Estimating Sum by Weighted Sampling

Rajeev Motwani[1], Rina Panigrahy[2], and Ying Xu[1⋆]

[1] Dept of Computer Science, Stanford University, USA
[2] Microsoft Research, Mountain View, CA, USA
{rajeev,xuying}@cs.stanford.edu, rina@microsoft.com

Abstract. We study the classic problem of estimating the sum of n variables. The traditional uniform sampling approach requires a linear number of samples to provide any non-trivial guarantees on the estimated sum. In this paper we consider various sampling methods besides uniform sampling, in particular sampling a variable with probability proportional to its value, referred to as *linear weighted sampling*. If only linear weighted sampling is allowed, we show an algorithm for estimating sum with $\tilde{O}(\sqrt{n})$ samples, and it is almost optimal in the sense that $\Omega(\sqrt{n})$ samples are necessary for any reasonable sum estimator. If both uniform sampling and linear weighted sampling are allowed, we show a sum estimator with $\tilde{O}(\sqrt[3]{n})$ samples. More generally, we may allow general weighted sampling where the probability of sampling a variable is proportional to any function of its value. We prove a lower bound of $\Omega(\sqrt[3]{n})$ samples for any reasonable sum estimator using general weighted sampling, which implies that our algorithm combining uniform and linear weighted sampling is an almost optimal sum estimator.

1 Introduction

We consider the classic problem of estimating the sum (or equivalently, the average) of n non-negative variables. This problem has numerous important applications in various areas of computer science, statistics and engineering. Measuring the exact value of each variable incurs some cost, so people want to get a reasonable estimator of the sum while measure as few variables as possible.

In the traditional setting, only uniform sampling is used, i.e. each time we can sample one variable uniformly at random and ask its value. Under this setting it is easy to see that any reasonable estimator requires a linear sample size if the underlying distribution is arbitrary. Consider the following two instances of inputs: in the first input all variables are 0, while in the second input all are 0 except one variable x_1 is a large number. Any sampling scheme cannot distinguish the two inputs until it sees x_1, and with uniform sampling it takes linear

⋆ Rajeev Motwani is supported in part by NSF Grants EIA-0137761 and ITR-0331640, and grants from Media-X and SNRC. Rina Panigrahy's work was done when he was at Stanford, and he was supported by Stanford Graduate Fellowship. Ying Xu is supported in part by a Stanford Graduate Fellowship and NSF Grants EIA-0137761 and ITR-0331640.

samples to hit x_1. We defer the formal definition of "reasonable estimator" to Section 2, but intuitively we cannot get a good estimator if we cannot distinguish the two inputs.

In this paper, we study the problem of estimating sum using other sampling methods besides uniform sampling. For example, suppose we now allow sampling a variable with probability proportional to its value, which we refer to as *linear weighted sampling*; in Section 1.1 we will discuss applications where such sampling is feasible. Using linear weighted sampling one sample is sufficient to distinguish the above two inputs, and it seems plausible that generally we can get good sum estimators with less samples using such sampling method. In this paper we show an algorithm for sum estimation with $\tilde{O}(\sqrt{n})$ samples using only linear weighted sampling, and it is almost optimal in the sense that $\Omega(\sqrt{n})$ samples are necessary for any reasonable estimator using only linear weighted sampling. Our algorithm assumes no prior knowledge about the input distribution.

Next, if we use both uniform sampling and linear weighted sampling, we can further reduce the number of samples needed. We present a sum estimator with $\tilde{O}(\sqrt[3]{n})$ samples using a combination of the two sampling methods, and prove a lower bound of $\Omega(\sqrt[3]{n})$ samples.

More generally, we may allow sampling where the probability of sampling a variable can be proportional to any function of its value (the function does not depend on n), referred as to *(general) weighted sampling*. While we are not sure whether general sampling is feasible in real applications, we show a negative result that such extra power does not provide a better estimator: we prove a lower bound of $\Omega(\sqrt[3]{n})$ samples for any reasonable sum estimator, using any combination of general weighted sampling methods. This implies that combining uniform and linear weighted sampling gives an almost optimal sum estimator (up to a poly-log factor), hence there is no need to pursue fancier sampling methods in this family for the purpose of estimating sum.

1.1 Applications

The problem of estimating sum is a classic problem with wide applications in various areas, and linear weighted sampling is a natural sampling method feasible in many applications. In particular, if we want to estimate the total number of some objects in a system and those objects fall into disjoint classes, then the problem becomes estimating the sum of variables with each variable indicating the number of objects in one class; if uniform sampling of the objects is possible, then linear weighted sampling can be implemented by sampling an object uniformly at random and returning the class of the sampled object.

One such application is estimating search engine index sizes or the web size, which has aroused interests in both academic and industrial world in recent years (see for example [12, 13, 10, 6, 4]). One method used in those papers is to partition the search index (web) into domains (web servers), and estimate the sum of those domain (server) sizes. It is relatively easy to get the total domain (web server) number n (either by uniformly sampling IP space or people publish this number periodically). For example in 1999 Lawrence and Giles estimated the

number of web servers to be 2.8 million by randomly testing IP addresses; then they exhaustively crawled 2500 web servers and found that the mean number of pages per server was 289, leading to an estimate of the web size of 800 million [13]. Lawrence and Giles essentially used uniform sampling to estimate the sum, however, the domain size distribution is known to be highly skewed and uniform sampling has high variance for such inputs. We can also do linear weighted sampling: uniformly sample a page from the web or a search engine index (the technique of uniform sampling a page from the web/index has been studied in for example [11, 3]) and take the domain of the page, then the probability of sampling a domain is proportional to its size. Then we can apply the techniques in this paper, which shall provide a more accurate estimate than using only uniform sampling.

1.2 Related Work

Estimating the sum of n variables is a classical statistical problem. For the case where all the variables are between $[0, 1]$, an additive approximation of the mean can be easily computed by taking a random sample of size $O(\frac{1}{\epsilon^2} \lg \frac{1}{\delta})$ and computing the mean of samples; [7] proves a tight lower bound on the sample size. However, uniform sampling works poorly on heavily tailed inputs when the variables are from a large range, and little is known beyond uniform sampling.

Weighted sampling is also known as "importance sampling". General methods of estimating a quantity using importance sampling have been studied in statistics (see for example [14]), but the methods are either not applicable here or less optimal. To estimate a quantity $h_\pi = \sum \pi(i)h(i)$, importance sampling generates independent samples $i_1, i_2, \ldots, i_N$ from a distribution p. One estimator for h_π is $\hat{\mu} = \frac{1}{N} \sum h(i_k)\pi(i_k)/p(i_k)$. For the sake of estimating sum, $\pi(i) = 1$ and $h(i)$ is the value of ith variable x_i. In linear weighted sampling, $p(i) = x_i/S$, where S is exactly the sum we are trying to estimate, therefore we are not able to compute this estimator $\hat{\mu}$ for sum. Another estimator is

$$\tilde{\mu} = \frac{\sum h(i_k)\pi(i_k)/\tilde{p}(i_k)}{\sum \pi(i_k)/\tilde{p}(i_k)},$$

where $\tilde{p}$ is identical to p up to normalization and thus computable. However, the variance of $\tilde{\mu}$ is even larger than the variance using uniform sampling.

A related topic is priority sampling and threshold sampling for estimating subset sums proposed and analyzed in [9, 1, 16]. But their cost model and application are quite different: they aim at building a sketch so that the sum of any subset can be computed (approximately) by only looking at the sketch; in particular their cost is defined as the size of the sketch and they can read all variables for free, so computing the total sum is trivial in their setting.

There is extensive work in estimating other frequency moments $F_k = \sum x_i^k$ (sum is the first moment F_1), in the random sampling model as well as in the streaming model (see for example [2, 8, 5]). The connection between the two models is discussed in [5]. Note that their sampling primitive is different from ours, and they assume F_1 is known.

2 Definitions and Summary of Results

Let $x_1, x_2, \ldots, x_n$ be n variables. We consider the problem of estimating the sum $S = \sum_i x_i$, given n. We also refer to variables as *buckets* and the value of a variable as its *bucket size*.

In *(general) weighted sampling* we can sample a bucket x_i with probability proportional to a function of its size $f(x_i)$, where f is an arbitrary function of x_i (f independent on n). Two special cases are *uniform sampling* where each bucket is sampled uniformly at random ($f(x) = 1$), and *linear weighted sampling* where the probability of sampling a bucket is proportional to its size ($f(x) = x$). We assume sampling with replacement.

We say an algorithm is an (ϵ, δ)-*estimator* ($0 < \epsilon, \delta < 1$), if it outputs an estimated sum S' such that with probability at least $1 - \delta$, $|S' - S| \leq \epsilon S$. The algorithm can take random samples of the buckets using some sampling method and learn the sizes as well as the labels of the sampled buckets. We measure the complexity of the algorithm by the total number of samples it takes. The algorithm has no prior knowledge of the bucket size distribution.

The power of the sum estimator is constrained by the sampling methods it is allowed to use. This paper studies the upper and lower bounds of the complexities of (ϵ, δ)-estimators under various sampling methods. As pointed out in Section 1, using only uniform sampling there is no (ϵ, δ)-estimator with sub-linear samples.

First we show an (ϵ, δ)-estimator using linear weighted sampling with $\tilde{O}(\sqrt{n})$ samples. While linear weighted sampling is a natural sampling method, to derive the sum from such samples does not seem straightforward. Our scheme first converts the general problem to a special case where all buckets are either empty or of a fixed size; now the problem becomes estimating the number of non-empty buckets and we make use of birthday paradox by examining how many samples are needed to find a repeat. Each step involves some non-trivial construction and the detailed proof is presented in Section 3.

In Section 4 we consider sum estimators where both uniform and linear weighted sampling are allowed. Section 4.1 proposes an algorithm with $\tilde{O}(\sqrt[3]{n})$ samples which builds upon the linear weighted sampling algorithm in Section 3. Section 4.2 gives a different algorithm with $\tilde{O}(\sqrt{n})$ samples: although it is asymptotically worse than the former algorithm in terms of n, it has better dependency on ϵ and a much smaller hidden constant; also this algorithm is much neater and easier to implement.

Finally we present lower bounds in Section 5. We prove that the algorithms in Section 3 and 4.1 are almost optimal in terms of n up to a poly-log factor. More formally, we prove a lower bound of $\Omega(\sqrt{n})$ samples using only linear weighted sampling (more generally, using any combination of general weighted sampling methods with the constraint $f(0) = 0$); a lower bound of $\Omega(\sqrt[3]{n})$ samples using any combination of general weighted sampling methods.

All algorithms and bounds can be extended to the case where the number of buckets n is only approximately known (with relative error less than ϵ). We omit the details for lack of space.

3 An $\tilde{O}(\sqrt{n})$ Estimator using Linear Weighted Sampling

Linear weighted sampling is a natural sampling method, but to efficiently derive the sum from such samples does not seem straightforward. Our algorithm first converts the general problem to a special case where all buckets are either empty or of a fixed size, and then tackle the special case making use of the *birthday paradox*, which states that given a group of $\sqrt{365}$ randomly chosen people, there is a good chance that at least two of them have the same birthday.

Let us first consider the special case where all non-zero buckets are of equal sizes. Now linear weighted sampling is equivalent to uniform sampling among non-empty buckets, and our goal becomes estimating the number of non-empty buckets, denoted by B ($B \leq n$). We focus on a quantity we call "*birthday period*", which is the number of buckets sampled until we see a repeated bucket. We denote by $r(B)$ the birthday period of B buckets; its expected value $E[r(B)]$ is $\Theta(\sqrt{B})$ according to the birthday paradox. We will estimate the expected birthday period using linear weighted sampling, and then use it to infer the value of B. Most runs of birthday period take $O(\sqrt{B}) = O(\sqrt{n})$ samples, and we can cut off runs which take too long; $\lg \frac{1}{\delta}$ runs are needed to boost confidence, thus in total we need $O(\sqrt{n})$ samples to estimate B.

Now back to the general problem. We first guess the sum is an and fix a uniform bucket size ϵa. For each bucket in the original problem, we round its size down to $k\epsilon a$ (k being an integer) and break it into k buckets. If our guess of sum is (approximately) right, then the number of new buckets B is approximately n/ϵ; otherwise B is either too small or too large. We can estimate B by examining the birthday period as above using $O(\sqrt{n/\epsilon})$ samples, and check whether our guess is correct. Finally, since we allow a multiplicative error of ϵ, a logarithmic number of guesses suffice.

Before present the algorithm, we first establish some basic properties of birthday period $r(B)$. The following lemma bounds the expectation and variance of $r(B)$; property (3) shows that birthday period is "gap preserving" so that if the number of buckets is off by an ϵ factor, we will notice a difference of $c\epsilon$ in the birthday period. We can write out the exact formula for $E[r(B)]$ and $var(r(B))$, and the rest of the proof is merely algebraic manipulation. The detailed proof can be found in the Appendix.

Lemma 1. *(1) $E[r(B)]$ monotonically increases with B;*
(2) $E[r(B)] = \Theta(\sqrt{B})$;
(3) $E[r((1+\epsilon)B)] > (1+c\epsilon)E[r(B)]$, where c is a constant.
(4) $var(r(B)) = O(B)$;

Lemma 2 tackles the special case, stating that with $\sqrt{b}$ samples we can tell whether the total number of buckets is at most b or at least $b(1+\epsilon)$. The idea is to measure the birthday period and compare with the expected period in the two cases. We use the standard "median of the mean" trick: first get a constant correct probability using Chebyshev inequality, then boost the probability using Chernoff bound. See details in the algorithm *BucketNumber*. Here c is the constant in Lemma 1(3); c_1 and c_2 are constants.

BucketNumber(b, ϵ, δ)

1. Compute $r = E[r(b)]$;
2. for $i = 1$ to $k_1 = c_1 \lg \frac{1}{\delta}$
3. for $j = 1$ to $k_2 = c_2/\epsilon^2$
4. sample until see a repeated bucket; let r_j be the number of samples
5. if $\sum_{j=1}^{k_2} r_j/k_2 \leq (1 + c\epsilon/2)r$ then $s_i = true$, else $s_i = false$
6. if more than half of s_i are $true$ then output "$\leq b$ buckets"
   else output "$\geq b(1+\epsilon)$ buckets"

Lemma 2. *If each sample returns one of B buckets uniformly at random, then the algorithm BucketNumber tells whether $B \leq b$ or $B \geq b(1+\epsilon)$ correctly with probability at least $1 - \delta$; it uses $\Theta(\sqrt{b} \lg \frac{1}{\delta}/\epsilon^2)$ samples.*

Proof. We say the algorithm does k_1 runs, each run consisting of k_2 iterations. We first analyze the complexity of the algorithm. We need one small trick to avoid long runs: notice that we can cut off a run and set $s_i = false$ if we have already taken $(1 + c\epsilon/2)rk_2$ samples in this run. Therefore the total number of samples is at most

$$(1 + c\epsilon/2)rk_2k_1 = (1 + c\epsilon/2)E[r(b)]\frac{c_2}{\epsilon^2}c_1 \lg \frac{1}{\delta} = \Theta(\frac{\sqrt{b} \lg \frac{1}{\delta}}{\epsilon^2}).$$

The last equation uses Property (2) of Lemma 1.

Below we prove the correctness of the algorithm. Consider one of the k_1 runs. Let r' be the average of the k_2 measured birthday periods r_j. Because each measured period has mean $E[r(B)]$ and variance $var(r(B))$, we have $E[r'] = E[r(B)]$ and $var(r') = var(r(B))/k_2$.

If $B \leq b$, then $E[r'] = E[r(B)] \leq r$. By Chebyshev inequality [15],

$$Pr[r' > (1+\frac{c\epsilon}{2})r] \leq Pr[r' > E[r(B)]+\frac{rc\epsilon}{2}] \leq \frac{var(r(B))/k_2}{(rc\epsilon/2)^2} \leq \frac{O(b)\epsilon^2/c_2}{(\Theta(\sqrt{b})c\epsilon/2)^2} = \frac{O(1)}{c_2}$$

If $B \geq b(1+\epsilon)$, then $E[r'] \geq E[r(b(1+\epsilon))] \geq (1+c\epsilon)r$ by Lemma 1.

$$Pr[r' < (1 + \frac{c\epsilon}{2})r] \leq Pr[r' < (1 - \frac{c\epsilon}{4})E[r']] \leq \frac{var(r(B))/k_2}{(E[r(B)]c\epsilon/4)^2} = \frac{O(1)}{c_2}$$

We choose the constant c_2 large enough such that both probabilities are no more than 1/3. Now when $B \leq b$, since $Pr[r' > (1+c\epsilon/2)r] \leq 1/3$, each run sets $s_i = false$ with probability at most 1/3. Our algorithm makes wrong judgement only if more than half of the k_1 runs set $s_i = false$, and by Chernoff bound [15], this probability is at most $e^{-c'k_1}$. Choose appropriate c_1 so that the error probability is at most δ. Similarly, when $B \geq (1+\epsilon)b$, each run sets $s_i = true$ with probability at most 1/3, and the error probability of the algorithm is at most δ.□

Algorithm *LWSE* (stands for *Linear Weighted Sampling Estimator*) shows how to estimate sum for the general case. The labelling in step 3 is equivalent to the following process: for each original bucket, round its size down to a multiple

of $\epsilon_1 a$ and split into several "standard" buckets each of size $\epsilon_1 a$; each time sampling returns a standard bucket uniformly at random. The two processes are equivalent because they have the same number of distinct labels (standard buckets) and each sampling returns a label uniformly at random. Therefore by calling $BucketNumber(n(1+\epsilon_1)/\epsilon_1, \epsilon_1, \delta_1)$ with such samples, we can check whether the number of standard buckets $B \le n(1+\epsilon_1)/\epsilon_1$ or $B \ge n(1+\epsilon_1)^2/\epsilon_1$, allowing an error probability of δ_1.

LWSE(n, ϵ, δ)

1. get a lower bound L of the sum: sample one bucket using linear weighted sampling and let L be the size of the sampled bucket;
2. for $a = L/n, L(1+\epsilon_1)/n, \ldots, L(1+\epsilon_1)^k/n, \ldots$ (let $\epsilon_1 = \epsilon/3$)
3. for each sample returned by linear weighted sampling, create a label as follows: suppose a bucket x_i of size $s = m\epsilon_1 a + r$ is sampled (m is an integer and $r < \epsilon_1 a$); discard the sample with probability r/s; with the remaining probability generate a number l from $1..m$ uniformly at random and label the sample as i_l;
4. call $BucketNumber(n(1+\epsilon_1)/\epsilon_1, \epsilon_1, \delta_1)$, using the above samples in step 4 of $BucketNumber$. If $BucketNumber$ outputs "$\le n(1+\epsilon_1)/\epsilon_1$", then output $S' = an$ as the estimated sum and terminate.

Theorem 1. *LWSE is an (ϵ, δ)-estimator with $O(\sqrt{n}(\frac{1}{\epsilon})^{\frac{7}{2}} \log n(\log \frac{1}{\delta} + \log \frac{1}{\epsilon} + \log\log n))$ samples, where n is the number of buckets.*

Proof. We first show that the algorithm terminates with probability at least $1-\delta_1$. S must fall in $[a_0 n, a_0 n(1+\epsilon_1)]$ for some a_0, and we claim that the algorithm will terminate at this a_0, if not before: since $S \le a_0 n(1+\epsilon_1)$, the sum after rounding down is at most $a_0 n(1+\epsilon_1)$ and hence the number of standard buckets $B \le n(1+\epsilon_1)/\epsilon_1$; by Lemma 2 it will pass the check with probability at least $1-\delta_1$ and terminate the algorithm.

Next we show that given that $LWSE$ terminates by a_0, the estimated sum is within $(1 \pm \epsilon)S$ with probability $1-\delta_1$. Since the algorithm has terminated by a_0, the estimated sum cannot be larger than S, so the only error case is $S' = an < (1-\epsilon)S$. The sum loses at most $na\epsilon_1$ after rounding down, so

$$B \ge \frac{S - an\epsilon_1}{a\epsilon_1} \ge \frac{\frac{an}{1-\epsilon} - an\epsilon_1}{a\epsilon_1} = \frac{n}{(1-\epsilon)\epsilon_1} - n \ge n\frac{1-\epsilon_1}{(1-\epsilon)\epsilon_1} \ge n\frac{(1+\epsilon_1)^2}{\epsilon_1}$$

The probability that it can pass the check for a fixed $a < a_0$ is at most δ_1; by union bound, the probability that it passes the check for any $a < a_0$ is at most $\delta_1 \log_{1+\epsilon} \frac{S}{L}$. Combining the two errors, the total error probability is at most $\delta_1(\log_{1+\epsilon} \frac{S}{L} + 1)$. Choose $\delta_1 = \delta/(\log_{1+\epsilon} \frac{S}{L} + 1)$, then with probability at least $1-\delta$ the estimator outputs an estimated sum within $(1 \pm \epsilon)S$.

Now we analyze the complexity of $LWSE$. Ignore the discarded samples for now and count the number of valid samples. By Lemma 2, for each a we need

$$N_1 = O(\frac{\log \frac{1}{\delta_1} * \sqrt{\frac{n(1+\epsilon_1)}{\epsilon_1}}}{\epsilon_1^2}) = O(\sqrt{n}(\frac{1}{\epsilon})^{\frac{5}{2}}(\log \frac{1}{\delta} + \log \frac{1}{\epsilon} + \log\log \frac{S}{L}))$$

samples, and there are $\log_{1+\epsilon} \frac{S}{L} = O(\log \frac{S}{L}/\epsilon)$ as. As for the discarded samples, the total discarded size is at most $an\epsilon_1$, and we always have $S \geq an$ if the algorithm is running correctly, therefore the expected probability of discarded samples is at most $\epsilon_1 = \epsilon/3 \leq 1/3$. By Chernoff bound, with high probability the observed probability of discarded samples is at most half, i.e. the discarded samples at most add a constant factor to the total sample number.

Finally, the complexity of the estimator has the term $\log \frac{S}{L}$. Had we simply started guessing from $L = 1$, the cost would depend on $\log S$. The algorithm chooses L to be the size of a sampled bucket using linear weighted sampling. We claim that with high probability $L \geq S/n^2$: otherwise $L < S/n^2$, then the probability that linear weighted sampling returns any bucket of size no more than L is at most $n * L/S < 1/n$.

Summing up, the total sample number used in $LWSE$ is

$$N_1 * O(\frac{\log n^2}{\epsilon}) = O(\sqrt{n}(\frac{1}{\epsilon})^{\frac{7}{2}} \log n(\log \frac{1}{\delta} + \log \frac{1}{\epsilon} + \log \log n)).\square$$

4 Combining Uniform and Linear Weighted Sampling

In this section we design sum estimators using both uniform sampling and linear weighted sampling. We present two algorithms. The first algorithm uses $LWSE$ in Section 3 as a building block and only needs $\tilde{O}(\sqrt[3]{n})$ samples. The second algorithm is self-contained and easier to implement; its complexity is worse than the first algorithm in terms of n but has better dependency on ϵ and a much smaller hidden constant.

4.1 An Estimator with $\tilde{O}(\sqrt[3]{n})$ Samples

In this algorithm, we split the buckets into two types: $\Theta(\sqrt[3]{n^2})$ *large* buckets and the remaining *small* buckets. We estimate the partial sum of the large buckets using linear weighted sampling as in Section 3; we stratify the small buckets into different size ranges and estimate the number of buckets in each range using uniform sampling.

Theorem 2. *CombEst is an (ϵ, δ)-estimator with $O(n^{1/3}(\frac{1}{\epsilon})^{\frac{9}{2}} \log n(\log \frac{1}{\delta} + \log \frac{1}{\epsilon} + \log \log n))$ samples, where n is the number of buckets.*

Proof. We analyze the error of the estimator. Denote by $S_{large}(S_{small})$ the actual total size of large (small) buckets; by n_i the actual bucket number in level i.

In Step 2, since we are using linear weighted sampling, the expected fraction of large buckets in the samples equals to S_{large}/S. If $S_{large}/S > \epsilon_1$, then by Chernoff bound the observed fraction of large buckets in the sample is larger than $\epsilon_1/2$ with high probability, and we will get S'_{large} within $(1 \pm \epsilon_1)S_{large}$ with probability at least $1 - \delta/2$ according to Theorem 1; otherwise we lose at most $S_{large} = \epsilon_1 S$ by estimating $S'_{large} = 0$. Thus, with probability at least $1 - \delta/2$, the error introduced in Step 2 is at most $\epsilon_1 S$.

CombEst(n, ϵ, δ)

1. find t such that the number of buckets whose sizes are larger than t is $N_t = \Theta(n^{2/3})$ (we leave the detail of this step later); call a bucket *large* if its size is above t, and *small* otherwise
2. use linear weighted sampling to estimate the total size of large buckets S'_{large}:
   if the fraction of large buckets in the sample is less than $\epsilon_1/2$, let $S'_{large} = 0$;
   otherwise ignore small buckets in the samples and estimate S'_{large} using $LWSE(N_t, \epsilon_1, \delta/2)$, where $\epsilon_1 = \epsilon/4$
3. use uniform sampling to estimate the total size of small buckets S'_{small}:
   divide the small bucket sizes into levels $[1, 1+\epsilon_1), \ldots, [(1+\epsilon_1)^i, (1+\epsilon_1)^{i+1}), \ldots, [(1+\epsilon_1)^{i_0}, t)$; we say a bucket in level i $(0 \le i \le i_0)$ if its size $\in [(1+\epsilon_1)^i, (1+\epsilon_1)^{i+1})$
   make $k = \Theta(n^{1/3} \log n/\epsilon_1^4)$ samples using uniform sampling; let k_i be the number of sampled buckets in level i. Estimate the total number of buckets in level i to be $n'_i = k_i n/k$ and $S'_{small} = \sum_i n'_i (1+\epsilon_1)^i$
4. output $S'_{small} + S'_{large}$ as the estimated sum

In Step 3, it is easy to see that n'_i is an unbiased estimator of n_i. For a fixed i, if $n_i \ge \epsilon_1^2 n^{2/3}$ then by Chernoff bound the probability that n'_i deviates from n_i by more than an ϵ_1 fraction is

$$Pr[|n'_i - n_i| \ge \epsilon_1 n_i] \le exp(-ck\epsilon_1^2 n_i/n) \le exp(-c' \frac{n^{1/3}\log n}{\epsilon_1^4} \epsilon_1^2 \frac{\epsilon_1^2 n^{2/3}}{n}) = n^{-c'}$$

This means that for all $n_i \ge \epsilon_1 n^{2/3}$, with high probability we estimate n_i almost correctly, introducing a relative error of at most ϵ_1.

We round all bucket sizes of small buckets down to the closest power of $1+\epsilon_1$; this rounding introduces a relative error of at most ϵ_1.

For all levels with $n_i < \epsilon_1^2 n^{2/3}$, the total bucket size in those levels is at most

$$\sum_{0 \le i \le i_0} n_i (1+\epsilon_1)^{i+1} < \epsilon_1^2 n^{2/3} \sum_i (1+\epsilon_1)^{i+1} < \epsilon_1^2 n^{2/3} \frac{t}{\epsilon_1} = \epsilon_1 t n^{2/3} < \epsilon_1 S_{large} < \epsilon_1 S$$

The errors introduced by those levels add up to at most ϵ_1.

Summing up, there are four types of errors in our estimated sum, with probability at least $1-\delta$ each contributing at most $\epsilon_1 S = \epsilon S/4$, so S' has an error of at most ϵS.

Now we count the total number of samples in *CombEst*. According to Theorem 1, Step 2 needs $O(\sqrt{n^{2/3}}(\frac{1}{\epsilon})^{\frac{7}{2}} \log n^{2/3}(\log \frac{1}{\delta} + \log \frac{1}{\epsilon} + \log\log n^{2/3}))$ samples of large buckets, and by our algorithm the fraction of large buckets is at least $\epsilon_1/2$. Step 3 needs $\Theta(n^{1/3}\log n/\epsilon_1^4)$ samples, which is dominated by the sample number of Step 2. Therefore the total sample number is

$$O(n^{1/3}(\frac{1}{\epsilon})^{\frac{9}{2}} \log n(\log \frac{1}{\delta} + \log \frac{1}{\epsilon} + \log\log n)).\square$$

There remains to be addressed the implementation of Step 1. We make $n^{1/3}\log n$ samples using uniform sampling and let t be the size of the $2\log n$-th largest bucket in the samples. Let us first assume all the sampled bucket have

different sizes. Let N_t be the number of buckets with size at least t; we claim that with high probability $n^{2/3} \leq N_t \leq 4n^{2/3}$. Otherwise if $N_t < n^{2/3}$, then the probability of sampling a bucket larger than t is $N_t/n < n^{-1/3}$ and the expected number of such buckets in the samples is at most $\log n$; now we have observed $2\log n$ such buckets, by Chernoff bound the probability of such event is negligible. Similarly the probability that $N_t \geq 4n^{2/3}$ is negligible. Hence t satisfies our requirement. Now if there is a tie at position $2\log n$, we may cut off at any position $c\log n$ instead of $2\log n$, and N_t will still be $\Theta(n^{2/3})$ using the same argument. In the worst case where all of them are ties, let t be this size, define those buckets with sizes strictly larger than t as large buckets and those with sizes strictly less than t as small, estimating S_{large} and S_{small} using Steps 2 and 3; estimate separately the number of buckets with size exactly t using uniform sampling – since the number is at least $\Theta(n^{2/3}\log n)$, $O(n^{1/3})$ samples are sufficient. Finally we only know the approximate number of large buckets, denoted by N_t', and have to pass N_t' instead of N_t when call $LWSE$. Fortunately an approximate count of n suffices for $LWSE$, and a constant factor error in n only adds a constant factor in its complexity.

4.2 An Estimator with $\tilde{O}(\sqrt{n})$ Samples

Next we present a sum estimator using uniform and weighted sampling with $\tilde{O}(\sqrt{n})$ samples. Recall that uniform sampling works poorly for skewed distributions, especially when there are a few large buckets that we cannot afford to miss. The idea of this algorithm is to use weighted sampling to deal with such heavy tails: if a bucket is large enough it will keep appearing in weighted sampling; after enough samples we can get a fairly accurate estimate of its frequency of being sampled, and then infer the total size by only looking at the size and sampling frequency of this bucket. On the other hand, if no such large bucket exists, the variance cannot be too large and uniform sampling performs well.

CombEstSimple(n, ϵ, δ)

1. Make $k = c_1\sqrt{n}\log\frac{1}{\delta}/\epsilon^2$ samples using linear weighted sampling. Suppose the most frequently sampled bucket has size t and is sampled k_1 times (breaking ties arbitrarily). If $k_1 \geq k/2\sqrt{n}$, output $S' = tk/k_1$ as estimated sum and terminate.
2. Make $l = \sqrt{n}/\delta\epsilon^2$ samples using uniform sampling and let a be the average of sampled bucket sizes. Output $S' = an$ as estimated sum.

Theorem 3. *CombEstSimple is an (ϵ, δ)-estimator with $O(\sqrt{n}/\epsilon^2\delta)$ samples.*

Proof. Obviously *CombEstSimple* uses $k + l = O(\sqrt{n}/\epsilon^2\delta)$ samples. Below we prove the accuracy of the estimator.

We first prove that if Step 1 outputs an estimated sum S', then S' is within $(1\pm\epsilon)S$ with probability $1-\delta/2$. Consider any bucket with size t whose frequency of being sampled $f' = k_1/k$ is more than $1/2\sqrt{n}$. Its expected frequency of being sampled is $f = t/S$, so we can bound the error $|f' - f|$ using Chernoff bound.

$$Pr[f - f' > \epsilon f] \leq exp(-ckf\epsilon^2) \leq exp(-ckf'\epsilon^2) = exp(\Theta(c1)\log\frac{1}{\delta}) = \delta^{\Theta(c1)}$$

$$Pr[f' - f > \epsilon f] \leq exp(-ckf\epsilon^2) \leq exp(-ck\frac{f'\epsilon^2}{1+\epsilon}) = exp(\Theta(c1)\log\frac{1}{\delta}) = \delta^{\Theta(c1)}$$

Choose c_1 large enough to make $Pr[|f - f'| > \epsilon f]$ less than $\delta/2$, then with probability $1 - \delta/2$, $f' = k_1/k$ is within $(1 \pm \epsilon)t/S$, and it follows that the estimated sum tk/k_1 is within $(1 \pm \epsilon)S$.

We divide the input into two cases, and show that in both cases the estimated sum is close to S.

Case 1, the maximum bucket size is greater than $S/\sqrt{n}$. The probability that the largest bucket is sampled less than $k/2\sqrt{n}$ times is at most $exp(-ck\frac{1}{\sqrt{n}}) < \delta/2$; with the remaining probability, Step 1 outputs an estimated sum, and we have proved it is within $(1 \pm \epsilon)S$.

Case 2, the maximum bucket size is no more than $S/\sqrt{n}$. If Step 1 outputs an estimated sum, we have proved it is close to S. Otherwise we use the estimator in Step 2. a is an unbiased estimator of the mean bucket size. The statistical variance of x_i is

$$var(x) \leq E[x^2] = \frac{\sum_i x_i^2}{n} \leq \frac{(\frac{S}{\sqrt{n}})^2\sqrt{n}}{n} = \frac{S^2}{n\sqrt{n}}$$

and the variance of a is $var(x)/l$. Using Chebyshev inequality, the probability that a deviates from the actual average S/n by more than an ϵ fraction is at most $var(a)/(\epsilon S/n)^2 = \sqrt{n}/l\epsilon^2 = \delta$.□

5 Lower Bounds

Finally we prove lower bounds on the sample number of sum estimators. Those lower bound results use a special type of input instances where all bucket sizes are either 0 or 1. The results still hold if all bucket sizes are strictly positive, using similar counterexamples with bucket sizes either 1 or a large constant b.

Theorem 4. *There exists no (ϵ, δ)-estimator with $o(\sqrt{n})$ samples using only linear weighted sampling, for any $0 < \epsilon, \delta < 1$.*

Proof. Consider two instances of inputs: in one input all buckets have size 1; in the other, $(1 - \epsilon)n/(1 + \epsilon)$ buckets have size 1 and the remaining are empty. If we cannot distinguish the two inputs, then the estimated sum deviates from the actual sum by more than an ϵ fraction.

For those two instances, linear weighted sampling is equivalent to uniform sampling among non-empty buckets. If we sample $k = o(\sqrt{n})$ buckets, then the probability that we see a repeated bucket is less than $1 - exp(-k(k-1)/((1 - \epsilon)n/(1 + \epsilon))) = o(1)$ (see the proof of Lemma 1). Thus in both cases with high probability we see all distinct buckets of the same sizes, so cannot distinguish the two inputs in $o(\sqrt{n})$ samples.□

More generally, there is no estimator with $o(\sqrt{n})$ samples using any combination of general weighted sampling methods with the constraint $f(0) = 0$. Recall

that weighted sampling with function f samples a bucket x_i with probability proportional to a function of its size $f(x_i)$. When $f(0) = 0$, it samples any empty bucket with probability 0 and any bucket of size 1 with the same probability, thus is equivalent to linear weighted sampling for the above counterexample.

Theorem 5. *There exists no (ϵ, δ)-estimator with $o(\sqrt[3]{n})$ samples using any combination of general weighted sampling (the sampling function f independent on n), for any $0 < \epsilon, \delta < 1$.*

Proof. Consider two instances of inputs: in one input $n^{2/3}$ buckets have size 1 and the remaining buckets are empty; in the other, $3n^{2/3}$ buckets have size 1 and the remaining are empty. If we cannot distinguish the two inputs, then the estimated sum deviates from the actual sum by more than $\frac{1}{2}$. We can adjust the constant to prove for any constant ϵ.

We divide weighted sampling into two types:

(1) $f(0) = 0$. It samples any empty bucket with probability 0 and any bucket of size 1 with the same probability, thus it is equivalent to uniform sampling among non-empty buckets. There are at least $n^{2/3}$ non-empty buckets and we only make $o(n^{1/3})$ samples, with high probability we see $o(n^{1/3})$ distinct buckets of size 1 for both inputs.

(2) $f(0) > 0$. The probability that we sample any non-empty buckets is

$$\frac{f(1)cn^{2/3}}{f(1)cn^{2/3} + f(0)(n - cn^{2/3})} = \Theta(n^{-1/3}),$$

so in $o(n^{1/3})$ samples with high probability we only see empty buckets for both inputs, and all these buckets are distinct.

Therefore whatever f we choose, we see the same sampling results for both inputs in the first $o(n^{1/3})$ samples, i.e. we cannot distinguish the two inputs with $o(n^{1/3})$ samples using any combination of weighted sampling methods.□

6 Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments, especially for pointing out an implicit assumption in the proof.

References

1. N. Alon, N.G. Duffield, C. Lund, M. Thorup. *Estimating arbitrary subset sums with few probes.* PODS 2005.
2. N. Alon, Y. Matias and M. Szegedy. *The space complexity of approximating the frequency moments.* JCSC 58:137-147, 1999.
3. Z. Bar-Yossef and M. Gurevich. *Random sampling from a search engine's index.* WWW 2006.
4. Z. Bar-Yossef and M. Gurevich. *Efficient search engine measurements.* WWW 2007.

5. Z. Bar-Yossef, R. Kumar, and D. Sivakumar. *Sampling algorithms: lower bounds and applications.* STOC 2001.
6. A. Broder, M. Fontura, V. Josifovski, R. Kumar, R. Motwani, S. Nabar, R. Panigrahy, A. Tomkins, Y. Xu. *Estimating corpus size via queries.* CIKM 2006.
7. R. Canetti, G. Even, and O. Goldreich. *Lower Bounds for Sampling Algorithms for Estimating the Average.* Information Processing Letters, 53:17-25, 1995.
8. M. Charikar, S. Chaudhuri, R. Motwani, and V. Narasayya. *Towards estimation error guarantees for distinct values.* PODS 2000.
9. N.G. Duffield, C. Lund, and M. Thorup. *Learn more, sample less: control of volume and variance in network measurements.* IEEE Trans. on Information Theory, 51:1756-1775, 2005.
10. A. Gulli and A. Signorini. *The indexable Web is more than 11.5 billion pages.* WWW 2005.
11. M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. *On near-uniform URL sampling.* WWW 2000.
12. S. Lawrence and C. Giles. *Searching the World Wide Web.* Science 280:98-100, 1998.
13. S. Lawrence and C. Giles. *Accessibility of information on the web.* Nature 400:107-109, 1999.
14. J. Liu. *Metropolized independent sampling with comparisons to rejection sampling and importance sampling.* Statist. Comput. 6:113-119, 1996.
15. R. Motwani and P. Raghavan. *Randomized Algorithm.* 1995.
16. M. Szegedy. *The DLT priority sampling is essentially optimal.* STOC 2006.

7 Appendix

Proof of Lemma 1

(1) $r(B) > i$ when there is no repeated buckets in the first i samples.

$$Pr[r(B) > i] = \frac{B}{B}\frac{B-1}{B}\dots\frac{B-(i-1)}{B} = (1-\frac{1}{B})\dots(1-\frac{i-1}{B})$$

$$E[r(B)] = \sum_{1<i\leq B+1} Pr[r(B)=i]*i = \sum_{1<i\leq B+1} Pr[r(B)\geq i] = \sum_{1\leq i\leq B} Pr[r(B)>i]$$

$Pr[r(B) > i]$ monotonically increases with B for all i, so $E[r(B)]$ also monotonically increases with B.

(2) First bound $Pr[r(B) > i]$ using the fact $e^{-2x} < 1 - x < e^{-x}$:

$$Pr[r(B) > i] \leq e^{-\frac{1}{B}}e^{-\frac{2}{B}}\dots e^{-\frac{i-1}{B}} = e^{-\frac{i(i-1)}{2B}}$$

$$Pr[r(B) > i] \geq e^{-\frac{2}{B}}e^{-\frac{4}{B}}\dots e^{-\frac{2(i-1)}{B}} = e^{-\frac{i(i-1)}{B}}$$

Using the first inequality,

$$E[r(B)] = \sum_{1\leq i\leq B} Pr[r(B)>i] \leq \sum_{1\leq i\leq B} exp(-\frac{i(i-1)}{2B})$$
$$\leq \int_1^B exp(-\frac{i(i-1)}{2B})di$$

$$= \sqrt{\frac{B\pi}{2}} exp(\frac{1}{8B}) erf(\frac{2*B-1}{2\sqrt{2B}}) - \sqrt{\frac{B\pi}{2}} exp(\frac{1}{8B}) erf(\frac{2*1-1}{2\sqrt{2B}})$$
$$\leq \sqrt{\frac{B\pi}{2}} exp(\frac{1}{8B}) = O(\sqrt{B})$$

Similarly, using the second inequality we can prove

$$E[r(B)] \geq \sum_{1 \leq i \leq B} exp(-\frac{i(i-1)}{B} = \Omega(\sqrt{B})$$

Therefore $E[r(B)] = \Theta(\sqrt{B})$.

(3) Let $b_i = \frac{B-i}{B}$, $b'_i = \frac{(1+\epsilon)B-i}{(1+\epsilon)B}$; let $a_i = \prod_{j=1..i-1} b_j$, $a'_i = \prod_{j=1..i-1} b'_j$.

It is easy to see $E[r(B)] = \sum_{1 \leq i \leq B} a_i$ and $E[r((1+\epsilon)B)] = \sum_{1 \leq i \leq (1+\epsilon)B} a'_i$, therefore $E[r((1+\epsilon)B)] - E[r(B)] \geq \sum_{1 \leq i \leq B} a'_i - a_i$. We will prove that $\Delta a_i = a'_i - a_i \geq c'\epsilon$ for all $i \in [\sqrt{B}, 2\sqrt{B}]$, which gives a lower bound on $E[r((1+\epsilon)B)] - E[r(B)]$.

Notice that $a_i = a_{i-1}b_{i-1} < a_{i-1}$. Let $\Delta b_i = b'_i - b_i = \frac{\epsilon i}{(1+\epsilon)B} > 0$.

For $i \in [\sqrt{B}, 2\sqrt{B}]$, $a'_i > a_i > exp(-\frac{i(i-1)}{B}) > e^{-4}$, therefore

$$\begin{aligned}
a'_i - a_i &= a'_{i-1}b'_{i-1} - a_{i-1}b_{i-1} = a_{i-1}(b'_{i-1} - b_{i-1}) + b'_{i-1}(a'_{i-1} - a_{i-1}) \\
&> a_{i-1}\Delta b_{i-1} + b_{i-1}\Delta a_{i-1} \\
&> a_{i-1}\Delta b_{i-1} + b_{i-1}(a_{i-2}\Delta b_{i-2} + b_{i-2}\Delta a_{i-2}) \\
&> a_i(\Delta b_{i-1} + \Delta b_{i-2}) + b_{i-1}b_{i-2}\Delta a_{i-2} \\
&\cdots \\
&> a_i(\Delta b_{i-1} + \Delta b_{i-2} + \ldots + \Delta b_1) = a_i \frac{\epsilon}{(1+\epsilon)B} * \frac{i(i-1)}{2} \\
&> e^{-4}\frac{\epsilon}{2(1+\epsilon)} = \Theta(\epsilon)
\end{aligned}$$

Finally

$$E[r((1+\epsilon)B)] - E[r(B)] > \sum_{i \in [\sqrt{B}, 2\sqrt{B}]} \Delta a_i = \Theta(\epsilon\sqrt{B}) = \Theta(\epsilon)E[r(B)]$$

(4)

$$\begin{aligned}
var(r(B)) &= E[r(B)^2] - E[r(B)]^2 \leq E[r(B)^2] \\
&= \sum_{2 \leq i \leq B+1} Pr[r(B) = i]i^2 = \sum_{2 \leq i \leq B+1} \frac{B}{B}\frac{B-1}{B}\cdots\frac{B-(i-2)}{B} * \frac{i-1}{B} * i^2 \\
&< \sum_{2 \leq i \leq B+1} \frac{i^3}{B} exp(-\frac{(i-1)(i-2)}{2B}) \\
&\leq (\frac{9}{16B}e^{-\frac{5}{2B}}\sqrt{2\pi B}(4B+3)erf(\frac{2x-3}{2\sqrt{2B}}) - \frac{1}{4}e^{-\frac{x^2-3x+2}{2B}}(4x^2+6x+8B+9))|_2^{B+1} \\
&= O(B)
\end{aligned}$$