Estimating arbitrary subset sums with few probes

Noga Alon† Nick Duffield* Carsten Lund* Mikkel Thorup*
† School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel
*AT&T Labs—Research, 180 Park Avenue, Florham Park, NJ 07932, USA
nogaa@post.tau.ac.il {duffield,lund,mthorup}@research.att.com

ABSTRACT

Suppose we have a large table T of items i, each with a weight w_i, e.g., people and their salary. In a general preprocessing step for estimating arbitrary subset sums, we assign each item a random priority depending on its weight. Suppose we want to estimate the sum of an arbitrary subset $I \subseteq T$. For any $q > 2$, considering only the q highest priority items from I, we obtain an unbiased estimator of the sum whose relative standard deviation is $O(1/\sqrt{q})$. Thus to get an expected approximation factor of $1 \pm \varepsilon$, it suffices to consider $O(1/\varepsilon^2)$ items from I. Our estimator needs no knowledge of the number of items in the subset I, but we can also estimate that number if we want to estimate averages.

The above scheme performs the same role as the on-line aggregation of Hellerstein et al. (SIGMOD'97) but it has the advantage of having expected good performance for any possible sequence of weights. In particular, the performance does not deteriorate in the common case of heavy-tailed weight distributions. This point is illustrated experimentally both with real and synthetic data.

We will also show that our approach can be used to improve Cohen's size estimation framework (FOCS'94).

Categories and Subject Descriptors

E.1 [**Data**]: Data Structures; E.5 [**Data**]: Files; F.2 [**Theory of Computation**]: Analaysis of Algorithms and Problem Complexity; G.3 [**Mathematics of Computing**]: Probability and Statistics; H.3 [**Information Systems**]: Information Storage and Retrieval

General Terms

Algorithms, Experimentation, Measurement, Performance, Theory

Keywords

Databases, Preprocessing, Sampling, IP Flows

1. INTRODUCTION

We use "priority sampling" to preprocess a large table of items, each of which carries a weight. For example, the items could be people weighted by their salary. The preprocessing assigns priorities to the items. After preprocessing, we can estimate the total weight of an arbitrary subset I using only a few high priority samples from the samples I. For example we could estimate the total salary of all individuals in Massachusetts of age between 30 and 32, or of people with surname Gates. Using unit weights, we could also get the number of people in a subset, and thus estimate averages. The basic idea of sampling for such aggregate queries is standard (see, e.g., [4, 13, 15, 16]). What is new is that the estimates do not deteriorate in the common case of a heavy tailed weight distribution [5, 17]. Figure 1 illustrates how big a difference priority sampling makes on a concrete data set.

1.1 Priority Sampling

Our scheme relates to our recent work in [12], and we shall return to this relationship in Section 1.6. We are given a table T of items i each of which carries a non-negative weight w_i. In a preprocessing step, we generate for each item i an independent uniformly distributed random number $\alpha_i \in (0, 1)$, and give item i the priority $q_i = w_i/\alpha_i$.

We now want to estimate the total weight of an arbitrary selected subset $I \subseteq T$ of items. The selection of I should be oblivious to the generated priorities. We collect a *sample* $S \subset I$ consisting of the k items with the highest priority in I. In case of equal priorities, those with smaller number have higher priority. Moreover, we let the *threshold* τ be the $(k+1)^{\text{st}}$ highest priority in I—if $|I| \leq k$, there is no need to sample I. Magically it turns out that $\sum_{i \in S} \max\{\tau, w_i\}$ is an unbiased estimator of the total weight in I, that is,

$$\mathsf{E}\left[\sum_{i \in S} \max\{\tau, w_i\}\right] = \sum_{i \in I} w_i \tag{1}$$

We call the above a *priority sample of size k for I*, and it is illustrated in Figure 2.

Note that if a new item j with weight w_j is added to T, we just have to generate a random $\alpha_j \in (0, 1)$ and give j priority $q_j = w_j/\alpha_j$. This defines the role of j in any later subset sum.

Also, note that the above scheme only works for non-negative weights. However, we can treat items with negative weights

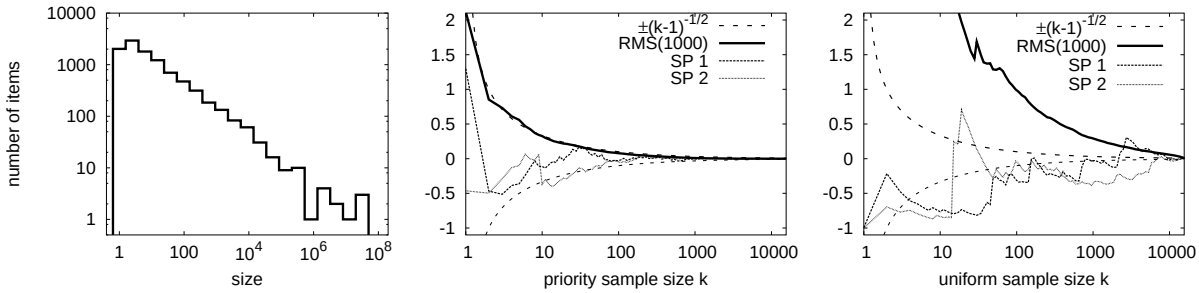

Figure 1: Experiment with the 15,566 files in a unix directory. They have total size 2,139,913,858, maximal size 78,438,400, average size 137,473, median size 2,329 and 20 files of minimal size 1. We are trying to estimate the total size from k samples. Left: histogram of log weights. Center: relative estimation error for priority sampling as function of sample size k, for two sample paths, Root Mean Square error over 1000 sample paths, and compared with conjectured Relative Standard Deviation envelope of $1/\sqrt{k-1}$. Right: relative estimation error for uniform sampling without replacement.

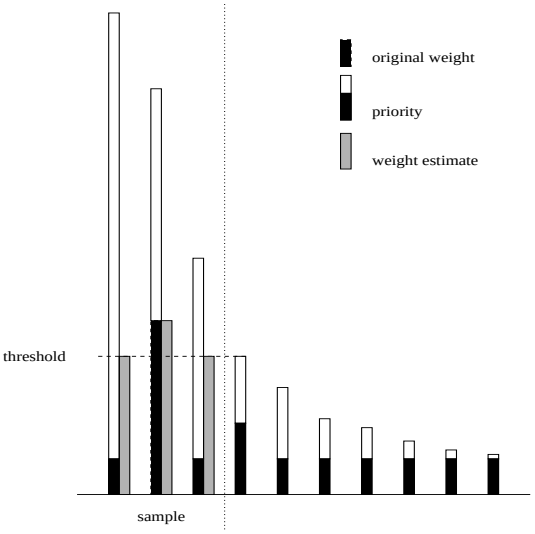

Figure 2: Priority sample of size 3 for a set of 10 weighted items. The sum of the shaded weight estimates approximates the sum of the black original weights.

separately, negate them, and give them priorities as described above. To estimate a subset sum, we take the estimate of the sum of the positive weights and subtract the estimate of the sum of the negated negative weights. (The relative error can, however, increase this way, as there may be cancellations.)

Finally, note that our scheme works for *arbitrary* subsets. If the subsets were restricted in some specific way,

1.2 Storing the Table

In this paper, we assume that items from the table T are stored so that items from a selected subset I can be retrieved in order of decreasing priority. We do not specify how this is to happen. We assume that whatever method is used to retrieve items from selected subsets, the same method is augmented to retrieve items in order of decreasing priority.

As the most primitive example, if the table T was just a flat file that we would scan for a selected elements, then our preprocessing sorts the items in T in order of decreasing priority. The ones of higher priority can be put in faster memory or we can rely on the caching hierarchy to improve their access time. Now, to get a good estimate of the total salary of people with surname ending 'son', we scan the sorted table until we have found, say, 102 such people. The last gives the threshold for the sample of the first 101 and we get an estimate of the total wealth using (1). As we shall see in Section 1.4, we expect an error of about 10% for any distribution of salaries.

A more advanced alternative is to store the table in a data base indexing attributes such as name, state, and age. To support priority sampling, each index should now support retrieval in order of decreasing priority. Algorithms for such priority observing indexing in external memory are described in [2, 3].

1.3 Selecting Arbitrary Subsets

To illustrate the significance of arbitrary subsets, suppose a large retailer with many stores saved all their sales in records containing information such as item, location, time, and price. (Wal-Mart, for example, has 3,600 stores in the US, and roughly 100 million customers each week; see [14].) They might ask questions like how many days of rain does it take before we get a boom in the sale of rain gear. Knowing this would tell them how long they would need to order and disperse the gear if the weather report promised a long period of rain. Now, the weather information was not part of the sales records, but if they had a data base with historical weather information, they could look up each sampled sales record with rain gear, and check how many days it had rained at that location before the sale. The point in this example is that selection can be based on external information not even imagined relevant at the time when measurements are made.

1.4 How Many Samples?

A fundamental question is how many samples we need to ensure a small variability in our estimate. Since our estimate is unbiased, we use the relative standard deviation of the estimate as a measure of its variability. More precisely, let W be the total weight in a subset I and let $\widehat{W}$ be our estimate of W. Thus, with our previous notation, $W = \sum_{i\in I} w_i$, $\widehat{W} = \sum_{i\in S} \max\{w_i, \tau\}$, and $\mathsf{E}\left[\widehat{W}\right] = W$. The relative standard deviation (Rsd) is then

$$\mathsf{Rsd}\left[\widehat{W}\right] = \sqrt{\mathsf{Var}\left[\widehat{W}\right]}/W$$
$$= \sqrt{\mathsf{E}\left[\left(\frac{\widehat{W}-W}{W}\right)^2\right]} \geq \mathsf{E}\left[\frac{|\widehat{W}-W|}{W}\right]$$

The last inequality states that the relative standard deviation is an upper-bound on the expected relative error.

CONJECTURE 1. *With a priority sample of size k, the relative standard deviation is less than $1/\sqrt{k-1}$.*

The experiment shown in Figure 1 conforms nicely with our conjecture. For n unit weights, Mathematica (or a simple computation) tell that the variance is $\mathsf{Var}(\widehat{W}) = n(n-k)/(k-1)$, hence that $\mathsf{Rsd}(\widehat{W}) = \sqrt{(1-k/n)/(k-1)}$, so the conjecture is true for unit weights, and tight for $n \to \infty$.

Our conjecture is based on the intuition that our estimates benefit from skew, hence that identical weights form a worst-case distribution. For example, we note that the more dominant an item i is, the more likely it is to be sampled, and this generally reduces variance. Moreover, if item i is not sampled, it is because $\tau \geq q_i > w_i$ and all sampled items have weight estimate at least τ. Thus dominant items are guaranteed a direct or indirect contribution to the estimated total. We will present experimental evidence supporting our intuition that skew decreases the variance.

In this paper, we establish Conjecture 1 asymptotically, proving

THEOREM 1. *With a priority sample of size $k \geq 2$, the relative standard deviation is $O(1/\sqrt{k})$. For the expected relative error, this bound holds when $k \geq 1$.*

In particular it follows that to get an expected approximation factor of $1 \pm \varepsilon$, it suffices to use a priority sample of size $O(1/\varepsilon^2)$.

An objection to priority sampling could be that after the preprocessing, we cannot make independent priority samples. However, we note that both with the conjecture and with the theorem, we get slightly better bounds if we use a priority sample of size $2k$ than if we took the mean of two independent priority samples of size k. More precisely, for the priority sample of size $2k$, we consider $2k+1$ items, and with our conjecture, we get a relative standard deviation bound of $1/\sqrt{2k-1}$. However, for two independent priority samples of size k, we consider $2(k+2)$ items, and for their mean, we get relative standard deviation bound of $(1/\sqrt{k-1}))/\sqrt{2} = 1/\sqrt{2k-2} > 1/\sqrt{2k-1}$. Thus we are better off increasing the priority sample size than we would be averaging over independent samples.

Instead of looking at the standard deviation, we can bound the probability that the relative error exceeds a certain value. More precisely, we show [1]

THEOREM 2. *For $\varepsilon \leq 1$, the probability that the relative error $\frac{|\widehat{W}-W|}{W}$ exceeds ε is bounded by $\exp(-\Omega(\varepsilon^2 k))$.*

Thus, for a target error probability of p_{error}, it suffices to take $O(-(\log p_{error})/\varepsilon^2)$ samples.

1.5 Alternatives

We will now compare our scheme with what can be done using standard probabilistic techniques.

1.5.1 Uniform Sampling

The most basic idea is to make a random permutation of the n items in the table T. Let T consist of the first pn items. Now, each item is in T with probability p, so by linearity of expectation,

$$\mathsf{E}\left(\frac{1}{p}\sum_{i\in I\cap T} w_i\right) = \sum_{i\in I} w_i \tag{2}$$

This estimator is equivalent to the one used in [15]. Priority sampling is compared with this uniform sampling scheme in Figure 1. The main caveat of uniform sampling is that each item is picked with the same probability, and that is not good for heavy tailed distributions. In [15, Eq. (1)], the problem shows up in their deviation bound which is proportional to the difference between the maximum and minimum weight.

Another caveat of the uniform sampling scheme is that it doesn't quite have the right format. More precisely, suppose we retrieve the k first items from the subset I in the permuted table T. How do we get an unbiased estimator of the sum of the subset? If we knew that the k^{th} item from I was item number q in the permuted table T, we would like to apply (2) with $p = q/n$, but this would not be an unbiased estimator since p now depends on the distribution of the items from I among all the items in T. It is easy to see that this gives a bias toward bigger estimates. This caveat of uniform sampling is bypassed in Figure 1 where we consider the special case of $I = T$.

1.5.2 Dividing Weights into Levels

One standard idea to accommodate heavy tailed weight distributions is to divide items into levels J_ℓ with weights in $[2^\ell, 2^{\ell+1})$. Within each level, the items have similar weights, and then we can use uniform sampling as in Section 1.5.1. Now, if r is the ratio of the biggest weight over the smallest weight, we need $\log_2 r$ levels. In our salary example, we would need to make independent samples on more than 20

[1] For $f_n, g_n > 0$, $f_n = \Omega(g_n)$ iff $g_n = O(f_n)$

levels ($2^{20} \approx 1,000,000$, which is, at least, the ratio of salary in the world today). We can reduce the number of levels by increasing the ratio of the largest and smallest element in each level from 2 to some higher value, but that will increase the relative standard deviation in each level. By contrast, priority sampling gives a clean and efficient mix of big and small weights without any need for division into levels.

1.5.3 Weighted Sampling with Replacement

Another idea for dealing with skewed weight distributions is to make a list of the items where each element is an item picked with probability proportional to its weight. Then each list segment forms a sample with replacement. In the presence of heavy tails, the list is expected to have many duplicates. Also, since we want to represent all subsets, including subsets with a single item of small weight, we have to keep sampling for the list till all items have been included at least once. Instead of a permutation of length n, with many small items, we need a list of length [2]

$$\Theta\left(\frac{\sum_{i\in T} w_i}{\min_{i\in T} w_i} \log n\right).$$

We note that with a heavy tailed distribution, the expected length is unbounded because the expected ratio of the sum of all weights to the smallest weight is infinite. For the concrete data set from Figure 1, we expect a list length in the order of $10,000,000,000$, which is unattractive compared with our priority determined permutation of length $15,566$.

In addition to the length problem, we have the same format problems as in Section 1.5.1, that is, if we consider the first k items from a given subset I in the list, it is not clear how we get an unbiased estimator of the total weight in I.

1.5.4 Splitting Weights into Units

Assuming integer weights, we can split each item i into w_i subitems of unit weight. We now have a table of $\sum_{i\in T} w_i$ items. Essentially, this becomes like weighted sampling but with an upper-bound w_i on how many copies we can get of item i.

Supposing that we had an upper bound $k_{\max}$ on how many samples we would ever consider, we would at most need the $k_{\max}$ highest priority unit subitems from each original item i, giving us a combined bound of $\max\{k_{\max}, w_i\}$. However, just to get down to an expected 1% error, even if we believe the strong bound in Conjecture 1, we would need $k_{\max} = 10,001$. Moreover, typically, we do not want to limit k at all, as the estimate precision is something that should be determined independently for each subset sum query.

If the weights are non-integers or the minimum weight is not 1, we can use the same idea but with $\left\lfloor \frac{\sum_{i\in T} w_i}{\min_{i\in T} w_i} \right\rfloor$ subitems with weights in $[1, 2]$. We note that this is a better bound on the sequence length than the $\Theta\left(\frac{\sum_{i\in T} w_i}{\min_{i\in T} w_i} \log n\right)$ bound we had with standard weighted sampling with replacement, yet it is much larger than n if the weight distribution is heavy tailed. Taking the example from Figure1, we now get a permutation of 2,139,913,858 unit subitems as opposed to our priority permutation of the 15,566 original items.

[2] $f_n = \Theta(g_n)$ iff $f_n = O(g_n)$ and $f_n = \Omega(g_n)$

1.5.5 Weighted Sampling without Replacement

One might instead try weighted sampling without replacement to avoid the duplicates with replacement. The book [7] mentions 50 such schemes, but none of these provides estimates of sums. The basic problem is that the probability that a given item is included in the sample is a complicated function of all the involved weights. The dependence between weights is implicit in priority sampling where the ordering encodes information about all the weights.

1.5.6 Cohen's Size-estimation Framework

We now compare our scheme with Cohen's size-estimation framework [8] which has many similarities with our framework. Like us, she considers n weighted items and gives each item i a randomized priority correlated with its weight w_i. To estimate the total weight of a given subset I, she considers the lowest priority item from I. This way she gets an unbiased estimator based on a single sampled item. To control the variance, she repeats the experiment k times independently. Averaging the resulting estimators, she gets one with smaller variance. For the averaged estimator, she proves error bounds similar to those we presented in Theorems 1 and 2.

For contrast, we get our k samples from a single set of priorities. Thus, from our perspective, the caveat in Cohen's scheme is that it requires not one but $k_{\max}$ randomized priorities for each item. As discussed previously, we should think of $k_{\max}$ as at least 10,000, and really, we do not want any limit on $k_{\max}$. Thus, storing $k_{\max}$ sets of n independent priorities would be a major problem in space.

It should be noted that in Cohen's applications, the selected sets are known in advance as the reachability sets of vertices in a graph. She can then generate one set of priorities at the time, find the estimators of all selected subsets, and then reuse the space for the next set of priorities. In such applications, she only uses linear space, but she still has a disadvantage of using kn random numbers where we only need n (in both schemes it can be seen that we expect to use $\log_2 n$ bits per number). Thus we reduce the need for randomness in Cohen's work.

At the end of [8] Cohen hints at a variant of her scheme that like ours is based on a single priority ordering. She is not very specific about the exact estimator and its properties, but based on personal communication with her, we believe that such a variant would be more complicated than ours, and that it would not work as well for heavy-tailed distributions.

1.5.7 Summary of Advantages

Summing up, the advantage of our new scheme is that it combines all of the following qualities:

- It works well for arbitrary sequences of weights, unlike uniform sampling.
- It uses linear space, unlike weighted sampling with replacement, weight splitting, and Cohen's original scheme.
- It provides simple unbiased weight estimators, unlike

previous schemes for weighted sampling without replacement.

- Both sampling and estimation are simple to implement.

1.6 Our Previous Streaming Scheme

The idea for this paper grew out of work on a streaming type problem [12]. At a high speed Internet router, we collected flows of different byte sizes. We created *one* priority sample for these flows, and used it to estimate arbitrary subset sums.

More precisely, let I be the set of these flows. As flow $i \in I$ of size w_i streams by, we generate $\alpha_i \in (0,1)$ and the priority $q_i = w_i/\alpha_i$. Using a priority queue, we maintain the $k+1$ flows of highest priority. The k highest form the sample set S and the $(k+1)^{\text{st}}$ highest is the threshold τ. With this scenario, we can view the sampling of S as a weight-sensitive version of reservoir sampling [19]. In contrast to the weighted reservoir sampling in [9], our scheme is without replacement so that we do not get any duplicates.

Now, let $H \subseteq I$ be a subset of flows, e.g., those from a given source to a given destination IP address. To estimate the total size of flows in H, i.e., the total traffic from the source to the destination, we can use the estimator $\sum_{i\in S\cap H} \max\{\tau, w_i\}$, which we showed was unbiased, i.e.,

$$\mathsf{E}\left(\sum_{i\in S\cap H} \max\{\tau, w_i\}\right) = \sum_{i\in H} w_i \tag{3}$$

In [12], we summed over all items in $S\cap H$. If S is large, this may be a lot of work, and if S is small, too much information may be lost on smaller subsets.

1.6.1 What's (not) New?

First we note that (1) follows from (3) with $H = I$, so the unbiasedness of our estimator follows from our original work on priority sampling [12]. In this paper we suggest organizing a large table so as to provide priority samples of arbitrary selected subsets. Similar organizations for other sampling schemes were used in [8, 15].

The fundamental discovery of this paper is that the errors of our estimates are bounded in terms of the sample size for any distribution of weights. We conjectured the worst-case behavior in Conjecture 1 which we will support with experiments. Moreover, we will prove the tight asymptotic bounds from Theorem 1 and 2. We note that bounds in terms of the sample size were not as relevant to our previous work [12], for such bounds only tell us that the priority sample S from I gives a good estimate of the total weight in I. However, they do not tell us how well S can be used to estimate the total weight of an arbitrary subset $H \subseteq I$. That quality depends on the specifics of the distribution of H in I.

1.6.2 Combining Old and New

For a very large data set I, it may be worthwhile using our old and new schemes in tandem. First we compress I to a priority sample T of I. We then give each item $i \in T$ a new weight $w_i' = \max\{w_i, \gamma\}$ where γ is the threshold of T. Next we use T as a compressed input table for our new scheme, generating a new independent priority for each $i \in T$. To estimate the weight of a subset $J \subseteq I$, we generate a priority sample S of $J \cap T$, and return $\sum_{i\in S} \max\{w_i', \tau\}$ where τ is the threshold of T. From (1) and (3) it follows that this is an unbiased estimator of $\sum_{i\in J} w_i$.

2. BOUNDING THE ERROR PROBABILITY

In this section, we prove Theorem 2, bounding the probability of a certain relative error in our weight estimate. More precisely, with $W = \sum_{i\in I} w_i$ the weight and $\widehat{W} = \sum_{i\in I} \widehat{w}_i$ the estimated weight of the set I, the *relative error* is $\frac{|\widehat{W}-W|}{W}$. For $\varepsilon \le 1$, we will show that

$$\Pr\left[\frac{|\widehat{W}-W|}{W} > \varepsilon\right] < 2\exp(-k\varepsilon^2/6) \tag{4}$$

To prove this, we first note that our weight estimate is determined exclusively by the threshold τ. More precisely, we know that we will include all items i with $w_i > \tau$. Any other sampled items has weight estimate of τ, and the total number of sampled items is k. Hence, with $W^{\Delta}(\tau) = \sum_{w_i>\tau}(w_i - \tau)$, our weight estimate becomes

$$\widehat{W}(\tau) = \tau k + W^{\Delta}(\tau).$$

We note that there can be at most k items bigger than τ, and this puts a lower bound on τ and $\widehat{W}(\tau)$. Otherwise $\widehat{W}(\tau)$ is a continuously increasing function of τ.

2.1 Underestimates

We will now bound the probability that the weight estimate is too small, that is,

$$\widehat{W} < (1-\varepsilon)W$$

Define τ^- such that $\widehat{W}(\tau^-) = (1-\varepsilon)W$; if this is not possible, the event is impossible. We are asking for the probability that the measured threshold τ is less than τ^-, but this is the case if and only if we have less than $k+1$ items i with priority $q_i \ge \tau^-$. The expected number of priorities above τ^- is

$$\begin{aligned}
\mu &= \sum_{i\in I} \Pr[q_i \ge \tau^-] \\
&= \sum_{i\in I} \Pr[\alpha_i \le w_i/\tau^-] \\
&= \sum_{i\in I} (\min\{\tau^-, w_i\}/\tau^-) \\
&= (W - W^{\Delta}(\tau^-))/\tau^-
\end{aligned}$$

On the other hand, we have

$$(1-\varepsilon)W = \widehat{W}(\tau^-) = \tau^- k + W^{\Delta}(\tau^-)$$

so

$$k = (W - W^{\Delta}(\tau^-) - \varepsilon W)/\tau^-$$

In particular, it follows that

$$k \le (1-\varepsilon)\mu$$

Now standard Chernoff bounds (see, e.g., [1]) show that the probability of getting at most k priorities $q_i \ge \tau^-$ is at most

$$\exp(-\mu\varepsilon^2/2) < \exp(-k\varepsilon^2/2)$$

2.2 Overestimates

Next we consider the probability that the weight estimate is too large, that is,

$$\widehat{W} > (1+\varepsilon)W.$$

The analysis is very similar to that for underestimates. We define τ^+ such that $\widehat{W}(\tau^+) = (1+\varepsilon)W$. We are asking for the probability that the measured threshold τ is bigger than τ^+, but this is the case if and only if we have at least $k+1$ items i with priority $q_i > \tau^+$. The expected number of priorities $q_i > \tau^+$ is

$$\mu^+ = (W - W^{\Delta}(\tau^+))/\tau^+.$$

while

$$k = (W - W^{\Delta}(\tau^+) + \varepsilon W)/\tau+$$

In particular, it follows that

$$k+1 > k \geq (1+\varepsilon)\mu^+.$$

Using standard Chernoff bounds, the probability of getting at least $k+1$ priorities $q_i > \tau^+$ grows with the expected number μ^+ which we for an upper bound can replace by $\mu^* = (k+1)/(1+\varepsilon) > \mu^+$. Then the probability of an overestimate is bounded by

$$\left[\frac{e^{\varepsilon}}{(1+\varepsilon)^{(1+\varepsilon)}}\right]^{\mu^*} = \left[\frac{e^{\varepsilon/(1+\varepsilon)}}{(1+\varepsilon)}\right]^{k+1}. \tag{5}$$

For $\varepsilon \leq 1$, we use the simpler bound

$$\exp(-\mu^*\varepsilon^2/3) \leq \exp(-k\varepsilon^2/6)$$

Together with the underestimate bound, this completes the proof of (4).

Note that for $\varepsilon \geq 1$, our probability bound is of the form $\Omega(1/(1+\varepsilon))^{k+1}$. Thus, for fixed k, the probability only decreases polynomially in the factor $(1+\varepsilon)$ that the estimate is bigger than the true total weight.

3. THE RELATIVE STANDARD DEVIATION

We will now bound the relative variance (Rvar) which is the square of the relative standard deviation, that is, $\mathsf{Rvar}\left[\widehat{W}\right] = \mathsf{Var}\left[\widehat{W}\right]/W^2 = \mathsf{E}\left[\left(\frac{\widehat{W}-W}{W}\right)^2\right] = \mathsf{Rsd}\left[\widehat{W}\right]^2$. To prove Theorem 1, for $k > 1$ samples, we show that

$$\mathsf{Rvar}\left[\widehat{W}\right] = O(1/k)$$

Let $R^{[a,b]}$ denote the contribution to the relative variance when the relative error $\frac{|\widehat{W}-W|}{W}$ is in the interval $[a,b]$. Thus $\mathsf{Rvar}\left[\widehat{W}\right] = R^{[0,\infty)}$.

First we note that if the relative error is less than $1/\sqrt{k}$, the contribution to the relative variance is less than $1/k$, that is,

$$R^{[0,1/\sqrt{k}]} < 1/k.$$

Next suppose the relative error is between $1/\sqrt{k}$ and 1. We are going to consider exponentially increasing intervals for the relative error and use (4) to bound the probability of being in that interval.

$$\begin{aligned}
&R^{[1/\sqrt{k},1)} \\
&= \sum_{i=0}^{\lfloor \log \sqrt{k} \rfloor} R^{[2^i/\sqrt{k}, 2^{i+1}/\sqrt{k})} \\
&< \sum_{i=0}^{\lfloor \log \sqrt{k} \rfloor} \left(\begin{array}{l} \Pr\left[\frac{|\widehat{W}-W|}{W} \in [2^i/\sqrt{k}, 2^{i+1}/\sqrt{k}]\right] \\ \times (2^{i+1}/\sqrt{k})^2 \end{array} \right) \\
&< \sum_{i=0}^{\lfloor \log \sqrt{k} \rfloor} \left(2\exp(-(2^i/\sqrt{k})^2 k/6) 2^{2i+2}/k\right) \\
&= \sum_{i=0}^{\lfloor \log \sqrt{k} \rfloor} \left(2\exp(-(2^{2i}/6) 2^{2i+2}/k\right) \\
&= O(1/k).
\end{aligned}$$

Finally we consider the overestimate case where the relative error is more than 1. Again we consider exponentially increasing intervals, but this time bounding the probability with (5).

$$\begin{aligned}
&R^{[1,\infty)} \\
&= \sum_{i=0}^{\infty} R^{[2^i, 2^{i+1})} \\
&< \sum_{i=0}^{\infty} \left(\Pr\left[\frac{|\widehat{W}-W|}{W} \in [2^i, 2^{i+1}\right] (2^{i+1})^2\right) \\
&< \sum_{i=0}^{\infty} \left(\left[\frac{e^{2^i/(1+2^i)}}{1+2^i}\right]^{k+1} 2^{2i+2}\right) \\
&= \sum_{i=0}^{\infty} O\left(\left[\frac{\sqrt{e}}{1+2^i}\right]^{k-1}\right) \\
&< O\left(\left[\frac{\sqrt{e}}{2}\right]^{k-1}\right) \\
&= O(1/k).
\end{aligned}$$

Above, the sum is dominated by its first term because the exponent $k-1$ is positive for $k \geq 2$. Thus, for $k \geq 2$, we have bounded by relative variance as $\mathsf{Rvar}\left[\widehat{W}\right] = R^{[0,\infty)} = R^{[0,1/\sqrt{k})} + R^{[1/\sqrt{k},1)} + R^{[1,\infty)} = O(1/k)$. In particular, it follows that $\mathsf{Rsd}\left[\widehat{W}\right] = \sqrt{\mathsf{Rvar}\left[\widehat{W}\right]} = O(1/\sqrt{k})$, as desired.

3.1 The Expected Relative Error

The expected relative error is bounded by the relative standard deviation, but for $k = 1$, the relative standard deviation is infinite. However, the infinite contribution to the relative variance was in the sum bounding $R^{[1,\infty)}$. For the expected relative error, the corresponding sum is

$$\sum_{i=0}^{\infty} \left(\left[\frac{e^{2^i/(1+2^i)}}{1+2^i}\right]^{k+1} 2^{i+1}\right) = \sum_{i=0}^{\infty} O\left(\left[\frac{\sqrt{e}}{1+2^i}\right]^{k}\right).$$

Thus, our exponent is increased from $k-1$ to k. In particular, the exponent is 1 for $k = 1$, and then the whole sum is bounded by a constant. Thus, we conclude that the expected relative error is constant for $k = 1$. Combined

with the standard deviation bound this implies that the expected relative error is bounded by $O(1/\sqrt{k})$ for any sample size $k \geq 1$. This completes the proof of Theorem 1.

4. EXPERIMENTAL PERFORMANCE

In this section we demonstrate the performance of our sampling technique in two ways. The results are presented in Figures 1, 3–6. Each figure is based on a different set of weights displayed in the left hand plot. The plot is a histogram of the discrete logarithms of the weights, with exponentially growing bin widths, replotted on a log scale.

In the center plots, we tested priority sampling, calculating the relative error $(\widehat{W}_k - W)/W$ as a function of the sample size k. Two sample paths, SP 1 and SP 2 are presented. Each sample path is generated from a single set of random α_i and priorities q_i; if separate random variables were used for each value of k the curves would be noticeably more jagged. We also present the root mean square (RMS) of the relative error over 1000 sample paths. We note that the RMS is more sensitive to big errors than the regular mean would have been. The RMS is our measured relative standard deviation $\sqrt{\mathsf{E}\left[(\widehat{W}_k - W)/W\right]}$. By Conjecture 1, the relative standard deviation is bounded by $1/\sqrt{k-1}$ for all weight distributions, and we include this envelope for comparison.

The right plots display the same quantities for uniform sampling without replacement as used in [15].

In Figure 1 we considered files in a Unix directory. As synthetic data we use Pareto distributions displaying different degrees of heavy-tailedness. We consider Pareto(β) Distributions with $\beta = 0.5, 1.0, 1.5, 2.5$ in Figures 3–6. The Pareto distribution of a random variable X has complementary cumulative distribution function $P[X > x] = x^{-\beta}$ for x greater than some x_0. The distribution has infinite variance for $\beta \leq 2$ and infinite mean for $\beta \leq 1$. The smaller β, the more heavy tailed the distribution is. We note that Pareto-like distributions are common in real life [5], often with β close to 1. For example, for the Internet flow sizes studied in [11], we found $\beta \approx 1.05$.

We summarize the experiments as follows. Firstly, with priority sampling, the relative error is generally smaller for the more heavy-tailed distributions and the RMS stays nicely within the bound of Conjecture 1. In particular, priority sampling does very well in the presence of a very large weight like in Figure 3. Priority sampling will always include such a large weight, either directly as a sample, or indirectly because the threshold exceeds the weight and then all weight estimates are larger. Analytically, we know that the bound of Conjecture 1 holds and is tight in case of unit weights. Thus we see our experiments as a good indication that the conjecture holds true.

Second we see that uniform sampling does somewhat better in the beginning in the least heavy-tailed distribution in Figure 6 but much worse in the more heavy-tailed cases in Figures 1, 3, and 5. The difference is particularly evident in Figure 3, where one large weight is sampled well by priority sampling, but leads to wildly inaccurate estimates in uniform sampling. More precisely, in the regular sample paths, a large weight mostly leads to underestimates because it is not sampled. However, with some small probability it is sampled early with a very large estimate and this has a strong impact on the RMS.

When comparing uniform and priority sampling, it is not just a question of uniform doing better without heavy tails and priority sampling doing better with heavy tails. The significant point is that priority sampling is *always* expected to do well with a reasonable number of samples. For example, in our experimental sample paths, we see that priority sampling always got the answer within a 50% error in 10 samples. For contrast, with uniform sampling and heavy tails, it is not until most of the 10,000 items are sampled that we start getting good convergence. Since the underlying weight distribution may not be known and since heavy-tailed distributions are common in practice [5], it is crucial to have a scheme like priority sampling that can be trusted for any underlying weight distribution.

5. EXTENSIONS

In this section, we briefly sketch how our approach can be used to generate some other types of estimates.

5.1 Secondary Weights

Suppose our items have a secondary weight x_i that we want to estimate. For sampled items we can use $\widehat{x}_i = x_i \max\{\tau, w_i\}/w_i$ as an unbiased estimator of x_i. This way we can view the weight w_i as the importance attached to the item i. Note that we do not provide any error bounds for $\widehat{x}_i$ as they depend on the correlation between x_i and w_i.

5.2 Identifying Small Sums

Suppose we do not have direct access to items from a selection H but that they are contained in a larger set I that we can scan in priority order. After scanning many items from I without seeing H, we would like to conclude that H constitutes a small fraction of the weight in I.

As described in Section 1.6, we can use a priority sample of I to estimate the total weight of any subset H of I. Generalizing Theorem 2, one can show that if our sample size is k and the weight of H is a fraction f of the weight of I, the probability that the relative error of estimating H exceeds ε is bounded by $\exp(-\Omega(\varepsilon^2 fk))$. Getting no samples from a non-empty H corresponds to $\varepsilon = 1$, so the probability of this event is bounded by $\exp(-\Omega(fk))$.

6. CONCLUDING REMARKS

We have presented a scheme for estimation of selection sums based on samples which does not degenerate with skewed weight distribution. In Conjecture 1 we suggested that with k samples, the relative standard deviation is bounded by $1/\sqrt{k-1}$. We proved this bound asymptotically and supported it by experiments. Very recently Szegedy [18] announced a proof of Conjecture 1.

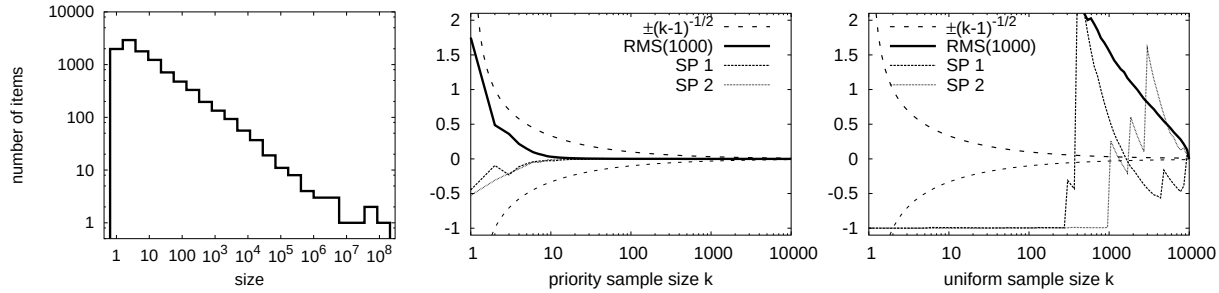

Figure 3: 10,000 Weights from Pareto(0.5) Distribution. Total = 2.83×10^8, Max = 1.34×10^8, Mean = 2.83×10^4, Median 3.97, Min = 1.00.

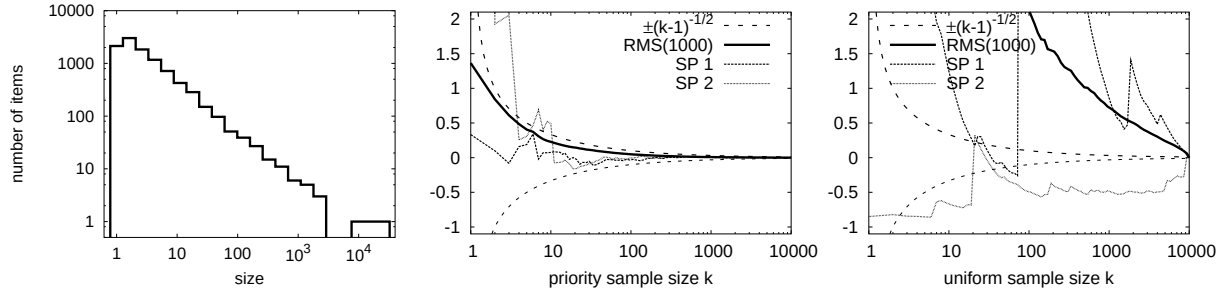

Figure 4: 10,000 Weights from Pareto(1.0) Distribution. Total = 1.41×10^5, Max = 2.56×10^4, Mean = 14.1, Median = 2.00, Min = 1.00

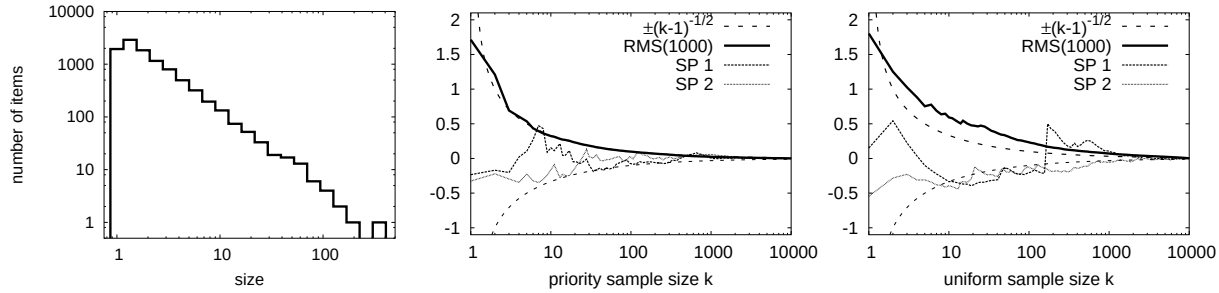

Figure 5: 10,000 Weights from Pareto(1.5) Distribution. Total = 2.82×10^4, Max = 351., Mean = 2.82, Median = 1.59, Min = 1.00

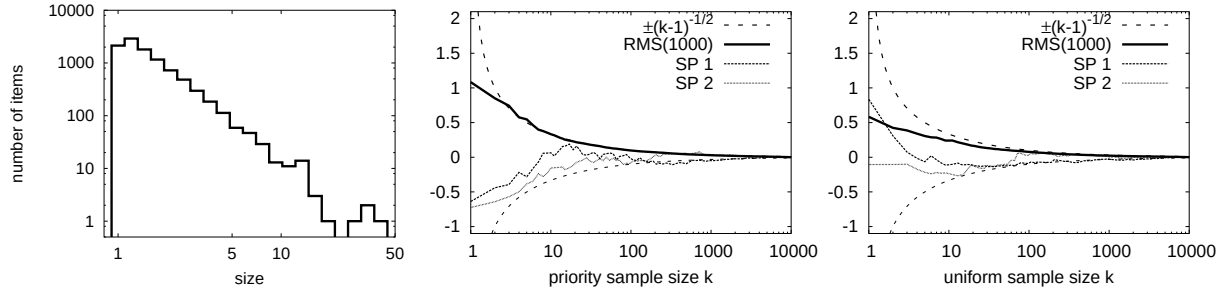

Figure 6: 10,000 Weights from Pareto(2.5) Distribution. Total = 1.66×10^4, Max = 40.9, Mean = 1.66, Median = 1.31, Min = 1.00

7. REFERENCES

[1] N. Alon and J. H. Spencer, *The Probabilistic Method, 2nd ed.*, Wiley, 2000.

[2] L. Arge "External Memory Data Structures" In Handbook of Massive Data Sets, J. Abello, P.M. Pardalos, M.G.C. Resende (Eds.), Kluwer Academic Publishers, 2002, pages 313-357

[3] L. Arge and V. Samoladas and J. S. Vitter, On Two-Dimensional Indexability and Optimal Range Search Indexing, PODS 1999: 346-357

[4] S. Acharya, P.B. Gibbons, V. Poosala, S. Ramaswamy: The Aqua Approximate Query Answering System. SIGMOD Conference 1999: 574-576

[5] R.J Adler, R.E. Feldman and M.S. Taqqu, (Eds.) "A practical guide to heavy tails", Birkhauser, Boston, 1998.

[6] B.C. Arnold and N. Balakrishnan, "Relations, Bounds and Approximations for Order Statistics", Lecture Notes in Statistics, vol. 53, Springer, New York, 1988.

[7] K.R.W. Brewer and M. Hanif, "Sampling With Unequal Probabilities", Lecture Notes in Statistics, vol. 15, Springer, New York, 1983.

[8] E. Cohen, "Size-estimation framework with applications to transitive closure and reachability", *J. Comput. Syst. Sci.*, 55(3):441–453, 1997.

[9] S. Chaudhuri, R. Motwani, V.R. Narasayya, "On Random Sampling over Joins", SIGMOD Conference 1999: 263-274

[10] H.A. David, "Order Statistics", Second Edition, Wiley Series in Probability and Mathematical Statistics, Wiley, New York, 1981

[11] N.G. Duffield, C. Lund, M. Thorup, "Learn more, sample less: control of volume and variance in network measurements", Manuscript, 2002, to appear in *IEEE Trans. on Information Theory.*

[12] N.G. Duffield, C. Lund, M. Thorup, "Flow Sampling Under Hard Resource Constraints", ACM SIGMETRICS 2004, pages 85–96.

[13] P.B. Gibbons, Y. Matias, "New Sampling-Based Summary Statistics for Improving Approximate Query Answers", SIGMOD Conference 1998: 331-342.

[14] C.L. Hays, "What They Know About You", New York Times, November 14, 2004, Section 3, Page 1.

[15] J.M. Hellerstein, P.J. Haas, H.J. Wang, "Online Aggregation", SIGMOD Conference 1997: 171-182

[16] F. Olken, D. Rotem: "Random Sampling from Databases - A Survey" `http://pueblo.lbl.gov/~olken/ sampling.html` March 1994

[17] K. Park, G. Kim, and M. Crovella, "On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic". In Proc. 4th Int. Conf. Network Protocols (ICNP), 1996.

[18] M. Szegedy, "Near optimality of the priority sampling procedure". Electronic Colloquium on Computational Complexity Report TR05-001, 2005.

[19] J.S. Vitter, "Random Sampling with a Reservoir", *ACM Trans. Math. Softw.* 11(1): 37-57 (1985)