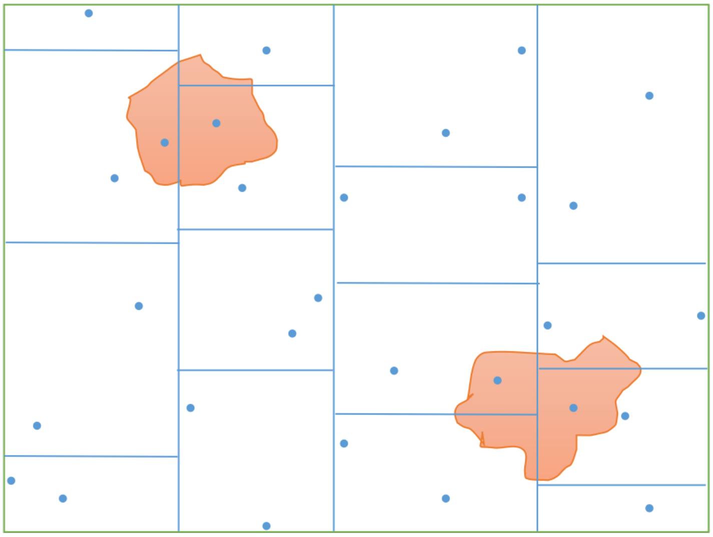


Figure 3: equi-depth stratified sampling

draw one random sample from each grid.

2.4 progressive sampling

Progressive sampling algorithm 4 is to perform equi-width or equi-depth stratified sampling level-by-level. In the first level, we divide each dimension into 2 buckets, so we have 2^d grids, and we select a random sample from each grid. When we go to the second level, we divide each dimension into 2^2 buckets, so we have 2^{2d} grids. Then we select one random sample from each of these smaller grids. We continue this sampling process until we get one sample from the user interest area.

Algorithm 4 Progressive Sampling

```
procedure PROGRESSIVE_SAMPLING
    $sample\_set = \{\}$
    $level \leftarrow 1$
    while no point in sample_set is within user interest area do
        $k \leftarrow 2^{level*d}$
        $samples \leftarrow SELECT\_EQUIWIDTH(k)$ or $SELECT\_EQUIDEPTH(k)$
        $sample\_set.append(samples)$
        $level \leftarrow level + 1$
  return $sample\_set$
```

2.5 discussion about the high dimensional problems

The equi-width, equi-depth stratified sampling and progressive sampling will generate a large number of grids(exponential to d), especially in high-dimensional space. Like in the SDSS astronomy data set we use, there are hundreds of dimensions for each data point. However, we don't need to divide the data space for each data dimension. For each data exploration task, the user may only be interested in a small subset of the dimensions, and only these dimensions are relevant to our sampling algorithm. The parameter d in the above algorithm will be the number of relevant attributes for the user. For example, if the data set has 200 dimensions, but in some data exploration task, the user is only interested in 5 of the attributes, then $d = 5$. We only need to divide the 5 dimensional space into grids, and draw samples from the grids there. This dramatically reduces the number of grids for our sampling algorithms.

3 Theoretical Analysis

We want to analyze the lower bound of the probability that within k samples selected, at least one sample is within the user interested area. Suppose our data space is d dimensional. The size of the minimum bounding box for our dataset in the d-dimensional space is A_t, and the total number of data points within A_t is N_t. We assume the total size of the user interest space is A_i, and the number of data points within A_i is N_i. When we refer to user interest space or user interest area below, we assume they are one or multiple d-dimensional space.

3.1 Random sampling lower bound

We select k distinct random samples according to algorithm 1. We are interested in the probability p_{random} that using random sampling, at least one of the k samples is within the interest area. The success probability in each trial is equal to $\frac{N_i}{N_t}$, so we know that the probability p_{random} will be:

$$p_{random} = 1 - (1 - \frac{N_i}{N_t})^k \tag{1}$$

If we assume that the ratio of the number of data points in the user interest area compared with the total number of data points in the data space is α,

$$\frac{N_i}{N_t} = \alpha \tag{2}$$

Then according to formula (1) and (2), we can get the lower bound for p_{random} with respect to α and k:

$$p_{random} = 1 - (1 - \frac{N_i}{N_t})^k = 1 - (1 - \alpha)^k \tag{3}$$

3.2 Equi-width and equi-depth stratified sampling lower bound

For equi-width and equi-depth stratified sampling, we assume that the user interest area overlaps with grids $G_1, G_2, ..., G_s$, the area for each of these grids are $A_{t1}, A_{t2}, ..., A_{ts}$, the number of data points within each of these grids are $N_{t1}, N_{t2}, ..., N_{ts}$. The interest area overlap with these grids are $A_{i1}, A_{i2}, ..., A_{is}$ and the number of points within each of these overlap areas are $N_{i1}, N_{i2}, ..., N_{is}$. Then the probability $p_{stratified}$

that using equi-width or equi-depth sampling, at least one sample from the k samples selected is within the interest area, is as below:

$$p_{stratified} = 1 - (1 - \frac{N_{i1}}{N_{t1}}) * (1 - \frac{N_{i2}}{N_{t2}}) * ... * (1 - \frac{N_{is}}{N_{ts}}) \quad (4)$$

If any of the s grids is fully covered by the user interest area, which means that there exists some j, $N_{ij} = N_{tj}$ and $1 \leq j \leq s$, then we know that $1 - \frac{N_{ij}}{N_{tj}} = 0$, and $p_{stratified} = 1$. Otherwise, there is no grid that is fully covered by the user interest area, then we have $0 < N_{ij} < N_{tj}$ for any $1 \leq j \leq s$. In this case, we have $0 < 1 - \frac{N_{ij}}{N_{tj}} < 1$ for any $1 \leq j \leq s$. As the probability for the first case is trivial, we only consider the case when there is no grid fully covered by the user interest area in the analysis below.

We want to derive the lower bound of $p_{stratified}$. First, we can use Jensen's inequality.

Jensen's inequality

Let $x_1, x_2, ..., x_n \in \mathbb{R}$, $a_1, a_2, ..., a_n \geq 0$, and satisfy $a_1 + a_2 + ... + a_n = 1$. If $F(x)$ is a convex function with one variable x, according to Jensen's inequality, we have

$$F(a_1x_1 + a_2x_2 + ... + a_nx_n) \leq a_1F(x_1) + a_2F(x_2) + ... + a_nF(x_n) \quad (5)$$

If $F(x)$ is a concave function with one variable x, according to Jensen's inequality, we have

$$F(a_1x_1 + a_2x_2 + ... + a_nx_n) \geq a_1F(x_1) + a_2F(x_2) + ... + a_nF(x_n) \quad (6)$$

If we let $F(x)$ be a log function, $F(x) = log(x)$, then $F(x)$ is a concave function. If we let $a_1 = a_2 = ... = a_n = \frac{1}{n}$, we will have

$$log(\frac{1}{n}x_1 + \frac{1}{n}x_2 + ... + \frac{1}{n}x_n) \geq \frac{1}{n}log(x_1) + \frac{1}{n}log(x_2) + ... + \frac{1}{n}log(x_n) \quad (7)$$

$$log(\frac{x_1 + x_2 + ... + x_n}{n}) \geq log((x_1 * x_2 * ... * x_n)^{\frac{1}{n}}) \quad (8)$$

As log function is an increasing function, we have

$$\frac{x_1 + x_2 + ... + x_n}{n} \geq (x_1 * x_2 * ... * x_n)^{\frac{1}{n}} \quad (9)$$

$$x_1 * x_2 * ... * x_n \leq (\frac{x_1 + x_2 + ... + x_n}{n})^n \quad (10)$$

If we let $x_j = 1 - \frac{N_{ij}}{N_{tj}}$ in the equation (10) above, where $1 \leq j \leq s$, we know that $0 < x_j < 1$, which is valid for log function. According to equation (10), we have

$$(1 - \frac{N_{i1}}{N_{t1}}) * (1 - \frac{N_{i2}}{N_{t2}}) * ... * (1 - \frac{N_{is}}{N_{ts}}) \leq (\frac{(1 - \frac{N_{i1}}{N_{t1}}) + (1 - \frac{N_{i2}}{N_{t2}}) + ... + (1 - \frac{N_{is}}{N_{ts}})}{s})^s \quad (11)$$

We can simplify the right-hand side of the inequality (11) as below:

$$(\frac{s - (\frac{N_{i1}}{N_{t1}} + \frac{N_{i2}}{N_{t2}} + ... + \frac{N_{is}}{N_{ts}})}{s})^s = (1 - \frac{\frac{N_{i1}}{N_{t1}} + \frac{N_{i2}}{N_{t2}} + ... + \frac{N_{is}}{N_{ts}}}{s})^s \quad (12)$$

Therefore, we have

$$(1-\frac{N_{i1}}{N_{t1}})*(1-\frac{N_{i2}}{N_{t2}})*...*(1-\frac{N_{is}}{N_{ts}}) \leq (1-\frac{\frac{N_{i1}}{N_{t1}}+\frac{N_{i2}}{N_{t2}}+...+\frac{N_{is}}{N_{ts}}}{s})^s \tag{13}$$

$$p_{stratified} = 1-(1-\frac{N_{i1}}{N_{t1}})*(1-\frac{N_{i2}}{N_{t2}})*...*(1-\frac{N_{is}}{N_{ts}}) \geq 1-(1-\frac{\frac{N_{i1}}{N_{t1}}+\frac{N_{i2}}{N_{t2}}+...+\frac{N_{is}}{N_{ts}}}{s})^s \tag{14}$$

Formula (14) is general for both equi-depth and equi-width stratified sampling, and we can further make use of the properties of equi-depth and equi-width sampling methods to simplify it and get the lower bound for p_{depth} and p_{width}.

Equi-depth stratified sampling lower bound

We want to derive the lower bound for the probability p_{depth} that we can get at least one sample within the user interest area from the k samples selected using equi-depth stratified sampling method. When we use equi-depth stratified sampling method, we can assume that each grid has the same number of data points, so

$$N_{t1} = N_{t2} = ... = N_{ts} = \frac{1}{k}N_t \tag{15}$$

Then using equation (15), we can simplify the right-hand side of inequality (14) as below:

$$1-(1-\frac{\frac{N_{i1}}{N_{t1}}+\frac{N_{i2}}{N_{t2}}+...+\frac{N_{is}}{N_{ts}}}{s})^s = 1-(1-\frac{\frac{N_{i1}+N_{i2}+...+N_{is}}{\frac{1}{k}N_t}}{s})^s = 1-(1-\frac{k}{s}*\frac{N_i}{N_t})^s \tag{16}$$

As we assume the data ratio $\frac{N_i}{N_t}$ is α in formula (2), we can get

$$1-(1-\frac{k}{s}*\frac{N_i}{N_t})^s = 1-(1-\frac{k}{s}*\alpha)^s \tag{17}$$

Combining formula (14), (16) and (17), we will have

$$p_{depth} = 1-(1-\frac{N_{i1}}{N_{t1}})*(1-\frac{N_{i2}}{N_{t2}})*...*(1-\frac{N_{is}}{N_{ts}}) \geq \boldsymbol{1-(1-\frac{k}{s}*\alpha)^s} \tag{18}$$

If we define $f(x)$ as equation (19), then from formula (18) above, we know that the lower bound for equi-depth stratified sampling will be $f(s)$ as in formula (20).

$$f(x) = 1-(1-\frac{k\alpha}{x})^x \tag{19}$$

$$p_{depth} \geq 1-(1-\frac{k}{s}*\alpha)^s = f(s) \tag{20}$$

Analysis of the lower bound $f(s)$

We want to analyze the relationship between the lower bound $f(s)$ and the value $f(k)$. We can show that only when $s = k$, $f(s) = f(k)$, otherwise, when $s < k$, $f(x)$ is a decreasing function in the range $[s, k]$, so $f(s) > f(k)$. In sum, we have formula (21), and the equality is true only when $s = k$.

$$f(s) \geq f(k) = 1 - (1 - \alpha)^k \tag{21}$$

In order to prove that $f(x)$ is a decreasing function in the range $[s, k]$ when $s < k$, we only need to prove that $g(x)$ is an increasing function in the range $[s, k]$.

$$g(x) = (1 - \frac{k\alpha}{x})^x \tag{22}$$

First, we can prove that $k\alpha \leq s$.

We know that the number of points within each interest overlap area is smaller than or equal to the total number of points within that grid, so we have

$$N_{i1} \leq N_{t1}, N_{i2} \leq N_{t2}, ..., N_{is} \leq N_{ts} \tag{23}$$

And as a result,

$$N_{i1} + N_{i2} + ... + N_{is} \leq N_{t1} + N_{t2} + ... + N_{ts} \tag{24}$$

As we are using equi-depth stratified sampling, we can combine the formula (15) and (24), and get

$$N_{i1} + N_{i2} + ... + N_{is} \leq \frac{s}{k} N_t \tag{25}$$

And the sum of the left-hand side terms in formula (25) is N_i, which should be greater than 0, so we have

$$0 < N_i \leq \frac{s}{k} N_t \tag{26}$$

If we divide both sides by N_t, we can get

$$0 < \frac{N_i}{N_t} \leq \frac{s}{k} \tag{27}$$

As we have the data ratio $\frac{N_i}{N_t}$ is α as in formula (2), together with equation (27), we can get

$$\alpha = \frac{N_i}{N_t} \leq \frac{s}{k} \tag{28}$$

Multiplying the left-hand side and right-hand side of formula (28) by k, we can get

$$k\alpha \leq s \tag{29}$$

Because $k\alpha \leq s$ from formula (29), when we analyze $g(x)$ in the range $[s, k]$, we know that $\frac{k\alpha}{x} \leq 1$, and the term inside the parenthesis is non-negative, so $g(x)$ is well-defined and has real value in the range $[s, k]$.

Next, we can show that $g(x)$ is continuous in the range $[s, k]$. As displayed below, $g(x)$ is right-continuous when $x = s$, left-continuous when $x = k$, and continuous when $x \in (s, k)$. Therefore, $g(x)$ is continuous in the range $[s, k]$.

$$\lim_{x \to s^+} g(x) = g(s) = (1 - \frac{k\alpha}{s})^s \tag{30}$$

$$\lim_{x \to k^-} g(x) = g(k) = (1 - \frac{k\alpha}{k})^k \tag{31}$$

$$\lim_{x \to t} g(x) = g(t) = (1 - \frac{k\alpha}{t})^t, t \in (s, k) \tag{32}$$

Third, we can show that $g'(x) > 0$ for $x > s$.

For simplicity, we can let $k\alpha = b$, and write $g(x)$ in formula (22) as

$$g(x) = (1 - \frac{b}{x})^x \tag{33}$$

Then we can get the derivative of $g(x)$ as

$$g'(x) = (1 - \frac{b}{x})^x (log(1 - \frac{b}{x}) + \frac{b}{x - b}) \tag{34}$$

We know that $s \geq k\alpha = b$, so when $x > s$, we can get $x > b > 0$. Therefore, the first term $(1 - \frac{b}{x})^x$ in equation (34) is positive, so we only need to consider the second term. We can denote the second term as $m(x)$ as below, and we can show that $m(x) > 0$ when $x > b$.

$$m(x) = log(1 - \frac{b}{x}) + \frac{b}{x - b} \tag{35}$$

The derivative of $m(x)$ is

$$m'(x) = \frac{\frac{b}{x^2}}{1 - \frac{b}{x}} - \frac{b}{(x - b)^2} = \frac{b}{x^2 - bx} - \frac{b}{(x - b)^2} = \frac{-b^2}{x(x - b)^2} \tag{36}$$

As $x > b > 0$, we can see that $m'(x) < 0$, so $m(x)$ is a decreasing function in $(b, +\infty)$. We can also see that the limit value for $m(x)$ when $x \to +\infty$ is:

$$\lim_{x \to +\infty} m(x) = \lim_{x \to +\infty} (log(1 - \frac{b}{x}) + \frac{b}{x - b}) = 0 \tag{37}$$

Combining formula (36) and (37), we know that $m(x)$ is a decreasing function in $(b, +\infty)$, and the limit value is 0, so we know that $m(x) > 0$ in $(b, +\infty)$. Therefore, the second term in formula (34) is also positive. As both the first term and the second term are positive, we can get

$$g'(x) > 0, \text{when } x > s \tag{38}$$

Now, we can apply ***Lagrange mean value theorem***. As $g(x)$ is continuous on the closed interval $[s, k]$, and $g(x)$ is differentiable on the open interval (s, k), there exists some c in the open interval (s, k), such that

$$g'(c) = \frac{g(k) - g(s)}{k - s} \tag{39}$$

As $c \in (s, k)$, we know from (38) that $g'(c) > 0$. So according to equation (39), when $s < k$, we have $g(s) < g(k)$. Equivalently, $f(s) > f(k)$ when $s < k$. As we know that when $s = k$, $f(s) = f(k)$, we can get the formula (40), and the equality is true only when $s = k$.

$$f(s) \geq f(k) \tag{40}$$

Now we finish the proof for the lower bound in formula (21).

More analysis for the lower bound of equi-depth sampling

Interestingly, we notice that $f(k)$ is equivalent to the lower bound of random sampling. The lower bound for equi-depth stratified sampling($f(s)$ as in the formula (20)) will be equal to $f(k)$ only when $s == k$. However, in most real applications, the value of s (the number of grids that overlap with the user interest area) will be significantly smaller than k (the total number of grids in the data space), we have $s < k$, so the lower bound $f(s)$ will be greater than $f(k)$.

We use an example to show the difference between $f(s)$ and $f(k)$. Suppose $k = 25$, $\alpha = 0.01$, then $k * \alpha = 0.25$, and $s \in [1, 25]$. We have $f(k) = 1-(1-\alpha)^k = 0.2222$ and $f(s) = 1-(1-\frac{k}{s} * \alpha)^s = 1-(1-\frac{0.25}{s})^s$ as noted in formula (20). We plot the value of $f(s)$ against the value of s in Figure 4 below. We can see that $f(s)$ is a decreasing function, and the y value is above $f(k) = 0.2222$. But when the value of s increases, the y value becomes closer and closer to 0.2222. We can also see that when $s = 1$, the y value is 0.25, and $\frac{f(1)}{f(k)} = \frac{0.25}{0.2222} = 1.125$.

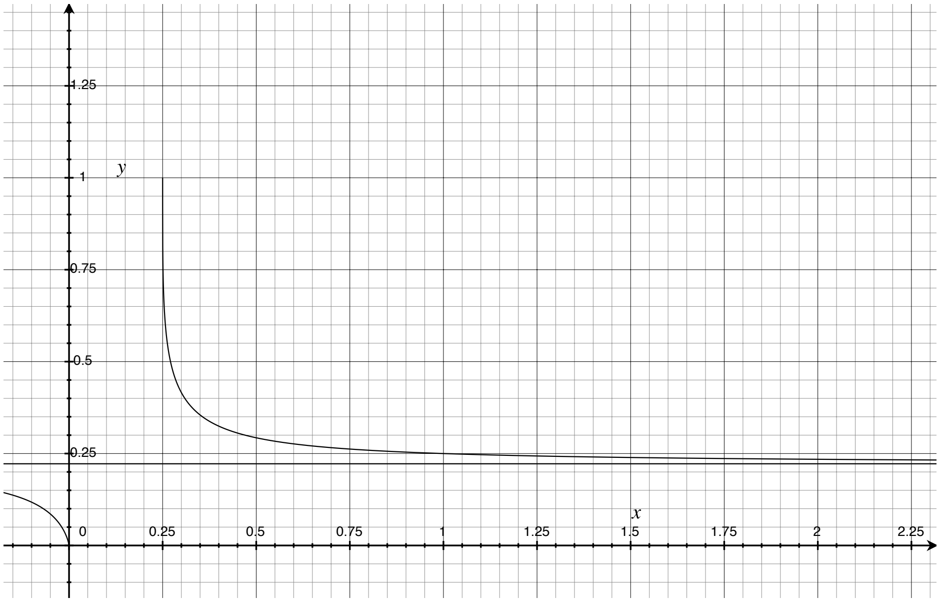


Figure 4: lower bound comparison 1

Then we assign different values for the parameters, $k = 100$ and $\alpha = 0.00691$, so $k * \alpha = 0.691$. We have $f(k) = 1-(1-\alpha)^k = 0.5$. We plot the value of $f(s) = 1-(1-\frac{0.691}{s})^s$ against the value of s in Figure 5 below. Again, we can see that $f(s)$ is a decreasing function against s, but the y value is above $f(k)$. When $s = 1$, the y value is 0.691, and $\frac{f(1)}{f(k)} = \frac{0.691}{0.5} = 1.382$. When $s = 2$, the y value is 0.572, and $\frac{f(2)}{f(k)} = \frac{0.572}{0.5} = 1.144$.

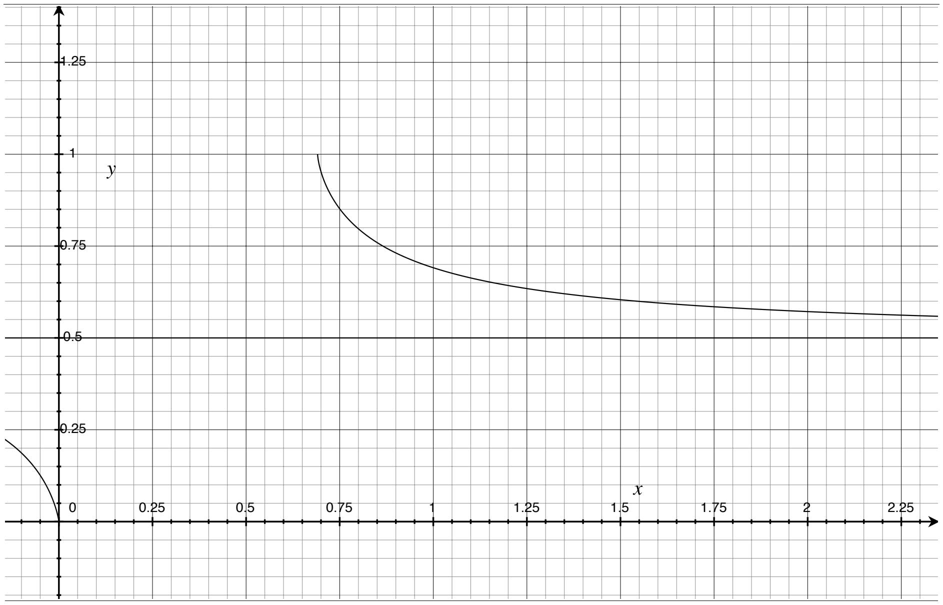


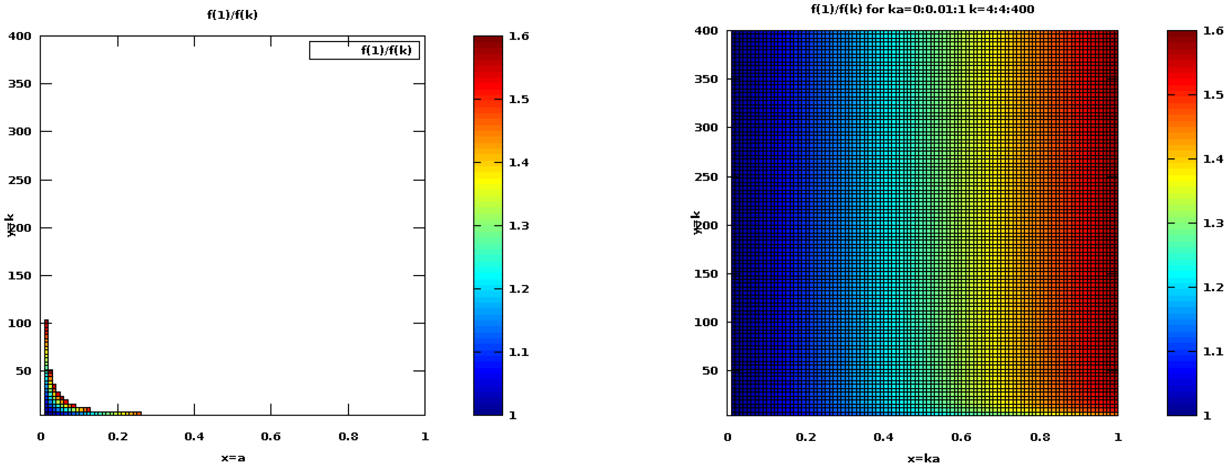
Figure 5: lower bound comparison 2

From the two examples above, we can see that when we change the value of k and α, the ratio $\frac{f(s)}{f(k)}$ also changes. We want to see the influence of k and α on the ratio $\frac{f(s)}{f(k)}$. There are three variables: s, the number of grids that overlap with the user interest areas, k, the total number of grids we divide the data space into, and α, the ratio of the number of points within the user interest area compared with the total number of points in the data space.

We generate the heatmap plot for $\frac{f(s)}{f(k)}$ when the value of s is equal to 1, 2, 3, 4, 20. In each of these plots, we change the values of k and α. There are some relationships that the values of s, k and α must hold. (1) $s \leq k$. It is obvious that the number of grids that overlap with the user interest area cannot exceed the total number of grids. When $s = 1, 2, 3, 4$, we assign the value range for k to be $[4, 400]$. When $s = 20$, the value range for k we assign is $[20, 400]$. (2) $k * \alpha \leq s$. We have proved this relationship in formula (29). The intuitive explanation is that in equi-depth stratified sampling, each grid has the same number of data points. Even if the user interest area covers all the data points in the s grids, the data ratio α cannot exceed $\frac{s}{k}$, so $k\alpha \leq s$. This relationship explains why there are values only in part of the plot in Figure 6a, 7a, 8a, 9a, 10a, other points in these plots don't satisfy the relationship, thus are invalid.

We can see from Figure 6a, 7a, 8a, 9a, 10a that when the value of s increases, the size of the valid area in the plot increases, but the maximum value for the ratio $\frac{f(s)}{f(k)}$ decreases. The maximum value for $\frac{f(1)}{f(k)}$ can reach 1.6, but the maximum value for $\frac{f(20)}{f(k)}$ is very close to 1.0. We can see from Figure 6b, 7b, 8b, 9b, 10b that the value of $k\alpha$ determines the value of $\frac{f(s)}{f(k)}$. When $s = 1$, the maximum value for $\frac{f(s)}{f(k)}$ occurs when $k\alpha$ is 1. When $s > 1$, the maximum value for $\frac{f(s)}{f(k)}$ occurs when $k\alpha$ is about 1.5.

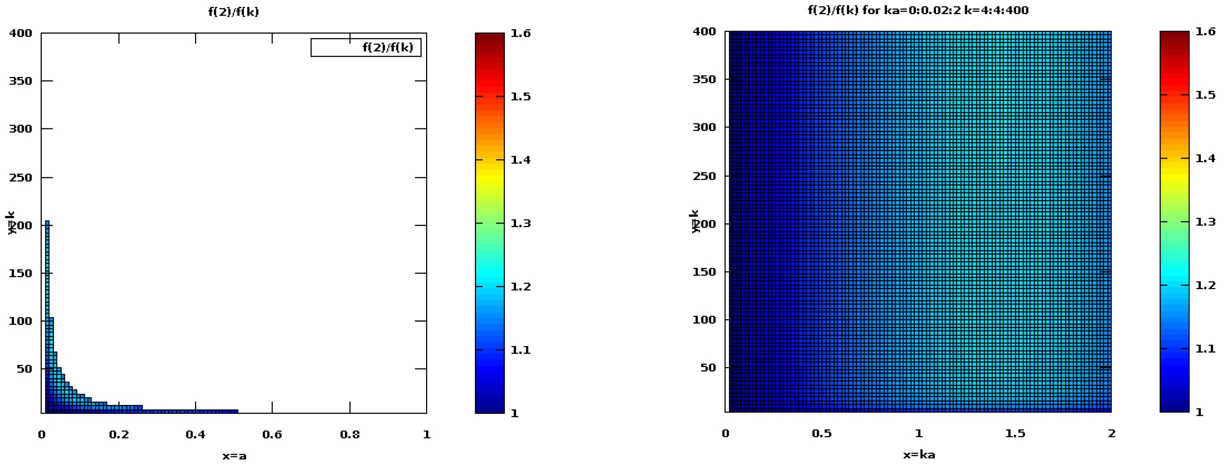
We can draw the conclusion from the analysis that it is the value of $k * \alpha$ that influences the value of $\frac{f(s)}{f(k)}$. The smaller the value of s, the larger the maximum value for $\frac{f(s)}{f(k)}$.



(a) f(1)/f(k) over a, k

(b) f(1)/f(k) over ka, k

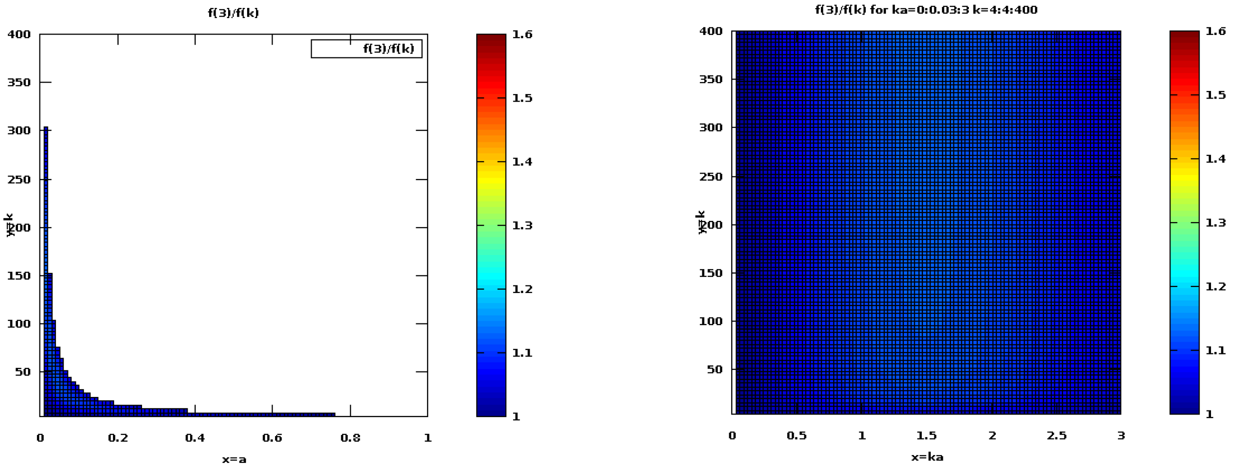
Figure 6: f(1)/f(k)



(a) f(2)/f(k) over a, k

(b) f(2)/f(k) over ka, k

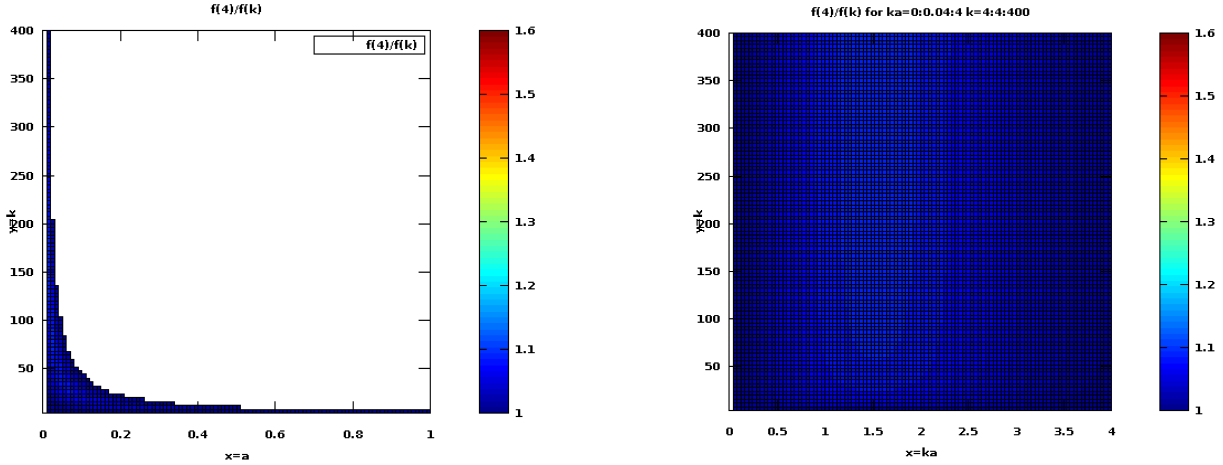
Figure 7: f(2)/f(k)



(a) f(3)/f(k) over a, k

(b) f(3)/f(k) over ka, k

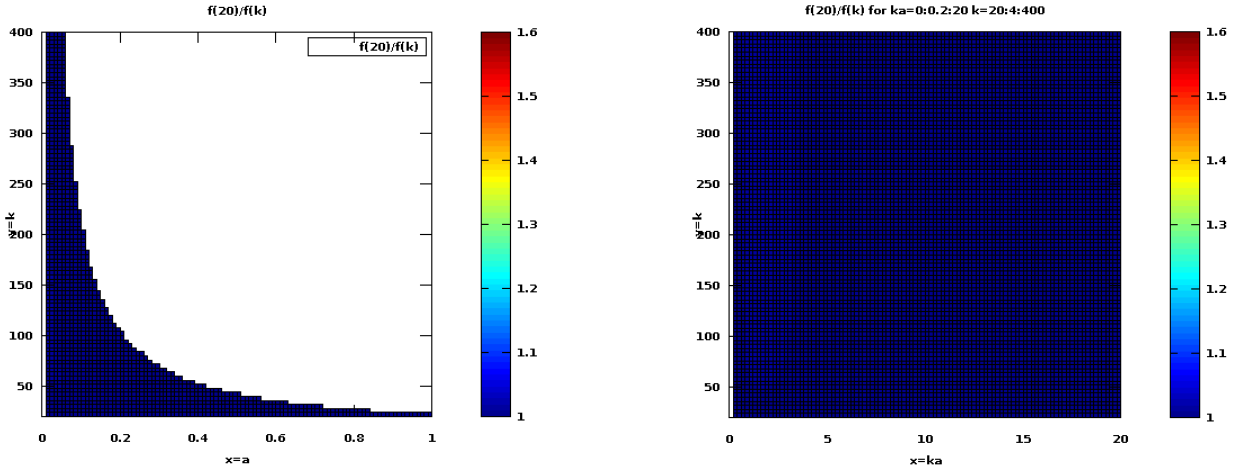
Figure 8: f(3)/f(k)



(a) f(4)/f(k) over a, k

(b) f(4)/f(k) over ka, k

Figure 9: f(4)/f(k)



(a) f(20)/f(k) over a, k

(b) f(20)/f(k) over ka, k

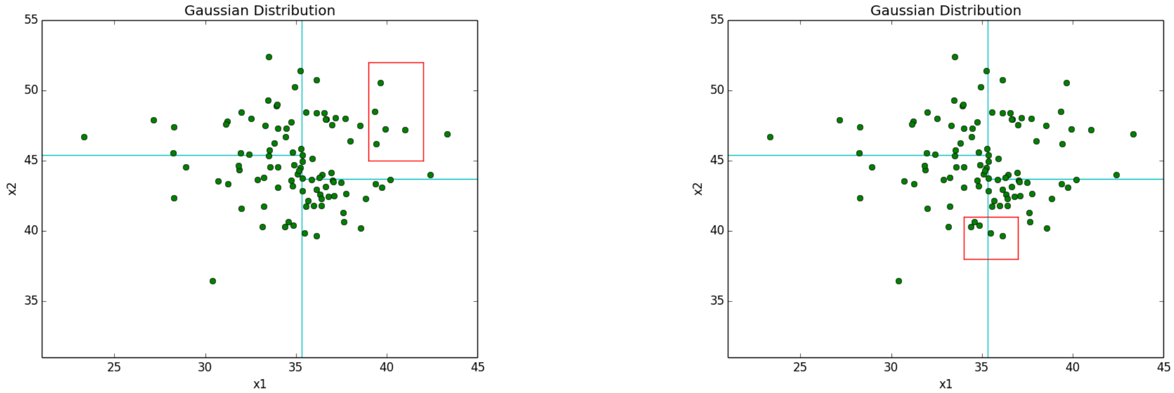
Figure 10: f(20)/f(k)

Some explanation of what it means that f(s) is a decreasing function

When we say $f(s)$ is a decreasing function with s, we assume the value of k and α is given, and we only change the value of s. As $\alpha = \frac{N_i}{N_t}$ and the data set size N_t is given, it is the same as we assume N_i, the number of data points within the user interest area, is fixed.

$$f(s) = 1 - (1 - \frac{k}{s}\alpha)^s = 1 - (1 - \frac{k}{s}\frac{N_i}{N_t})^s \tag{41}$$

This means that when we change s, we assume the data set size N_t, the number of grids k, and the number of points within the user interest area N_i, all stay constant. We give an example in Figure 11 below. In both Figure 11a and Figure 11b, $k = 4$, $N_t = 100$, and for both area 1 and area 2, $N_i = 5$. While for user interest area 1, $s = 1$, for user interest area 2, $s = 2$. And we can get $f(1) = 1 - (1 - \frac{4}{1} * \frac{5}{100})^1 = 0.2$ for area 1, and $f(2) = 1 - (1 - \frac{4}{2} * \frac{5}{100})^2 = 0.19$ for area 2. Therefore, $f(1) > f(2)$.

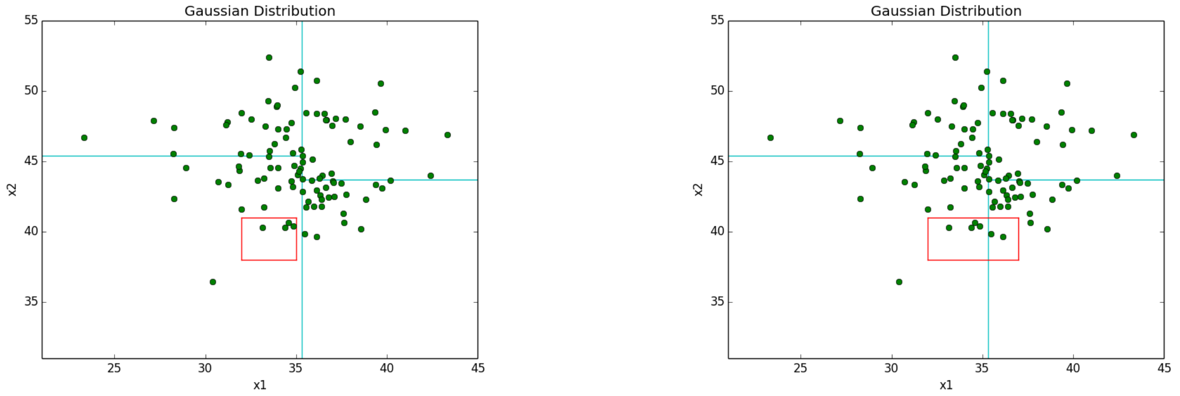


(a) area 1

(b) area 2

Figure 11: N_i is fixed

However, if the assumption doesn't hold, for example, if N_i, the number of points within the user interest area, increases when we increase the value of s, then $f(s)$ may increase as a result. We give an example in Figure 12 below. For user interest area 3 in Figure 12a, $N_i = 4$ and $s = 1$. After we expand the user interest area to be area 4 in Figure 12b, N_i becomes 6 and $s = 2$. In this scenario, $f(1) = 1 - (1 - \frac{4}{1} * \frac{4}{100})^1 = 0.16$, while $f(2) = 1 - (1 - \frac{4}{2} * \frac{6}{100})^2 = 0.2256$. Therefore, $f(1) < f(2)$.



(a) area 3

(b) area 4

Figure 12: N_i is not fixed

As a conclusion, $f(s)$ is a decreasing function for s when k, N_t and N_i are fixed. If N_i can increase when we increase s, the probability lower bound $f(s)$ is not necessarily a decreasing function.

Equi-width stratified sampling lower bound

When we use equi-width stratified sampling method, we can assume that the data distribution is known, so we know the density $\rho(t)$ for any point t in the data space. Then we can write the number of data points N_{ij} and N_{tj} as in formula (42) below, where $j = 1, 2, ..., s$.

$$N_{ij} = \int_{A_{ij}} \rho(t)dA, N_{tj} = \int_{A_{tj}} \rho(t)dA \tag{42}$$

We can write the data ratio as

$$\frac{N_{ij}}{N_{tj}} = \frac{\int_{A_{ij}} \rho(t)dA}{\int_{A_{tj}} \rho(t)dA} = \frac{A_{ij}\frac{\int_{A_{ij}} \rho(t)dA}{A_{ij}}}{A_{tj}\frac{\int_{A_{tj}} \rho(t)dA}{A_{tj}}} = \frac{A_{ij}}{A_{tj}} * \frac{\frac{\int_{A_{ij}} \rho(t)dA}{A_{ij}}}{\frac{\int_{A_{tj}} \rho(t)dA}{A_{tj}}} \tag{43}$$

If we define γ_j as in formula (44) below, for $1 \leq j \leq s$.

$$\gamma_j = \frac{\frac{\int_{A_{ij}} \rho(t)dA}{A_{ij}}}{\frac{\int_{A_{tj}} \rho(t)dA}{A_{tj}}} \tag{44}$$

Then according to (43), we have

$$\frac{N_{ij}}{N_{tj}} = \frac{A_{ij}}{A_{tj}}\gamma_j \tag{45}$$

As $j = 1, 2, ..., s$, we can write the formula (43) with γ_1, γ_2, ..., γ_s, and get

$$\frac{N_{i1}}{N_{t1}} = \frac{A_{i1}}{A_{t1}}\gamma_1, \frac{N_{i2}}{N_{t2}} = \frac{A_{i2}}{A_{t2}}\gamma_2, ..., \frac{N_{is}}{N_{ts}} = \frac{A_{is}}{A_{ts}}\gamma_s \tag{46}$$

If we let $\gamma' = \min(\gamma_1, \gamma_2, ..., \gamma_s)$, then we have

$$\begin{aligned} \frac{N_{i1}}{N_{t1}} + \frac{N_{i2}}{N_{t2}} + ... + \frac{N_{is}}{N_{ts}} &= \frac{A_{i1}}{A_{t1}}\gamma_1 + \frac{A_{i2}}{A_{t2}}\gamma_2 + ... + \frac{A_{is}}{A_{ts}}\gamma_s \\ &\geq \frac{A_{i1}}{A_{t1}}\gamma' + \frac{A_{i2}}{A_{t2}}\gamma' + ... + \frac{A_{is}}{A_{ts}}\gamma' \\ &= (\frac{A_{i1}}{A_{t1}} + \frac{A_{i2}}{A_{t2}} + ... + \frac{A_{is}}{A_{ts}})\gamma' \end{aligned} \tag{47}$$

When the number of total grids k increases, each grid will become smaller, and the difference between the user interest area and the union of the s overlap grids will get smaller as well. When k goes to infinity, we can think that the difference goes to zero, and each of the s overlap grids is totally covered by the user interest area. So in this asymptotic case, we can get formula (48) below, and have $\gamma_j = 1$ for $1 \leq j \leq s$. Therefore, in the asymptotic case, $\gamma' = \min(\gamma_1, \gamma_2, ..., \gamma_s) = 1$.

$$\lim_{k \to \infty} \gamma_j = \lim_{k \to \infty} \frac{\frac{\int_{A_{ij}} \rho(t)dA}{A_{ij}}}{\frac{\int_{A_{tj}} \rho(t)dA}{A_{tj}}} = \frac{\frac{\int_{A_{tj}} \rho(t)dA}{A_{tj}}}{\frac{\int_{A_{tj}} \rho(t)dA}{A_{tj}}} = 1 \tag{48}$$

When we are not in the asymptotic case, and k does not go to ∞, we can assume that inside the grids that overlap with the user interest area, the user interest area in each of these grids is relatively denser compared with the average density of the same grid, more precisely, if we assume $\frac{\int_{A_{ij}} \rho(t)dA}{A_{ij}} \geq \frac{\int_{A_{tj}} \rho(t)dA}{A_{tj}}$ for $1 \leq j \leq s$, then we can get $\gamma_j \geq 1$ for $1 \leq j \leq s$. Notice that j is between 1 and s, not between 1 and k. For those grids that don't overlap with the user interest areas, we don't care about them, even if they are very dense. As $\gamma' = \min(\gamma_1, \gamma_2, ..., \gamma_s)$, so we can get $\gamma' \geq 1$.

Considering both the asymptotic scenario and non-asymptotic scenario, we have formula (49) below.

$$\gamma' \geq 1 \tag{49}$$

Combining formula (47) and formula (49), we can get

$$\frac{N_{i1}}{N_{t1}} + \frac{N_{i2}}{N_{t2}} + ... + \frac{N_{is}}{N_{ts}} \geq \frac{A_{i1}}{A_{t1}} + \frac{A_{i2}}{A_{t2}} + ... + \frac{A_{is}}{A_{ts}} \tag{50}$$

As we use equi-width stratified sampling, we can assume the area in each grid is the same.

$$A_{t1} = A_{t2} = ... = A_{ts} = \frac{1}{k}A_t \tag{51}$$

So we can simplify the right-hand side of the inequality (14) as below:

$$\begin{aligned} 1 - (1 - \frac{\frac{N_{i1}}{N_{t1}} + \frac{N_{i2}}{N_{t2}} + ... + \frac{N_{is}}{N_{ts}}}{s})^s &\geq 1 - (1 - \frac{\frac{A_{i1}}{A_{t1}} + \frac{A_{i2}}{A_{t2}} + ... + \frac{A_{is}}{A_{ts}}}{s})^s \\ &= 1 - (1 - \frac{\frac{A_{i1}+A_{i2}+...+A_{is}}{\frac{1}{k}A_t}}{s})^s \\ &= 1 - (1 - \frac{k}{s} * \frac{A_i}{A_t})^s \end{aligned} \tag{52}$$

If we assume the area ratio, which is the size of the user interest areas compared with the size of the total area, is β,

$$\frac{A_i}{A_t} = \beta \tag{53}$$

then combining formula (52) and formula (53), we have

$$p_{width} \geq 1 - (1 - \frac{\frac{N_{i1}}{N_{t1}} + \frac{N_{i2}}{N_{t2}} + ... + \frac{N_{is}}{N_{ts}}}{s})^s \geq \boldsymbol{1 - (1 - \frac{k}{s}\beta)^s} \tag{54}$$

If we define $h(x)$ as formula (55) below, then from formula (54) above, we know that the probability lower bound for equi-width stratified sampling will be $h(s)$ as in formula (56) below.

$$h(x) = 1 - (1 - \frac{k}{x}\beta)^x \tag{55}$$

$$p_{width} \geq 1 - (1 - \frac{k}{s}\beta)^s = h(s) \tag{56}$$

Analysis of the lower bound $h(s)$

Similar to the proof for equi-depth stratified sampling, we can analyze the relationship between the lower bound $h(s)$ and the value $h(k)$. We can show that only when $s = k$, $h(s) = h(k)$, otherwise, when $s < k$, $h(x)$ is a decreasing function in the range $[s, k]$, so $h(s) > h(k)$. In sum, we have the formula (57), and the equality is true only when $s = k$. The proof is similar to that in equi-depth sampling, so we will skip the proof here.

$$h(s) \geq h(k) = 1 - (1 - \beta)^k \tag{57}$$

We can also compare $h(s)$ and $h(k)$ using the similar method in Figure 4, 5, 6, 7, 8, 9, 10. We will skip them here, too.

Comparing the probability lower bound for equi-width and equi-depth sampling

As in formula (18) and (54), the probability lower bound for equi-depth and equi-width stratified sampling is $f(s) = 1 - (1 - \frac{k}{s}\alpha)^s$ and $h(s) = 1 - (1 - \frac{k}{s}\beta)^s$ respectively. We have compared $f(s)$ and $h(s)$ with $f(k)$ and $h(k)$. Now we want to compare $f(s)$ and $h(s)$.

There are two differences between $f(s)$ and $h(s)$. First, the difference between α and β. α is equal to the data ratio $\frac{N_i}{N_t}$, while β is equal to the area ratio $\frac{A_i}{A_t}$, so the relationship between α and β relies on the relationship between $\frac{N_i}{N_t}$ and $\frac{A_i}{A_t}$. Second, the difference of the s value. When we divide the data space into the same number (k) of grids using equi-width or equi-depth sampling, the value s(the number of grids that overlap with the user interest area) for equi-width and equi-depth stratified sampling may be different. Both of the two differences depend on the data distribution and the number of grids (k) we divide the data space into, so we use some common data distribution and k value to compare the probability lower bound for equi-depth and equi-width sampling, $f(s)$ and $h(s)$.

(1) uniform distribution
If our data set is uniformly distributed, then the data ratio $\frac{N_i}{N_t}$ will be equal to the area ratio $\frac{A_i}{A_t}$, so we have $\alpha = \beta$. Moreover, in uniformly distributed dataset, if we divide the data space into k grids, then the grids

in equi-depth sampling will be the same as the grids in equi-width sampling, because each grid will have the same number of points as well as the same area size. Therefore, the s values in equi-depth sampling and equi-width samping are the same. According to the analysis above, the probability lower bounds in equi-depth sampling and equi-width sampling are the same, $f(s) == h(s)$.

(2) Gaussian distribution

To compare equi-depth and equi-width sampling methods for Gaussian distributed dataset, we generate a synthetic dataset following 2-d Gaussian distribution. The dataset contains 100 data points, so $N_t = 100$. The data space is: $x_1 \in [21, 45]$ and $x_2 \in [31, 55]$, so the size of the total data space area is $A_t = 24 * 24 = 576$. We generate two user interest areas: the first one(we call it 'area 1' below) locates at the relatively dense area($x_1 \in [28, 32]$ and $x_2 \in [44, 48]$), while the second one(we call it 'area 2' below) locates at the relatively sparse area($x_1 \in [28, 32]$ and $x_2 \in [35, 39]$). When applying equi-depth and equi-width sampling methods, we divide the data space into $k = 16$ grids.

For area 1, the size of the area is $A_i = 4 * 4 = 16$, and the number of points within the area is $N_i = 8$, therefore, $\alpha = \frac{N_i}{N_t} = \frac{8}{100} = 0.08$ and $\beta = \frac{A_i}{A_t} = \frac{16}{576} = 0.0278$. Because $\alpha > \beta$, the data ratio is greater than the area ratio for area 1, we can define area 1 as a relatively dense area. We apply both equi-depth stratified sampling(in Figure 13), and equi-width stratified sampling(in Figure 14) for area 1. From Figure 13, we can see that area 1 overlaps with $s = 3$ grids in equi-depth sampling, and from Figure 14, we can see that area 1 overlaps with only $s = 1$ grid in equi-width sampling. Therefore, for area 1, we can get the probability lower bound for equi-depth sampling $f(s) = 1 - (1 - \frac{k}{s}\alpha)^s = 1 - (1 - \frac{16}{3} * 0.08)^3 = 0.8115$, and the lower bound for equi-width sampling $h(s) = 1 - (1 - \frac{k}{s}\beta)^s = 1 - (1 - \frac{16}{1} * 0.0278)^1 = 0.4448$. For area 1, $f(s) > h(s)$.

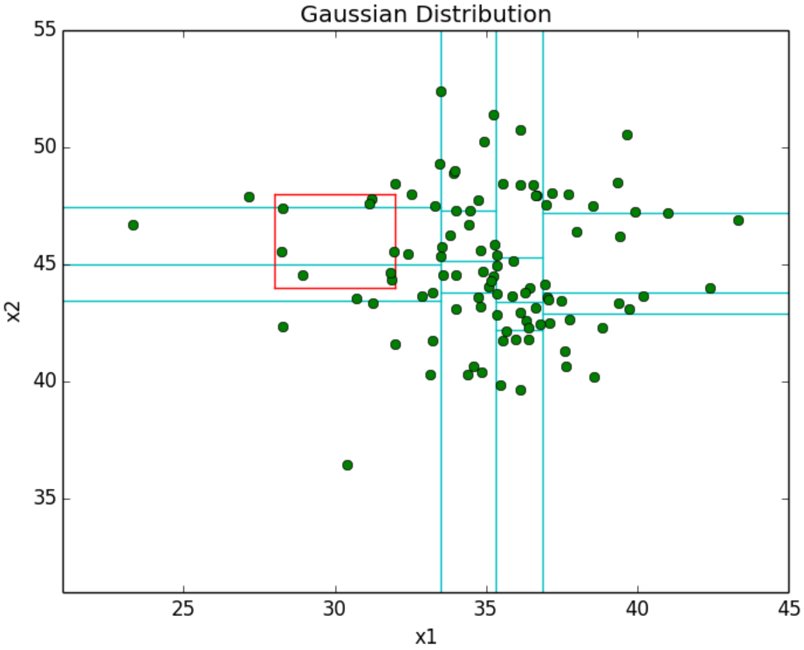


Figure 13: Equi-depth sampling for user interest area 1 under Gaussian distribution

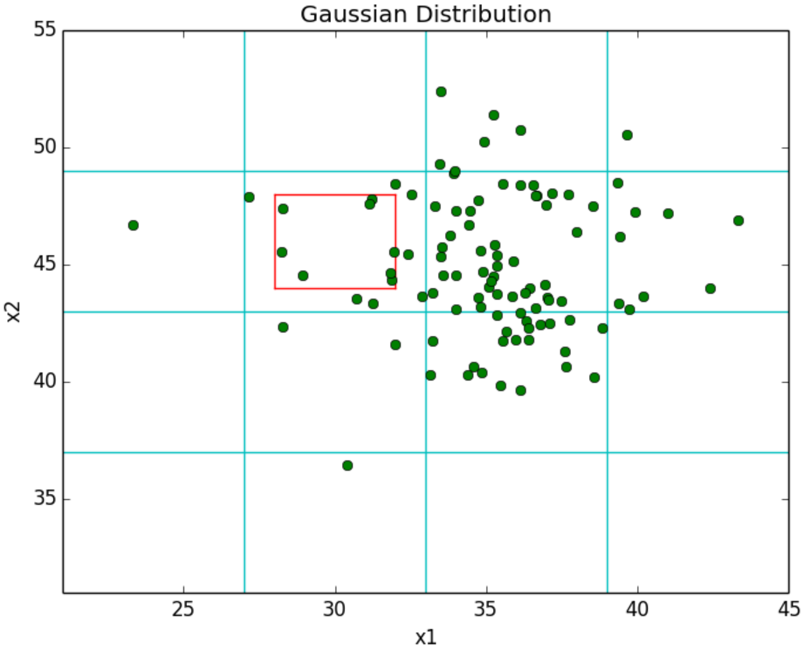


Figure 14: Equi-width sampling for user interest area 1 under Gaussian distribution

For area 2, the size of the area is $A_i = 4 * 4 = 16$, and the number of points within the area is $N_i = 1$, therefore, $\alpha = \frac{N_i}{N_t} = \frac{1}{100} = 0.01$ and $\beta = \frac{A_i}{A_t} = \frac{16}{576} = 0.0278$. Because $\alpha < \beta$, the data ratio is smaller than the area ratio for area 2, we can define area 2 as a relatively sparse area. The equi-depth stratified sampling and equi-width stratified sampling results for area 2 are shown in Figure 15 and Figure 16 respectively. From Figure 15, we can see that area 2 overlaps with $s = 1$ grid in equi-depth sampling, and from Figure 16, we can see that area 2 overlaps with $s = 2$ grids in equi-width sampling. Therefore, for area 2, we can get the probability lower bound for equi-depth sampling $f(s) = 1 - (1 - \frac{k}{s}\alpha)^s = 1 - (1 - \frac{16}{1} * 0.01)^1 = 0.16$, and the lower bound for equi-width sampling $h(s) = 1 - (1 - \frac{k}{s}\beta)^s = 1 - (1 - \frac{16}{2} * 0.0278)^2 = 0.3953$. For area 2, $f(s) < h(s)$.

In conclusion, for non-uniformly distributed(like Gaussian distributed) dataset, the relationship between $f(s)$ and $h(s)$ is not stable, which one is better depends on the density of the user interest area. If the user interest area is in some relatively dense area, where the data ratio $\alpha = \frac{N_i}{N_t}$ is greater than the area ratio $\beta = \frac{A_i}{A_t}$, then equi-depth stratified sampling will be preferred, even if the exact location of the user interest area is not known. Otherwise, if the user interest area locates at some relatively sparse area, where the data ratio $\alpha = \frac{N_i}{N_t}$ is smaller than the area ratio $\beta = \frac{A_i}{A_t}$, then equi-width stratified sampling might be a better choice, even if the exact location of the user interest area is not known.

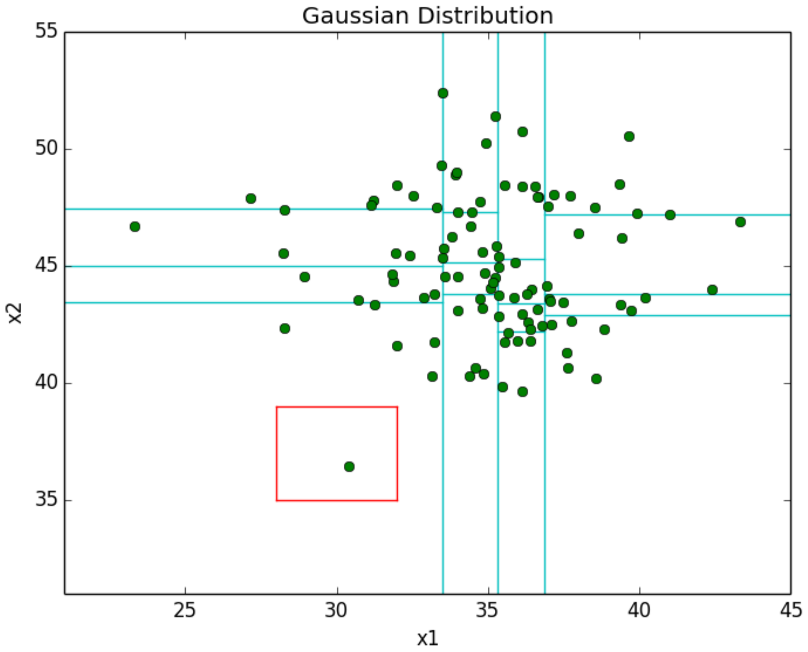


Figure 15: Equi-depth sampling for user interest area 2 under Gaussian distribution

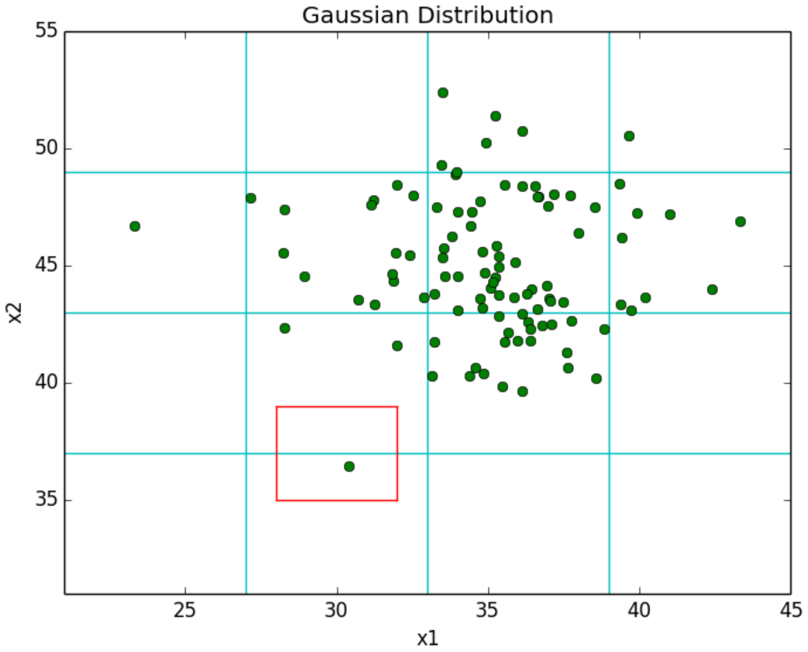


Figure 16: Equi-width sampling for user interest area 2 under Gaussian distribution

3.3 Progressive sampling lower bound

For progressive sampling, we use equi-depth or equi-width sampling method progressively, level by level. In level 1, we use equi-depth or equi-width stratified sampling method, dividing each dimension into 2 equi-depth or equi-width buckets, and select one random sample from each of the $k_1 = 2^d$ grids. If no sample is in the user interest area, we perform level 2 equi-depth or equi-width stratified sampling, dividing each dimension into 2^2 buckets, and select one random sample from each of the $k_2 = 2^{2d}$ grids. In level i equi-depth or equi-width stratified sampling, we divide each dimension into 2^i buckets, and select one random sample from each of the $k_i = 2^{i*d}$ grids. And so on. We stop when we get at least one sample within the user interest area.

Probability

If we let the probability that at level i, we get at least one positive sample with $k_i = 2^{i*d}$ samples to be $p(k_i)$, then the probability that we can get at least one positive sample within m levels in the progressive sampling process is:

$$p_m = 1 - (1 - p(k_1))(1 - p(k_2))...(1 - p(k_m)) \tag{58}$$

As we know the lower bound of $p(k_1)$, $p(k_2)$, ..., $p(k_m)$, which are single level equi-depth or equi-width sampling probability lower bound, we can get the lower bound for the probability p_m assuming we stop at level m.

Probability lower bound for progressive equi-depth sampling

For progressive equi-depth sampling, according to formula (20), the probability lower bounds for $p(k_1)$, $p(k_2)$, ..., $p(k_m)$ are in formula (59).

$$\begin{aligned} p(k_1) &\geq 1 - (1 - \frac{k_1}{s_1}\alpha)^{s_1} \\ p(k_2) &\geq 1 - (1 - \frac{k_2}{s_2}\alpha)^{s_2} \\ &... \\ p(k_m) &\geq 1 - (1 - \frac{k_m}{s_m}\alpha)^{s_m} \end{aligned} \tag{59}$$

Then we can derive the lower bound for p_m as in formula (60).

$$\begin{aligned} p_m =\ & 1 - (1 - p(k_1))(1 - p(k_2))...(1 - p(k_m)) \\ & \geq 1 - (1 - \frac{k_1}{s_1}\alpha)^{s_1}(1 - \frac{k_2}{s_2}\alpha)^{s_2}...(1 - \frac{k_m}{s_m}\alpha)^{s_m} \end{aligned} \tag{60}$$

According to the derivation result from Jensen's Inequality in formula (10), we can get

$$\begin{aligned} p' &= (1-\frac{k_1}{s_1}\alpha)^{s_1}(1-\frac{k_2}{s_2}\alpha)^{s_2}...(1-\frac{k_m}{s_m}\alpha)^{s_m} \\ &\leq (\frac{s_1(1-\frac{k_1}{s_1}\alpha)+s_2(1-\frac{k_2}{s_2}\alpha)+...+s_m(1-\frac{k_m}{s_m}\alpha)}{s_1+s_2+...+s_m})^{\sum_{i=1}^m s_i} \\ &= (\frac{\sum_{i=1}^m s_i-(\sum_{i=1}^m k_i)\alpha}{\sum_{i=1}^m s_i})^{\sum_{i=1}^m s_i} \\ &= (1-\frac{\sum_{i=1}^m k_i}{\sum_{i=1}^m s_i}\alpha)^{\sum_{i=1}^m s_i} \end{aligned} \tag{61}$$

Combining formula (60) and (61), we can get

$$p_m \geq 1-p' \geq 1-(1-\frac{\sum_{i=1}^m k_i}{\sum_{i=1}^m s_i}\alpha)^{\sum_{i=1}^m s_i} \tag{62}$$

Comparing with single level equi-depth sampling

The probability lower bound for progressive equi-depth sampling is in the right-hand side of formula (62). We want to compare it with the lower bound for single level equi-depth sampling. To make the comparison fair, we draw the same number of samples from either progressive equi-depth sampling or single level equi-depth sampling. In single level equi-depth sampling method, we divide the data space into k grids, and select one random sample from each grid, where k is

$$k = \sum_{i=1}^m k_i \tag{63}$$

When we divide the data space into k grids(k value is in formula (63)) according to single level equi-depth sampling algorithm, we assume that the number of grids that overlap with the user interest area is s'. Then according to formula (20), we know that the probability lower bound for the single level equi-depth sampling is in the right-hand side of formula (64).

$$p_{depth} \geq 1-(1-\frac{k}{s'}\alpha)^{s'} = 1-(1-\frac{\sum_{i=1}^m k_i}{s'}\alpha)^{s'} \tag{64}$$

We define L_1 as the probability lower bound of progressive equi-depth sampling(the right-hand side of formula (62)) and define L_2 as the probability lower bound of single level equi-depth sampling(the right-hand side of formula (64)), we need to compare L_1 and L_2.

$$L_1 = 1-(1-\frac{\sum_{i=1}^m k_i}{\sum_{i=1}^m s_i}\alpha)^{\sum_{i=1}^m s_i} \tag{65}$$

$$L_2 = 1-(1-\frac{\sum_{i=1}^m k_i}{s'}\alpha)^{s'} \tag{66}$$

The relationship between L_1 and L_2 depends on the relationship between $\sum_{i=1}^m s_i$ and s'. As we know from formula (19) that $f(x) = 1-(1-\frac{\sum_{i=1}^m k_i}{x}\alpha)^x$ is a decreasing function, if $\sum_{i=1}^m s_i < s'$, $L_1 > L_2$, if $\sum_{i=1}^m s_i > s'$, $L_1 < L_2$, and if $\sum_{i=1}^m s_i = s'$, $L_1 = L_2$.

To understand the relationship between $\sum_{i=1}^{m} s_i$ and s', we make use of the Gaussian distributed dataset we generated in the previous section. For progressive equi-depth sampling, we divide the data space into 4 grids(2 buckets in each dimension) in level 1, and we divide the data space into 16 grids(4 buckets in each dimension) in level 2. As a comparison, in single level equi-depth sampling, we divide the data space into 20 grids(5 buckets in dimension x_1 and 4 buckets in dimension x_2). We generate two user interest areas, area 1 and area 2. The location for area 1 is $[25, 29] * [46, 50]$, and the location for area 2 is $[35.5, 40] * [41.5, 49]$.

Figure 17 shows the progressive equi-depth sampling in level 1 for user interest area 1, and $s_1 = 1$. Figure 18 shows the progressive equi-depth sampling in level 2 for user interest area 1, and $s_2 = 2$. Figure 19 shows the single level equi-depth sampling for user interest area 1, and $s' = 2$. Therefore, for user interest area 1, we have $s_1 + s_2 > s'$, and $L_1 < L_2$.

Similarly, for user interest area 2, we can see that $s_1 = 2$ from Figure 20, $s_2 = 8$ from Figure 21, and $s' = 12$ from Figure 22. Therefore, for user interest area 2, we have $s_1 + s_2 < s'$, and $L_1 > L_2$.

From the result for user interest area 1 and area 2, we can see that the relationship between the probability lower bound of progressive equi-depth sampling and single level equi-depth sampling is not stable. As the data distribution influences how the grids are divided in equi-depth sampling, the performance of the two algorithms is related to the density of the user interest area. If the user interest area is in some sparse area(like area 1), the grids for single level equi-depth is not quite different from the grids in the highest level(level 2 in the example) in progressive sampling in the sparse area, so s' is almost the same as s_m, and smaller than $\sum_{i=1}^{m} s_i$. In this case, single level equi-depth sampling may have a greater probability lower bound than progressive equi-depth sampling. On the other hand, if the use interest area is in some dense area(like area 2), the additional grids will make s' greater than s_m, and even greater than $\sum_{i=1}^{m} s_i$. In this scenario, progressive equi-depth sampling may have a larger probability lower bound than the single level equi-depth sampling for user interest area in some dense region.

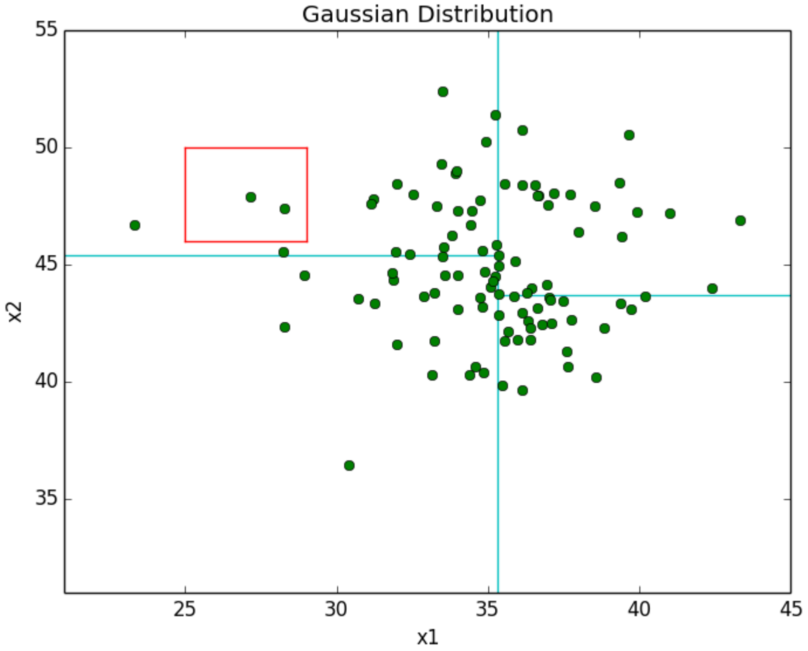


Figure 17: Progressive equi-depth sampling in level 1 for user interest area 1

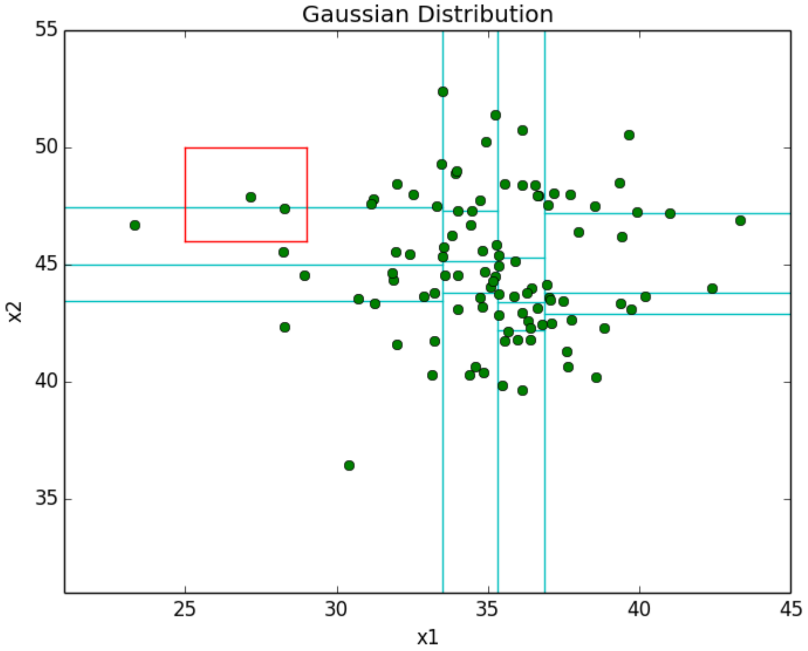


Figure 18: Progressive equi-depth sampling in level 2 for user interest area 1

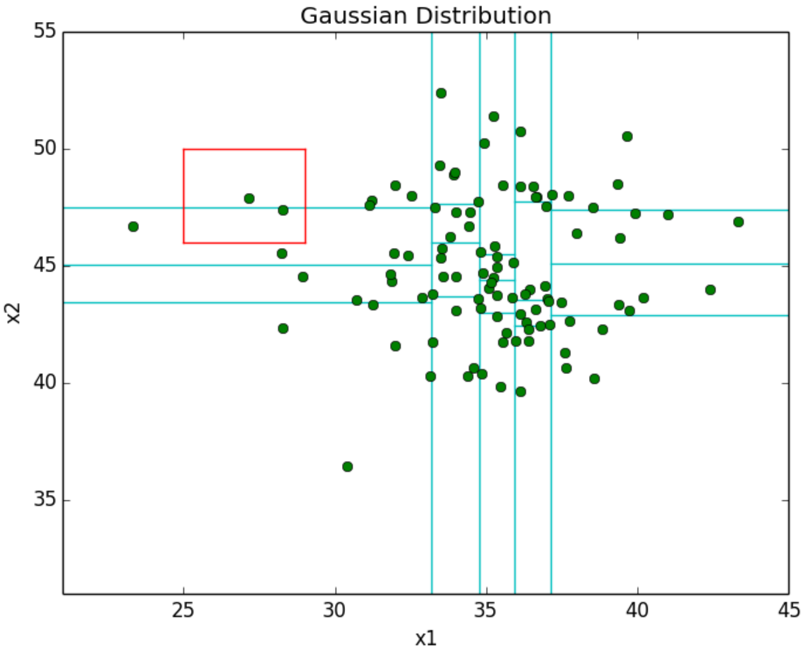


Figure 19: Single level equi-depth sampling for user interest area 1

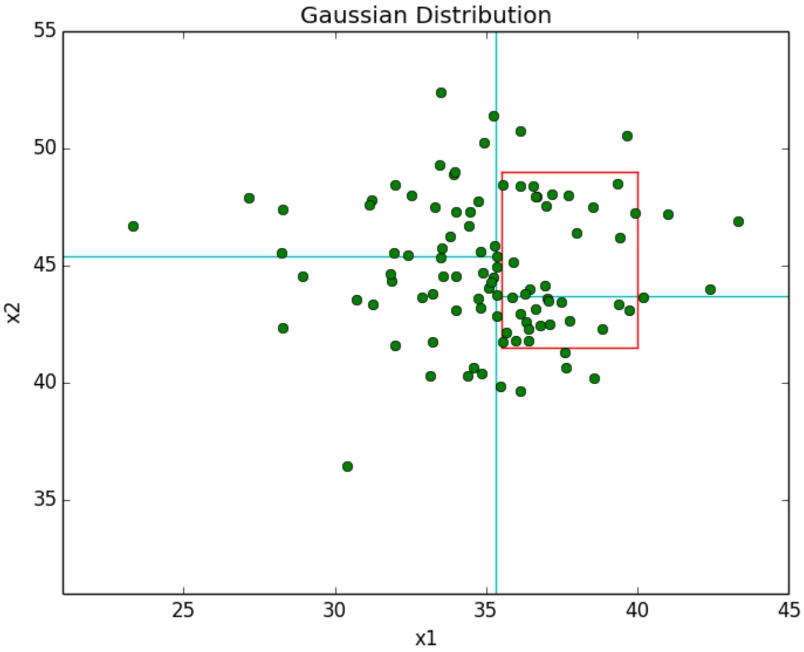


Figure 20: Progressive equi-depth sampling in level 1 for user interest area 2

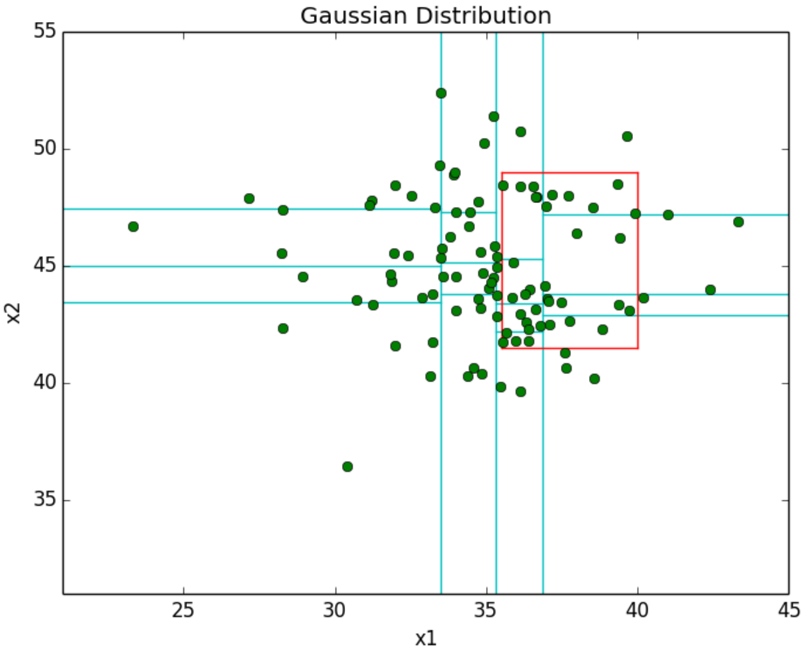


Figure 21: Progressive equi-depth sampling in level 2 for user interest area 2

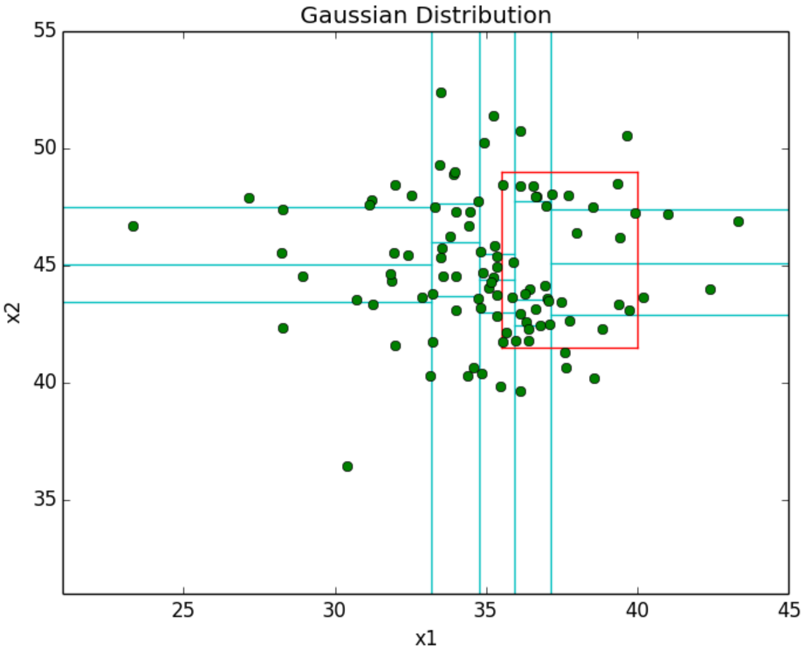


Figure 22: Single level equi-depth sampling for user interest area 2

Probability lower bound for progressive equi-width sampling

For progressive equi-width sampling, according to formula (56), the probability lower bounds for $p(k_1)$, $p(k_2)$, ..., $p(k_m)$ are in formula (67).

$$\begin{aligned} p(k_1) &\geq 1 - (1 - \frac{k_1}{s_1}\beta)^{s_1} \\ p(k_2) &\geq 1 - (1 - \frac{k_2}{s_2}\beta)^{s_2} \\ &... \\ p(k_m) &\geq 1 - (1 - \frac{k_m}{s_m}\beta)^{s_m} \end{aligned} \tag{67}$$

According to (58) and (67), we can get the lower bound for p_m as in formula (68).

$$\begin{aligned} p_m =&\ 1 - (1 - p(k_1))(1 - p(k_2))...(1 - p(k_m)) \\ &\geq 1 - (1 - \frac{k_1}{s_1}\beta)^{s_1}(1 - \frac{k_2}{s_2}\beta)^{s_2}...(1 - \frac{k_m}{s_m}\beta)^{s_m} \end{aligned} \tag{68}$$

According to the derivation result from Jensen's Inequality in formula (10), we can get

$$\begin{aligned} p'' &= (1 - \frac{k_1}{s_1}\beta)^{s_1}(1 - \frac{k_2}{s_2}\beta)^{s_2}...(1 - \frac{k_m}{s_m}\beta)^{s_m} \\ &\leq (\frac{s_1(1 - \frac{k_1}{s_1}\beta) + s_2(1 - \frac{k_2}{s_2}\beta) + ... + s_m(1 - \frac{k_m}{s_m}\beta)}{s_1 + s_2 + ... + s_m})^{\sum_{i=1}^{m} s_i} \\ &= (\frac{\sum_{i=1}^{m} s_i - (\sum_{i=1}^{m} k_i)\beta}{\sum_{i=1}^{m} s_i})^{\sum_{i=1}^{m} s_i} \\ &= (1 - \frac{\sum_{i=1}^{m} k_i}{\sum_{i=1}^{m} s_i}\beta)^{\sum_{i=1}^{m} s_i} \end{aligned} \tag{69}$$

Combining formula (68) and (69), we can get

$$p_m \geq 1 - p'' \geq 1 - (1 - \frac{\sum_{i=1}^{m} k_i}{\sum_{i=1}^{m} s_i}\beta)^{\sum_{i=1}^{m} s_i} \tag{70}$$

Comparing with single level equi-width sampling

Similar to the analysis in the comparison between progressive equi-depth sampling and single level equi-depth sampling, we want to compare the lower bound for progressive equi-width sampling in formula (70) with the lower bound for single level equi-width sampling. To make the comparison fair, we draw the same number of samples from single level equi-width sampling. In single level equi-width sampling, we divide the data space into k grids, and select one random sample from each grid, where k is

$$k = \sum_{i=1}^{m} k_i \tag{71}$$

We also assume that the number of grids that overlap with the user interest area in the single level equi-width sampling method is s''. Then according to formula (56), we can get the probability lower bound for the single level equi-width sampling:

$$p_{width} \geq 1 - (1 - \frac{k}{s''}\beta)^{s''} = 1 - (1 - \frac{\sum_{i=1}^{m} k_i}{s''}\beta)^{s''} \tag{72}$$

We define L_3 as the probability lower bound of progressive equi-width sampling (the right-hand side of formula (70)), and define L_4 as the probability lower bound of single level equi-width sampling (the right-hand side of formula (72)), we need to compare L_3 and L_4.

$$L_3 = 1 - (1 - \frac{\sum_{i=1}^{m} k_i}{\sum_{i=1}^{m} s_i}\beta)^{\sum_{i=1}^{m} s_i} \tag{73}$$

$$L_4 = 1 - (1 - \frac{\sum_{i=1}^{m} k_i}{s''}\beta)^{s''} \tag{74}$$

The relationship between L_3 and L_4 depends on the relationship between $\sum_{i=1}^{m} s_i$ and s''. As we know from formula (55) that $h(x) = 1 - (1 - \frac{\sum_{i=1}^{m} k_i}{x}\beta)^x$ is a decreasing function, if $\sum_{i=1}^{m} s_i < s''$, $L_3 > L_4$, if $\sum_{i=1}^{m} s_i > s''$, $L_3 < L_4$, and if $\sum_{i=1}^{m} s_i = s''$, $L_3 = L_4$.

To understand the relationship between $\sum_{i=1}^{m} s_i$ and s'', we make use of the same Gaussian distributed dataset as in previous section. For progressive equi-width sampling, we divide the data space into 4 grids(2 buckets in each dimension) in level 1, and we divide the data space into 16 grids(4 buckets in each dimension) in level 2. As a comparison, in single level equi-width sampling, we divide the data space into 20 grids(5 buckets in dimension x_1 and 4 buckets in dimension x_2). We also generate two user interest areas, area 3 and area 4. The location for area 3 is $[24, 28] * [47, 51]$, and the location for area 4 is $[28, 32] * [35, 39]$. The size of area 3 and area 4 are the same, and there is only one data point in both area 3 and area 4, so the data density for area 3 and area 4 are also the same.

Figure 23 shows the progressive equi-width sampling in level 1 for user interest area 3, and $s_1 = 1$. Figure 24 shows the progressive equi-width sampling in level 2 for user interest area 3, and $s_2 = 4$. Figure 25 shows the single level equi-width sampling for user interest area 3, and $s'' = 4$. Therefore, for user interest area 3, we have $s_1 + s_2 > s''$, and $L_3 < L_4$.

Similarly, for user interest area 4, we can see that $s_1 = 1$ from Figure 26, $s_2 = 2$ from Figure 27, and $s'' = 4$ from Figure 28. Therefore, for user interest area 2, we have $s_1 + s_2 < s''$, and $L_3 > L_4$.

From the result for user interest area 3 and area 4, we can see that the relationship between the probability lower bound of progressive equi-width sampling and single level equi-width sampling is not stable. As the data distribution does not influence the way grids are divided in equi-width sampling, whether the user interest area is in sparse area or dense area does not really matter in the relationship between the lower bound for progressive equi-width sampling and the lower bound for single level equi-width sampling. What really matters is the relative location of the user interest area in the data space. For user interest areas in some locations, where $\sum_{i=1}^{m} s_i < s''$, progressive equi-width sampling has higher probability lower bound, while for some other locations, where $\sum_{i=1}^{m} s_i > s''$, single level equi-width sampling has higher probability lower bound. In conclusion, we cannot say which one, progressive equi-width sampling or single level equi-width sampling, is better, and the result depends on the location of the user interest area.

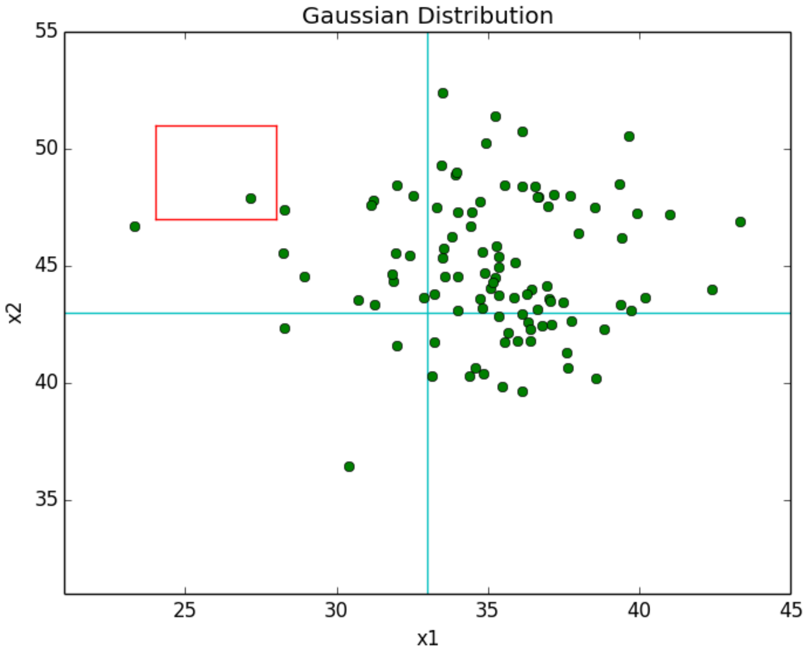


Figure 23: Progressive equi-width sampling in level 1 for user interest area 3

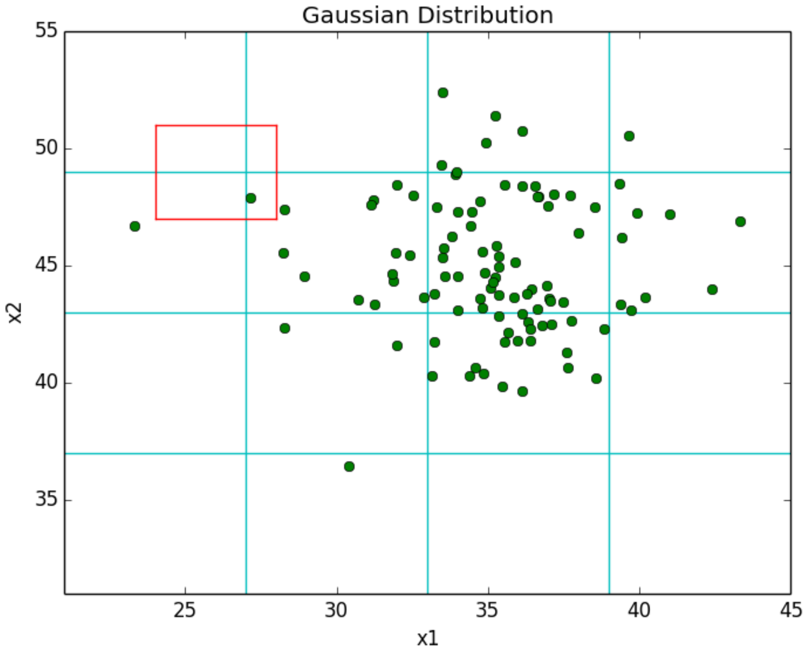


Figure 24: Progressive equi-width sampling in level 2 for user interest area 3

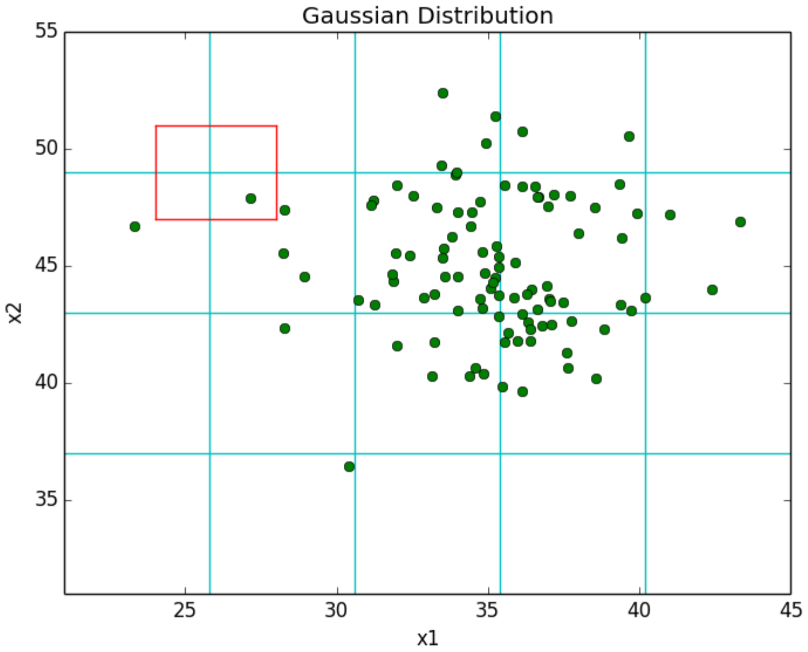


Figure 25: Single level equi-width sampling for user interest area 3

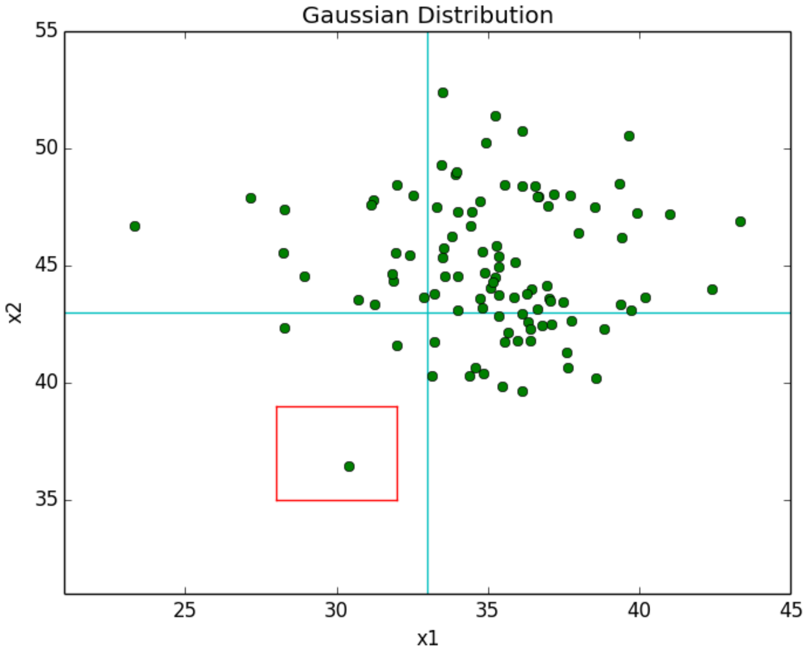


Figure 26: Progressive equi-width sampling in level 1 for user interest area 4

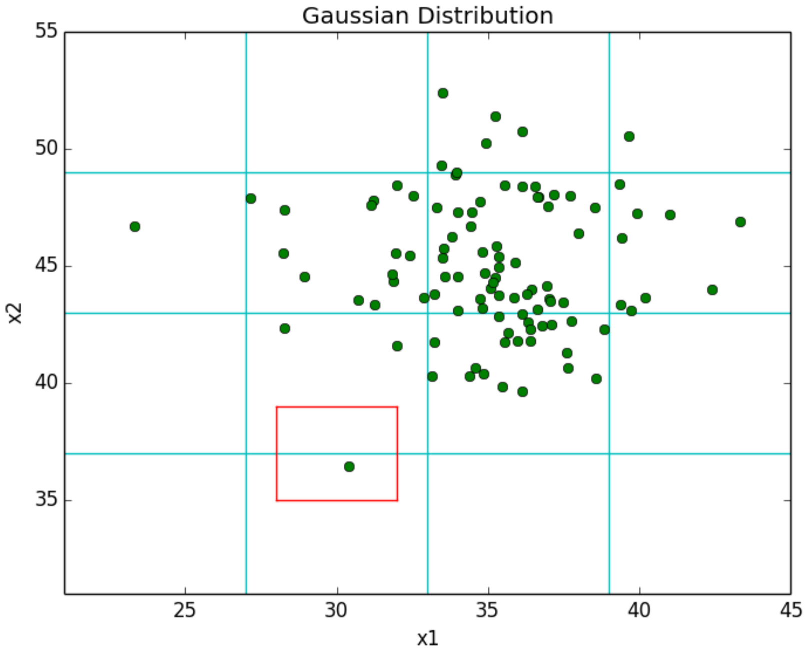


Figure 27: Progressive equi-width sampling in level 2 for user interest area 4

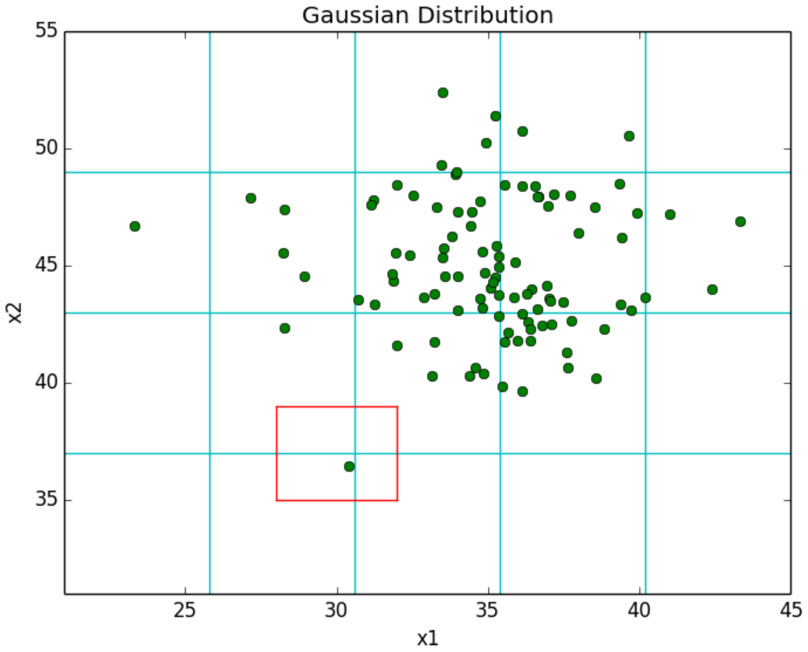


Figure 28: Single level equi-width sampling for user interest area 4

4 Experiment Evaluation

We carry out 3 experiments. First, we implement equi-width, equi-depth stratified sampling algorithm with SQL, evaluate their performance, and analyze the impact of three key factors to their performance. Second, we run simulations to test how good each sampling method is for finding the true samples inside the user interest area. The sampling methods we compare include random sampling, equi-depth stratified sampling and equi-width stratified sampling. Third, we run simulations to compare progressive equi-depth and equi-width stratified sampling with single level equi-depth and equi-width stratified sampling. The SQL query code for equi-depth and equi-width stratified sampling are in the Appendices.

System Information

For the first experiment, we test the performance of equi-depth and equi-width stratified sampling in a server machine. The server has two CPUs, each of which is Intel(R) Xeon(R) $3.00GHz$, $64bits$. The memory is $8GB$. The operating system is CentOS release 6.4, Linux version $2.6.32 - 358.23.2.el6.x86\_64$. The database we use is PostgreSQL 9.3.1.

For the second and third experiment, we run the simulation experiments to compare different sampling methods in a MacBook Pro. The CPU is $IntelCorei5(2.4GHz)$. The memory is $8GB$. The operating system is OS X EI Capitan. The simulation language is Python, and database we use is PostgreSQL.

Dataset

For the first experiment, we use the SDSS dataset [1] to test the performance of equi-depth and equi-width stratified sampling. We download data from the PhotoObj table of SkyServer DR8. The PhotoObj table contains 509 columns, which are all the attributes of each photometric(image) object. All the 509 columns include numeric values. The size of our base table is $78GB$.

For the second and third simulation experiments, we generate two synthetic datasets by ourselves. One is a 2 dimensional uniform distributed dataset and the other is a 2 dimensional Gaussian mixture dataset. Each of the two datasets includes 20000 tuples. The Gaussian mixture dataset contains 4 mixtures.

PostgreSQL buffer tuning

In the first experiment, when testing the performance of equi-depth and equi-width stratified sampling in our database, we set the buffer size of the database('work_mem' in PostgreSQL) large enough(for example 2GB), so that the sorting step in the query execution can be done completely in memory.

4.1 Stratified sampling performance

In the first experiment, we evaluate the performance of equi-width, equi-depth stratified sampling with respect to three key factors, which are the total number of tuples, the number of columns in the dataset and the number of dimensions we select to run sampling. We change the number of tuples by using different sampling databases, change the number of columns by using column tables, and change the number of dimensions by using different sampling space. We divide each dimension into 5 buckets. In the first two set of experiments, we perform sampling in 2 dimensional space, so we will get 25 buckets. We randomly select one sample from each bucket. We run each experiment 5 times, and report the average running time.

4.1.1 Vary the number of tuples

We change the number of tuples in the dataset by using sampling databases with different sampling ratios. The size of our base table is $78GB$. We create three sampling databases by random sampling tuples from the base table with three different sampling ratios, 10%, 20%, 30%, and record the size of the three result tables inside PostgreSQL. We run both equi-width and equi-depth stratified sampling in the 2 dimensional space(rowc and colc) over the three sampling databases, and record their running time using commands in PostgreSQL("explain analyze" tools). The dataset size, CPU time, I/O time and total time for equi-width sampling is in Table 1, and the result for equi-depth sampling is in Table 2. We also plot their running time in Figure 29.

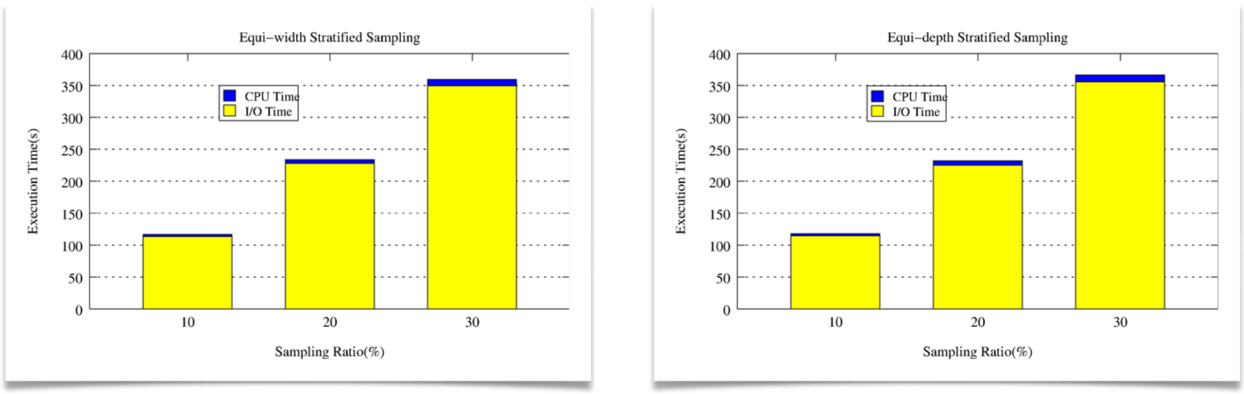
Dataset	Size(MB)	CPU Time(s)	I/O Time(s)	Total Time(s)
10% sampledb	7887	3.1428	113.5372	116.68
20% sampledb	15781	6.3434	227.7976	234.141
30% sampledb	23671	9.8084	349.7686	359.577

Table 1: Equi-width stratified sampling over sampling db

Dataset	Size(MB)	CPU Time(s)	I/O Time(s)	Total Time(s)
10% sampledb	7887	3.4994	114.589	118.0884
20% sampledb	15781	7.0886	225.355	232.4436
30% sampledb	23671	11.0612	355.5646	366.6258

Table 2: Equi-depth stratified sampling over sampling db

The number of tuples and table size of the three sampling database increases linearly. We can see from figure 29 below that both the CPU time and I/O time increases linearly with the size of the sampling database. This is true for both equi-width and equi-depth stratified sampling. We can also see that in this experiment, I/O time is much larger than CPU time. The database spends most of the time reading the table. The CPU time for equi-depth sampling is slightly larger than the CPU time for equi-width sampling because of the additional sorting steps to determine the group for each data point, but as the I/O time is dominant, the difference is not significant.



(a) Equi-width Sampling

(b) Equi-depth Sampling

Figure 29: Vary the number of tuples

4.1.2 Vary the number of table columns

In this experiment, we change the number of columns in the dataset by creating 5 column tables, which have 8 columns, 16 columns, 32 columns, 64 columns, 128 columns respectively. These columns have containment relationship, $8columns \in 16columns \in ... \in 128columns$. As these columns don't have the same data type or length, when we double the number of columns, the size of the column table is not necessarily doubled as shown in the Table 3 and Table 4. We run both equi-width and equi-depth stratified sampling over these column tables. The CPU time, I/O time and total running time for equi-width sampling are in Table 3, and the results for equi-depth sampling are in Table 4. We also plot the running time in Figure 30.

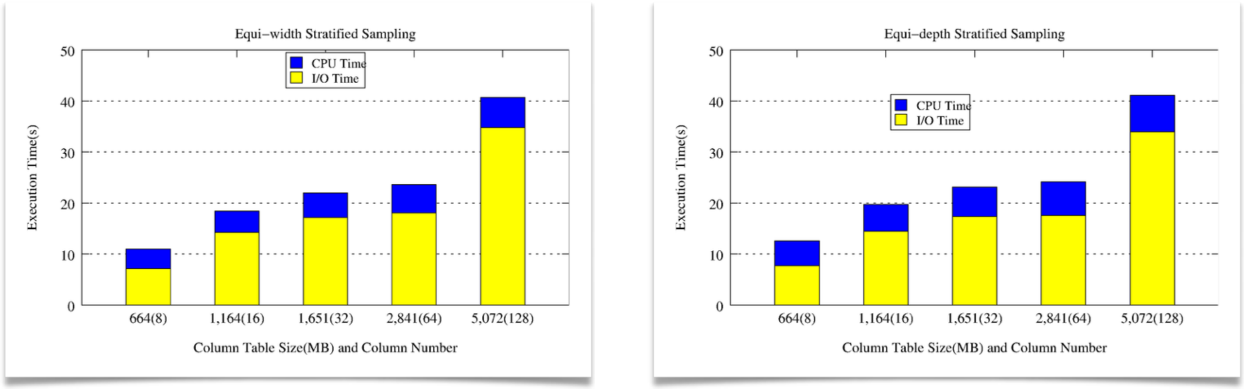
Dataset	Size(MB)	CPU Time(s)	I/O Time(s)	Total Time(s)
8 column	664	3.8498	7.1454	10.9952
16 column	1164	4.2046	14.251	18.4556
32 column	1651	4.8008	17.1732	21.974
64 column	2841	5.5914	18.0558	23.6472
128 column	5072	5.9502	34.7928	40.743

Table 3: Equi-width stratified sampling over column tables

Dataset	Size(MB)	CPU Time(s)	I/O Time(s)	Total Time(s)
8 column	664	4.8538	7.707	12.5608
16 column	1164	5.2324	14.496	19.7284
32 column	1651	5.8264	17.3508	23.1772
64 column	2841	6.5918	17.601	24.1928
128 column	5072	7.1568	34.0114	41.1682

Table 4: Equi-depth stratified sampling over column tables

We can see that in both equi-width and equi-depth sampling, when we use smaller column tables, the I/O time decreases, because the time to read column table is reduced. The CPU time also decreases, but the amount decreased is much smaller compared with the I/O time decrease. Comparing equi-width and equi-depth sampling, we can see that equi-depth sampling uses slightly more CPU time than equi-width sampling, because equi-depth sampling will perform one more sorting in each dimension than equi-width sampling to determine the group for each data point. The I/O time for equi-width and equi-depth sampling is similar.



(a) Equi-width Sampling

(b) Equi-depth Sampling

Figure 30: Vary the number of columns

4.1.3 Vary the number of sampling dimensions

In this experiment, we use the 10% sampling database with the full columns. We change the number of dimensions we select for sampling when running equi-width or equi-depth stratified sampling. We select 20 attributes from the SDSS dataset, which are *u, g, r, i, z, err_u, err_g, err_r, err_i, err_z, psfMag_u, psfMag_g, psfMag_r, psfMag_i, psfMag_z, psfMagErr_u, psfMagErr_g, psfMagErr_r, psfMagErr_i, psfMagErr_z*. For each experiment, we select the first 5, 10, 15, 20 dimensions to perform equi-width or equi-depth stratified sampling and compare their performance. The CPU time, I/O time and total running time for equi-width sampling are in Table 5, and the results for equi-depth sampling are in Table 6. We also plot the running time in Figure 31.

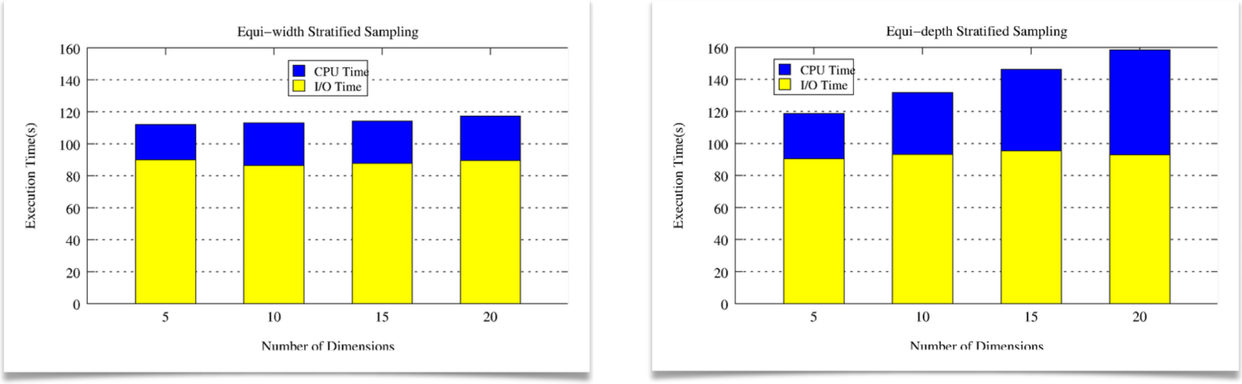
#dimensions	CPU Time(s)	I/O Time(s)	Total Time(s)
5	22.059	90.0035	112.0625
10	26.591	86.4745	113.0655
15	26.4115	87.77	114.1815
20	27.961	89.5045	117.4655

Table 5: Equi-width stratified sampling with different dimensions

#dimensions	CPU Time(s)	I/O Time(s)	Total Time(s)
5	28.202	90.491	118.693
10	38.7815	92.9915	131.773
15	50.848	95.31	146.158
20	65.5395	92.893	158.4325

Table 6: Equi-depth stratified sampling with different dimensions

We can see that in equi-width sampling, when we increase the number of sampling dimensions, the I/O time stays almost the same, because we read the same table each time. The CPU time slightly increases, but the increase is not significant compared with the total running time. For equi-depth sampling, when we increase the number of sampling dimensions, the I/O time stays almost the same, too. However, the CPU time increases significantly, because we need more sorting when we increase the number of dimensions. Comparing equi-width and equi-depth sampling, their I/O time cost is almost the same, but equi-depth spends more CPU time than equi-width, and as the number of dimensions increase, the difference becomes more significant.



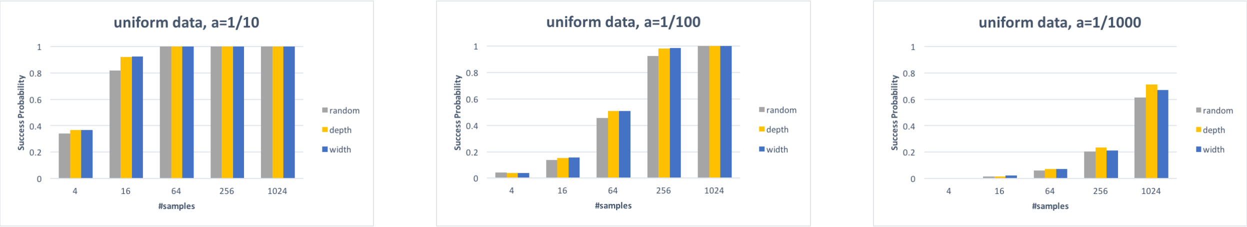
(a) Equi-width Sampling

(b) Equi-depth Sampling

Figure 31: Vary the number of sampling dimensions

4.2 Compare random sampling, equi-depth and equi-width sampling

We compare random sampling, equi-depth and equi-width stratified sampling by how good they are for finding the true samples inside the user interest area. We use both 2 dimensional uniformly distributed dataset and 2 dimensional Gaussian mixture dataset we generated by ourselves in our simulations. We select different user interest areas in different scenarios. We change the maximum number of samples permitted in each scenario. We run the simulation 1000 times for each scenario, and record the number of times T that the sampling method can get at least one user interested sample within the maximum number of samples, and then calculate the success probability as $\frac{T}{1000}$, and use it as our metric.

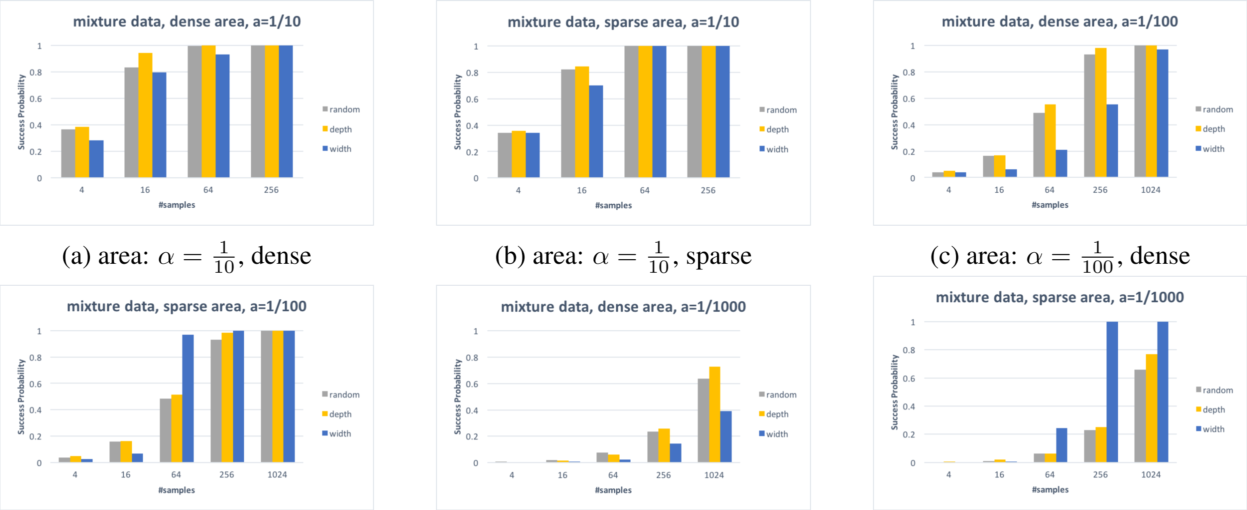


(a) area: $\alpha = \frac{1}{10}$

(b) area: $\alpha = \frac{1}{100}$

(c) area: $\alpha = \frac{1}{1000}$

Figure 32: uniform dataset



(a) area: $\alpha = \frac{1}{10}$, dense

(b) area: $\alpha = \frac{1}{10}$, sparse

(c) area: $\alpha = \frac{1}{100}$, dense

(d) area: $\alpha = \frac{1}{100}$, sparse

(e) area: $\alpha = \frac{1}{1000}$, dense

(f) area: $\alpha = \frac{1}{1000}$, sparse

Figure 33: mixture dataset

The simulation result for the uniformly distributed dataset is in Figure 32. We select three different user interest areas, where the data ratio α is $\frac{1}{10}$, $\frac{1}{100}$ and $\frac{1}{1000}$ respectively, and their results are in Figure 32a, 32b and 32c respectively.

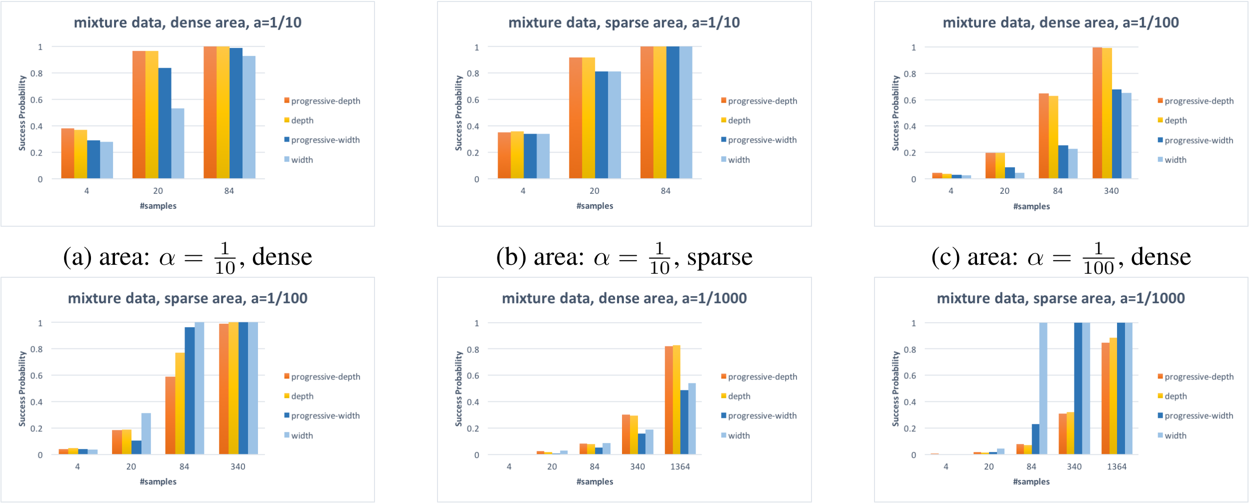
We can see that for uniformly distributed dataset, equi-depth stratified sampling and equi-width stratified sampling have similar success probability, which are slightly better than that of random sampling.

The simulation result for the Gaussian mixture dataset is in Figure 33. We select six different user interest areas, as displayed from Figure 33a to Figure 33f. These user interest areas have different data ratios, where α is equal to $\frac{1}{10}$, $\frac{1}{100}$ or $\frac{1}{1000}$. For each data ratio α, we select a dense user interest area and a sparse user interest area. For dense area, the data ratio α is greater than the area ratio β, which also means that the average density in the user interest area is greater than the average density of the whole data space. While for sparse area, the data ratio α is smaller than the area ratio β, which also means that the average density in the user interest area is smaller than the average density of the whole data space.

We can see that for Gaussian mixture dataset, the success probability for equi-depth stratified sampling is slightly better than that of random sampling. For dense user interest area, in most cases, equi-depth stratified sampling has higher success probability than equi-width stratified sampling, and equi-width stratified sampling is even worse than random sampling. For sparse user interest area, in most cases, equi-width stratified sampling is better than equi-depth stratified sampling and random sampling.

4.3 Compare progressive sampling and single-level stratified sampling

We also compare progressive equi-depth and equi-width sampling with single level equi-depth and equi-width sampling by how good they are for finding the true samples inside the user interest area. We use the same 2 dimensional Gaussian mixture dataset as in section 4.2. We select different user interest areas in different scenarios. We change the maximum number of samples permitted in each scenario. We run the simulation for each scenario by 1000 times, and record the number of times T that the sampling method can get at least one user interested sample within the maximum number of samples, and then calculate the success probability as $\frac{T}{1000}$, and use it as our metric.



(a) area: $\alpha = \frac{1}{10}$, dense (b) area: $\alpha = \frac{1}{10}$, sparse (c) area: $\alpha = \frac{1}{100}$, dense

(d) area: $\alpha = \frac{1}{100}$, sparse (e) area: $\alpha = \frac{1}{1000}$, dense (f) area: $\alpha = \frac{1}{1000}$, sparse

Figure 34: progressive sampling VS single-level sampling

The simulation result is in Figure 34. The six different user interest areas from Figure 34a to Figure 34f are the same six user interest areas as in Figure 33. Their data ratio α is equal to $\frac{1}{10}$, $\frac{1}{100}$ or $\frac{1}{1000}$ respectively. For each data ratio α, we have a dense user interest area and a sparse user interest area.

Comparing progressive equi-depth sampling with single-level equi-depth sampling, we can see that for dense region, progressive equi-depth sampling is slightly better than single-level equi-depth sampling, while for sparse region, single-level equi-depth sampling is slightly better than progressive equi-depth sampling. However, the difference between the two sampling methods is not significant.

Comparing progressive equi-width sampling with single-level equi-width sampling, which one is better doesn't depend on the density of the user interest area, but depends on the location of the user interest area. For some locations, progressive equi-width sampling has higher success probability, while for some other locations, single-level equi-width sampling has higher success probability.

5 Related Work

Olken studied how to obtain samples from query results without first performing the query [23]. In his problem, the query to perform is known. However, in our problem, we don't know the target query, which makes the two problems different. There are also many previous works on the problem of approximate query processing(APQ) [24] [25] [26] [27]. However, they try to apply sampling methods to generate approximate early results(i.e. aggregation) for known queries, and their purpose is for fast approximate query processing, which is different from our problem. [2] applies sampling method to generate approximate visualization. SearchLight [3] studies the exploration problem, but they use constraint programming solvers, which is different from our sampling methods. [4] focus on specifically object-centric exploration queries and their visualization, but our exploration problem is more generic. RINSE [5] studies exploration of data series data, which are different from our scenarios. Smart Drill-Down [6] build an efficient drill-down operation, providing summary of groups of tuples, however, the summary of tuples are different from our problems. [7] applies stratified sampling method to aggregation queries, which is different from our problem. DICE [8] is an exploration system for data cubes. [9] focus on the accuracy of sampling-based aggregation query estimation. [10] studies the problem of counting and sampling triangles in a massive graph, whose edges arrive as a stream. Our problem doesn't have the graph structure. [11] studies spatial data set, and want to find regions that include relevant points of interest based on relevant keywords. VSOutlier [12] is a system supporting efficient outlier detection in big data streams. SPIRE [13] is for efficient interactive rule-mining. [14] applies stratified sampling for online social network, and the implementation is based on MapReduce framework, which are different from our approaches. [15] designs an adaptive indexing approach for fast data series exploration, which are different from our problem. The semantic window paper [16] also studies the data exploration problem, but they perform exploration based on windows, on the other hand, we perform exploration based on individual samples. We study the same interactive data exploration problem as the Explore-by-example paper [17], but we extend the initial sampling algorithm, and compare different sampling methods in the initial sampling phase. [18] mainly studies Gibbs sampling, a Bayesian statistical method, and implements a high-throughput Gibbs sampling method for factor graphs that are larger than main memory, which are different from the problem we study. [19] tries to explore collaborative ratings based on aggregate queries, which are different from our problem. [20] studies structure-aware sampling methods to get summary for range-sum queries, which is essentially aggregation queries. [21] addresses mining the search engine's corpus using sampling, which are different from our problems. [22] studies efficient algorithms to compute approximate quantiles in large-scale sensor networks, which are different from our problems.

6 Conclusion

In this project, we designed and implemented several sampling algorithms, equi-depth stratified sampling, equi-width stratified sampling and progressive stratified sampling. We derive the probability lower bound for these sampling method, and compare their lower bound with each other in different scenarios. We also run several sets of experiments to demonstrate our theoretical analysis, and get consistent results.

References

[1] `http://skyserver.sdss.org/dr8/en/help/browser/browser.asp`

[2] Kim, Albert, Eric Blais, Aditya Parameswaran, Piotr Indyk, Sam Madden, and Ronitt Rubinfeld. "Rapid sampling for visualizations with ordering guarantees." Proceedings of the VLDB Endowment 8, no. 5 (2015): 521-532.

[3] Kalinin, Alexander, Ugur Cetintemel, and Stan Zdonik. "Searchlight: enabling integrated search and exploration over large multidimensional data." Proceedings of the VLDB Endowment 8, no. 10 (2015): 1094-1105.

[4] Wu, You, Boulos Harb, Jun Yang, and Cong Yu. "Efficient evaluation of object-centric exploration queries for visualization." Proceedings of the VLDB Endowment 8, no. 12 (2015): 1752-1763.

[5] Zoumpatianos, Kostas, Stratos Idreos, and Themis Palpanas. "RINSE: interactive data series exploration with ADS+." Proceedings of the VLDB Endowment 8, no. 12 (2015): 1912-1915.

[6] Joglekar, Manas, Hector Garcia-Molina, and Aditya Parameswaran. "Smart Drill-Down: A New Data Exploration Operator." Proceedings of the VLDB Endowment 8, no. 12 (2015): 1928-1931.

[7] Yan, Ying, Liang Jeff Chen, and Zheng Zhang. "Error-bounded sampling for analytics on big sparse data." Proceedings of the VLDB Endowment 7, no. 13 (2014): 1508-1519.

[8] Jayachandran, Prasanth, Karthik Tunga, Niranjan Kamat, and Arnab Nandi. "Combining user interaction, speculative query execution and sampling in the DICE system." Proceedings of the VLDB Endowment 7, no. 13 (2014): 1697-1700.

[9] Nirkhiwale, Supriya, Alin Dobra, and Christopher Jermaine. "A sampling algebra for aggregate estimation." Proceedings of the VLDB Endowment 6, no. 14 (2013): 1798-1809.

[10] Pavan, Aduri, Kanat Tangwongsan, Srikanta Tirthapura, and Kun-Lung Wu. "Counting and sampling triangles from a graph stream." Proceedings of the VLDB Endowment 6, no. 14 (2013): 1870-1881.

[11] Cao, Xin, Gao Cong, Christian S. Jensen, and Man Lung Yiu. "Retrieving regions of interest for user exploration." Proceedings of the VLDB Endowment 7, no. 9 (2014): 733-744.

[12] Cao, Lei, Qingyang Wang, and Elke A. Rundensteiner. "Interactive outlier exploration in big data streams." Proceedings of the VLDB Endowment 7, no. 13 (2014): 1621-1624.

[13] Lin, Xika, Abhishek Mukherji, Elke A. Rundensteiner, and Matthew O. Ward. "SPIRE: supporting parameter-driven interactive rule mining and exploration." Proceedings of the VLDB Endowment 7, no. 13 (2014): 1653-1656.

[14] Levin, Roy, and Yaron Kanza. "Stratified-sampling over social networks using mapreduce." In Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 863-874. ACM, 2014.

[15] Zoumpatianos, Kostas, Stratos Idreos, and Themis Palpanas. "Indexing for interactive exploration of big data series." In Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 1555-1566. ACM, 2014.

[16] Kalinin, Alexander, Ugur Cetintemel, and Stan Zdonik. "Interactive data exploration using semantic windows." In Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 505-516. ACM, 2014.

[17] Dimitriadou, Kyriaki, Olga Papaemmanouil, and Yanlei Diao. "Explore-by-example: An automatic query steering framework for interactive data exploration." In Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 517-528. ACM, 2014.

[18] Zhang, Ce, and Christopher Ré. "Towards high-throughput Gibbs sampling at scale: A study across storage managers." In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp. 397-408. ACM, 2013.

[19] Thirumuruganathan, Saravanan, Mahashweta Das, Shrikant Desai, Sihem Amer-Yahia, Gautam Das, and Cong Yu. "MapRat: meaningful explanation, interactive exploration and geo-visualization of collaborative ratings." Proceedings of the VLDB Endowment 5, no. 12 (2012): 1986-1989.

[20] Cohen, Edith, Graham Cormode, and Nick Duffield. "Structure-aware sampling: Flexible and accurate summarization." arXiv preprint arXiv:1102.5146 (2011).

[21] Zhang, Mingyang, Nan Zhang, and Gautam Das. "Mining a search engine's corpus: efficient yet unbiased sampling and aggregate estimation." In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, pp. 793-804. ACM, 2011.

[22] Huang, Zengfeng, Lu Wang, Ke Yi, and Yunhao Liu. "Sampling based algorithms for quantile computation in sensor networks." In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, pp. 745-756. ACM, 2011.

[23] Olken, Frank, and Doron Rotem. "Simple Random Sampling from Relational Databases." VLDB. Vol. 86. 1986.

[24] Hellerstein, Joseph M., Peter J. Haas, and Helen J. Wang. "Online aggregation." ACM SIGMOD Record. Vol. 26. No. 2. ACM, 1997.

[25] Acharya, Swarup, Phillip B. Gibbons, and Viswanath Poosala. "Aqua: A fast decision support systems using approximate query answers." Proceedings of the 25th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., 1999.

[26] Chaudhuri, Surajit, Gautam Das, and Vivek Narasayya. "Optimized stratified sampling for approximate query processing." ACM Transactions on Database Systems (TODS) 32.2 (2007): 9.

[27] Agarwal, Sameer, et al. "BlinkDB: queries with bounded errors and bounded response times on very large data." Proceedings of the 8th ACM European Conference on Computer Systems. ACM, 2013.

Appendices

A SQL query for equi-depth stratified sampling

```
select id, x1, x2
from (
        select id, x1, x2, grp_1, grp_2,
        row_number() over (partition by grp_1, grp_2 order by random()) as rn
        from (
                select id, x1, x2, grp_1,
                ntile(2) over (partition by grp_1 order by x2) as grp_2
                from (select id, x1, x2,
                        ntile(2) over (order by x1) as grp_1
                        from data_table
                        where x1 >= 200 and x1 < 300
                             and x2 >= 110 and x2 < 200
                ) as sub1
        ) as sub2
) as sub3
where rn <= 1;
```

B SQL query for equi-width stratified sampling

```
select id, x1, x2
from (
        select id, x1, x2, grp_1, grp_2,
                row_number() over(
                                partition by grp_1, grp_2
                                order by random()
                                ) as rn
        from (
                select id, x1, x2,
                        width_bucket(x1, 200, 300, 2) as grp_1,
                        width_bucket(x2, 110, 200, 2) as grp_2
                from data_table
                where x1 >= 200 and x1 < 300
                        and x2 >= 110 and x2 < 200
        ) as sub1
) as sub2
where rn <= 1;
```