Priority sampling for estimation of arbitrary subset sums

NICK DUFFIELD
AT&T Labs–Research
and
CARSTEN LUND
AT&T Labs–Research
and
MIKKEL THORUP
AT&T Labs–Research

From a high volume stream of weighted items, we want to create a generic sample of a certain limited size that we can later use to estimate the total weight of arbitrary subsets. Applied to Internet traffic analysis, the items could be records summarizing the flows of packets streaming by a router. Subsets could be flow records from different time intervals of a worm attack whose signature is later determined. The samples taken in the past thus allow us to trace the history of the attack even though the worm was unknown at the time of sampling.

Estimation from the samples must be accurate even with heavy-tailed distributions where most of the weight is concentrated on a few heavy items. We want the sample to be weight sensitive, giving priority to heavy items. At the same time, we want sampling without replacement in order to avoid selecting heavy items multiple times. To fulfill these requirements we introduce priority sampling, which is the first weight sensitive sampling scheme without replacement that works in a streaming context and is suitable for estimating subset sums. Testing priority sampling on Internet traffic analysis, we found it to perform an order of magnitude better than previous schemes.

Priority sampling is simple to define and implement: we consider a steam of items $i = 0, ..., n-1$ with weights w_i. For each item i, we generate a random number $\alpha_i \in (0, 1]$ and create a priority $q_i = w_i/\alpha_i$. The sample S consists of the k highest priority items. Let τ be the $(k+1)^{th}$ highest priority. Each sampled item i in S gets a weight estimate $\widehat{w}_i = \max\{w_i, \tau\}$, while non-sampled items get weight estimate $\widehat{w}_i = 0$.

Magically, it turns out that the weight estimates are unbiased, that is, $\mathsf{E}[\widehat{w}_i] = w_i$, and by linearity of expectation, we get unbiased estimators over any subset sum simply by adding the sampled weight estimates from the subset. Also, we can estimate the variance of the estimates, and find, surprisingly, that the covariance between estimates $\widehat{w}_i$ and $\widehat{w}_j$ of different weights is zero.

Finally, we conjecture an extremely strong near-optimality; namely that for any weight sequence, there exists no specialized scheme for sampling k items with unbiased weight estimators that gets smaller variance sum than priority sampling with $k+1$ items. Szegedy settled this conjecture at STOC'06.

A very preliminary version of this paper was presented at SIGMETRICS'04 [Duffield et al. 2004]. A more complete technical report has circulated since 2005 [Duffield et al. 2005c].
All authors same address: AT&T Labs—Research, Shannon Laboratory, 180 Park Avenue, NJ 07932, USA. E-Mail: `(duffield,lund,mthorup)@research.att.com`.

Categories and Subject Descriptors: C.2.3 [**Computer–Communications Networks**]: Network Operations—*Network monitoring*; E.1 [**Data**]: Data Structures; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; G.3 [**Mathematics of Computing**]: Probability and Statistics; H.3 [**Information Systems**]: Information Storage and Retrieval

General Terms: Measurement, Theory

Additional Key Words and Phrases: Subset sum estimation, weighted sampling, sampling without replacement, reservoir sampling

1. INTRODUCTION

In this paper we focus on sampling from a high volume stream of weighted items. The items arrive faster and in larger quantities than can be saved, so only a sample can be stored efficiently. We want to create a generic sample of a certain limited size that we can later use to estimate the total weight of arbitrary subsets. This scenario is the basis for data bases over streams [Johnson et al. 2005]. The subset sums can also be used for other aggregates over selected subsets. For example, using unit weights, we can compute subset sizes which together with the previous sums provide the subset averages.

Applied to Internet traffic analysis, the items could be records summarizing the flows streaming by a router. A subset could be flow records of a worm attack whose signature is only determined after sampling has taken place. The samples taken in the past allow us to trace the history of the attack even though the worm was unknown at the time of sampling.

Estimation from the samples must be accurate even with heavy-tailed distributions where most of the weight is concentrated on a few heavy items. We want the sample to be weight sensitive, giving priority to heavy items. At the same time, we want sampling without replacement in order to avoid selecting heavy items multiple times. To fulfill these requirements we introduce priority sampling, which is the first weight sensitive sampling scheme without replacement that works in a streaming context and is suitable for estimating subset sums. Testing priority sampling on Internet traffic analysis, we found it to perform an order of magnitude better than previous schemes.

1.1 Priority Sampling

Priority sampling is a fundamental new technique to sample k items from a stream of weighted items so as to later estimate arbitrary subset sums. The scheme is illustrated in Figure 1. We consider a stream of items with positive weights $w_0, ..., w_{n-1}$. For each item $i = 0, .., n-1$, we generate an independent uniformly random $\alpha_i \in (0, 1]$, and a *priority* $q_i = w_i/\alpha_i$. Assuming that all priorities are distinct, the *priority sample* S *of size* $k < n$ consists of the k items of highest priority. An associated *threshold* τ is the $(k+1)^{\text{th}}$ priority. Then $i \in S \iff q_i > \tau$. Each sampled item $i \in S$ gets a weight estimate $\widehat{w}_i = \max\{w_i, \tau\}$. If $i \notin S$, $\widehat{w}_i = 0$. We will prove

$$\mathsf{E}[\widehat{w}_i] = w_i. \tag{1}$$

By linearity of expectation, if we want to estimate the total weight of an arbitrary subset $I \subseteq [n] = \{0, 1, \ldots, n-1\}$, we just sum the corresponding weight estimates

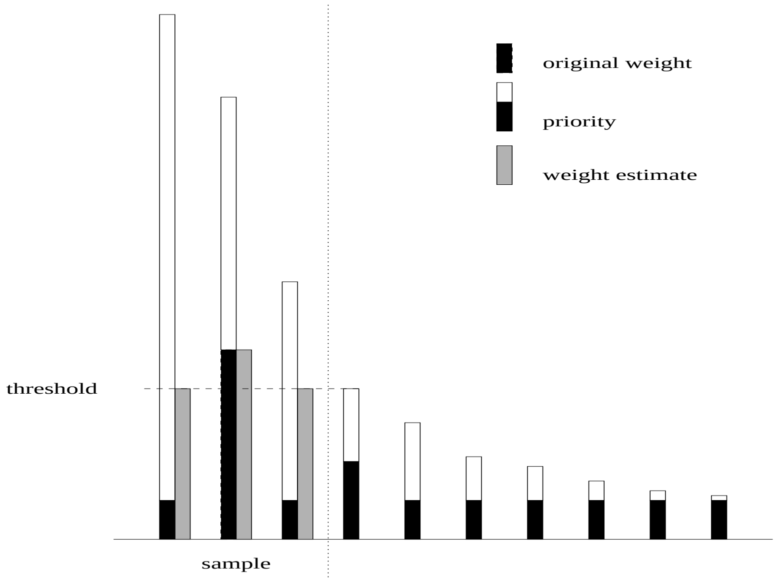


Fig. 1. Priority sampling of size 3 from a set of 10 weighted items.

in the sample, that is,

$$\mathsf{E}\left[\sum_{i\in S\cap I}\widehat{w}_i\right]=\mathsf{E}\left[\sum_{i\in I}\widehat{w}_i\right]=\sum_{i\in I}w_i. \tag{2}$$

Ties between priorities happen with probability zero, and can be resolved arbitrarily. We resolve them in favor of earlier items. Thus we view priority q_i as higher than q_j, denoted $q_i \succ q_j$, if either $q_i > q_j$ or $q_i = q_j$ and $i < j$. With any such resolution of ties, priority sampling works even if some weights are zero.

Note that in the case of unit weights, τ is just the $(k+1)^{\text{th}}$ largest value $1/\alpha_i$, and then (2) simplifies to

$$\mathsf{E}\left[k\tau\right]=n. \tag{3}$$

This unit case is a classic theorem in order statistics (see e.g., [Arnold and Balakrishnan 1988; David 1981]).

1.2 Reservoir sampling

Priority sampling is ideally suited for reservoir sampling (c.f. [Knuth 1969, p. 138–140]). In reservoir sampling, we operate on a stream of (weighted) items. The items arrive one at the time, and a reservoir maintains a sample S of the items seen thus far. When a new item arrives, it may be included in the sample S and old samples may be dropped from S. Old items outside S are never reconsidered. Reservoir sampling addresses two issues:

—The streaming issue [Muthukrishnan 2005] where we with limited memory want to compute a sample from a huge stream that passes by only once.

—The incremental data structure issue of maintaining a sample as new items are added. In our case, we use the sample to provide quick estimates of sums over arbitrary subsets of the items seen thus far.

In priority reservoir sampling, we simply use a standard priority queue to maintain the $k+1$ highest priority items. When an item i arrives, we generate the random $\alpha_i \in (0,1]$, and assign it the priority $q_i = w_i/\alpha_i$. The first $k+1$ items are all put in the queue. Each subsequent item is also put in the queue but afterward we remove the smallest item from the queue. This takes $O(\log k)$ time per item using a comparison based priority queue [Cormen et al. 2001, pp. 138–141] or $O(\log\log k)$ time if we exploit that the priorities are represented as standard double precision floating point numbers.

1.3 Relation to survey sampling

The above set-up is similar to that of classic survey sampling (see, e.g. [Särndal et al. 1992]). However, in survey sampling, typically, we do not know the weight w_i of an item i unless we sample it. We may have access to some auxiliary variable u_i that is somehow correlated with w_i, and we may use u_i to determine the sampling probability p_i for item i. For example, if the items i are households, the w_i may be income while the u_i is an approximation of w_i based on street address. A main challenge is to estimate the total weight based on the sampled weights.

Our context is that a large stream of weighted items pass by. When item i passes by, we get to see its weight w_i. If our goal was to compute the total, we would simply accumulate the weights in a counter. Hence, in our context, the challenge of survey sampling is trivial.

One thing that makes reservoir sampling hard is that sampling decisions are made on-line. This rules out off-line sampling schemes such as Sunter's [1977] method where we have to sort all the items before any sampling decision is made.

A cultural difference between survey sampling and our case is that survey sampling appears less focused on heavy tailed distributions. In our kind of applications, heavy tailed distributions are very prominent [Adler et al. 1998; Park et al. 1996].

Finally, we note that priority sampling can be used in the above survey sampling context, where the sampling decision is based on a weight u_i different from the weight w_i that we are really interested in. The standard observation is that priority sampling provides an unbiased estimator $\widehat{u}_i$ of u_i, and then we can use $\widehat{w}_i = \widehat{u}_i w_i/u_i$ as an unbiased estimator of w_i. However, in the rest of the paper, we are focused on the case where the weight of interest is known prior to the sampling decision.

1.4 Concrete examples

With two examples we will now illustrate the selection of subsets and the use of reservoir sampling for estimating the sum over these subsets. For the selection, the basic point is that an item, besides the weight, has other associated information, and selection of an item may be based on all its associated information. As stated in (2), to estimate the total weight of all selected items, we sum the weight estimates of all selected sampled items.

1.4.1 *Internet traffic analysis.* As mentioned earlier, our motivating application comes from Internet traffic analysis. Internet routers export information about flows

	Flow record 1	Flow record 2
src IP	192.20.225.32	10.255.255.255
src port	**80** (http)	48181
dest IP	10.0.0.1	192.168.1.1
dest port	48811	**1434**
protocol	**17** (tcp)	**6** (udp)
bytes (weight)	13408	**404**
packets	10	1

Fig. 2. Flow record examples. Record 1 is response from a web server 192.20.225.32. Record 2 is the slammer worm in action. Infected host 10.255.255.255 trying to infect 192.168.1.1 (all addresses here are hypothetical)

of packets passing through them. A flow could be transmissions an ftp transfer of a file, an email, or some other collection of related packets. The router summarizes each flow in a flow record which is exported to a collector for subsequent analysis. Each flow record contains the flow's key, i.e., the common properties that relate the flow's packets, such as their source and destination IP addresses and TCP/UDP port numbers and protocol number, and a summary of the packets in the flow, e.g., total number of packets and bytes. We think of the byte size as the weight. Some examples of flow records are presented in Figure 2.

We want to sample flow records in such a way that we can answer questions like how many bytes of traffic came from a given customer or how much traffic was generated by a certain application. Both of these questions ask what is the total weight of a certain selection of flows. If we knew in advance of measurement which selections were of interest, we could have a counter for each selection and increment these as flows passed by. The challenge here is that we must not be constrained to selections known in advance of the measurements. This would preclude exploratory studies, and would not allow a change in routine questions to be applied retroactively to the measurements.

A striking example where the selection is not known in advance was the tracing of the *Internet Slammer Worm* [Moore et al. 2003]. It turned out to have a simple signature in the flow record; namely as being udp traffic to port 1434 with a packet size of 404 bytes. Once this signature was identified, the worm could be studied by selecting records of flows matching this signature from the sampled flow records.

We note that data streaming algorithms have been developed that generalizes counters to provide answers to a range of selections such as, for example, range queries in a few dimensions [Muthukrishnan 2005]. However, each such method is still restricted to a limited type of selection to be decided in advance of the measurements.

1.4.2 *External information in the selection.* In our next example, suppose a large chain store saved samples of all their sales where each record contained information such as item, location, time, and price. The weight of a record is the price.

Based on sampled records, the store management might want to 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 allow them to tell 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 important lesson from this example is that selection can be based on external information not even imagined relevant at the time when samples are made. Such scenarios preclude any kind of streaming algorithm based on selections of limited complexity, and shows the inherent relevance of sampling preserving full records for the purpose of arbitrary selections.

To put this example in a reservoir sampling context, imagine a central reservoir maintaining a priority sample over all sales done so far. The reservoir is small enough that it can be shared easily over the Internet with an analyst at a different location.

1.5 Relation to known sampling schemes

We will now relate priority sampling to some known sampling schemes. We will discuss the schemes in the context of reservoir sampling from a stream. We have resources for a reservoir of $k = \omega(1)$ samples. We are particularly interested in the important case of heavy-tailed distributions where one or a few dominant items contain most of the total weight [Adler et al. 1998; Park et al. 1996].

Below we present the schemes and make some basic comparisons. In the body of the paper we shall compare the schemes both experimentally and theoretically.

First we have the standard schemes: uniform sampling without replacement and probability proportional to size sampling with replacement:

Uniform sampling without replacement (U−R). In uniform sampling without replacement, we pick a sample of k items uniformly at random. If item i is sampled it gets weight estimate $\widehat{w}_i = w_i n/k$. We denote this scheme U-R$_k$.

Uniform sampling has problems with heavy-tailed distribution because it is likely to miss the dominant items.

Probability proportional to size sampling with replacement (P+R). In probability proportional to size sampling with replacement, each sample $S_j \in [n]$, $j \in [k]$, is independent, and equal to i with probability w_i/W where W is the sum of all weights. Then i is sampled if $i = S_j$ for some $j \in [k]$. This happens with probability $p_i = 1-(1-w_i/W)^k$, and if i is sampled, $\widehat{w}_i = 1/p_i$. We denote this scheme P+R$_k$.

Probability proportional to size sampling with replacement also has a problem with heavy-tailed distributions for if one or a few dominant items contain most of the total weight, then most samples will be copies of these dominant items. As a result, we are left with comparatively few samples of the remaining items.

We note that there are many variations of probability proportional to size sampling with replacement. For example, an alternative estimator for item i is $\widehat{w}_i = w_i/W \cdot |\{j \in [k]|S_j = i\}|$. However, this alternative estimator has larger variance and is hence less popular. In the case of integer weights, another variant is to *divide them into unit weights* and then use uniform sampling without replacement on these units. The estimate of an item is then the sum of the estimates of its units. However, when the total weight is large compared with the number of samples, then this alternative is very similar to the previous one. None of these alternatives alleviate the basic problem that most samples are devoted to a few dominant items.

Skipping the duplicates. A simple fix for probability proportional to size sampling with replacement seems to be to skip duplicates until we have the desired number of distinct samples. The problem is that the probability that a given item is included in the sample is a complicated function of all the involved weights.

The above discussion suggests that we need *weight-sensitive sampling without replacement.* Below we discuss some schemes that do just that, but which unfortunately have problems sampling k items from a stream:

Threshold sampling (THR). The threshold sampling from [Duffield et al. 2005a] is a kind of Poisson sampling. In Poisson sampling, each item i is picked independently for S with some probability p_i. For unbiased estimation, we set $\widehat{w}_i = w_i/p_i$. The expected number of samples is $\mathsf{E}[|S|] = \sum_i p_i$.

In threshold sampling we pick a fixed threshold τ. For the sample S, we include all items with weight bigger than τ. Moreover, we include all smaller items with probability w_i/τ. If τ is bigger than the largest weight, this is sampling with probability proportional to size. Sampled items $i \in S$ have estimate $\widehat{w}_i = w_i/p_i = w_i/\min\{1, w_i/\tau\} = \max\{w_i, \tau\}$. With $k = \sum_i \min\{1, w_i/\tau\}$ the expected number of samples, we denote this scheme THR_k. Threshold sampling is an optimal Poisson sampling scheme in that it minimizes the sum of individual variances relative to the expected number of samples.

In our streaming context, we have the immediate problem that we do not know in advance which threshold τ to use so that the expected number of samples is k. In [Duffield et al. 2005a] it is suggested to update τ periodically, but a burst in traffic would still lead to a burst in samples. As a result of some independent interest, we will show how τ can be increased on-line as items arrive, dropping some previously sampled items, so as to maintain a threshold reservoir with an expected number of k samples.

However, in this paper, we assume that we have allocated resources only for a fixed number of samples, so we would have to go for a much smaller expected number to be sure to stay within the bound.

The systematic threshold sampling below addresses the problem of getting exactly k samples, but then it will run into other problems.

Systematic threshold sampling (SYS). We consider the general version of systematic sampling where each item i has an individual sampling probability p_i, and if picked, a weight estimate w_i/p_i. Contrasting Poisson sampling, the sampling decisions are not independent. Instead we pick a single uniformly random number $x \in [0, 1]$, and include i in S if and only if for some integer j, we have

$$\sum_{h<i} p_i \leq j + x < \sum_{h \leq i} p_i.$$

It is not hard to see that $\Pr[i \in S] = p_i$. Let $k = \sum_{i \in [n]} p_i$ be the expected number of samples. Then the actual number of samples is either $\lfloor k \rfloor$ or $\lceil k \rceil$. In particular, this number is fixed if k is an integer. Below we assume that k is integer.

In systematic threshold sampling, we will assume exactly the same sampling probabilities as in threshold sampling, and denote this scheme SYS_k. Hence for each item i we have identical marginal distributions $\widehat{w}_i$ with THR_k and SYS_k.

Thus systematic threshold sampling can be seen as fixing the variable number of samples in threshold sampling, thus fitting it to our fixed resource constraint on the samples. Unfortunately, it is not so easy to adapt the threshold τ on-line. The problem is that a small change in threshold completely changes the set of items sampled, unless the change is by a factor in which case we essentially drop every other item. Another objection to systematic threshold sampling in a streaming context is that we may have a very strong correlations between items in a subset depending on how they are placed in the stream. Hence some subsets may be subject to a huge variance.

Thus, even if threshold and systematic threshold sampling adapt well to different weight distributions, we have the problem that they are not suitable for sampling at most k items in a streaming context. We will thus view threshold and systematic threshold sampling as idealistic benchmarks.

Other schemes. The book [Brewer and Hanif 1983] mentions many other sampling schemes, all aiming at the same marginal sampling probabilities as the schemes described above. The ones discussed here are those closest to being implementable in an on-line streaming context.

Priority sampling (PRI). We basically claim that our new priority sampling simultaneously solves all our problems. We already saw that it was simple to maintain a reservoir of k priority samples using a standard priority queue for the $k+1$ items of highest priority. It is also trivial that priority sampling acts without replacement. To see that it is weight-sensitive, suppose we have an item i which is $r = w_j/w_i \geq 1$ times smaller than an item j. Then the probability that i gets higher priority than j is $1/2r$. More precisely,

$$\Pr[q_i > q_j] = \Pr[w_i/\alpha_i > w_j/\alpha_j] = \Pr[\alpha_i < \alpha_j/r] = \int_0^1 \alpha_j/r \, d\alpha_j = 1/2r. \tag{4}$$

Priority sampling is thus weight-sensitive without replacement, and, as stated in (2), it does provide unbiased estimates.

Priority based implementation of threshold sampling. Next we will demonstrate that priority sampling, technically speaking, is very similar to threshold sampling. First we show how to implement threshold sampling using priorities. For each item i, we generate a uniformly random $\alpha_i \in (0, 1]$, and assign i the priority $q_i = w_i/\alpha_i$. We sample i if q_i is bigger than the fixed threshold τ^{THR}. This way we sample i with probability $\min\{1, w_i/\tau^{THR}\}$ as required for threshold sampling. If sampled, the estimate is $\max\{w_i, \tau^{THR}\}$.

In priority sampling, the only technical difference is that we instead of using the fixed threshold τ^{THR} use the $(k+1)$th priority as a variable threshold τ^{PRI}.

Recall that the threshold sampling threshold τ^{THR} is the fixed value such that the expected number of larger priorities is k. The priority sampling threshold τ^{PRI} is the priority which has exactly k larger priorities. Thanks to this similarity, we would hope that priority sampling in its estimate quality would perform similarly to threshold sampling.

Despite the technical similarity between priority and threshold sampling, we note that priority sampling is much harder to analyze. In threshold sampling, sampling decisions are independent for different items. However, in priority sampling, the sampling of an item depends on τ^{PRI} which, as the $(k+1)$th priority, depends on all the random priorities. Just proving that unbiasedness as stated in (2) is non-obvious.

Priority sampling is special.. The use of the variable τ^{PRI} in the weight estimate $\max\{w_i, \tau^{PRI}\}$ sets priority sampling apart from all the other sampling schemes considered. These other sampling schemes use a fixed value if an item is sampled. More precisely, if item i is sampled with probability p_i, they use the Horwitz-Thompson estimator $1/p_i$ which is the fixed value leading to unbiased estimation.

Finally, we note that priority sampling has different sampling probabilities. As a simple example, suppose we sample one of two items with weights w_1 and w_2 where $w_1 < w_2$. In priority sampling, we sample the item of highest priority. According to (4), the probability that w_1 is picked is $w_1/(2w_2)$. The other weight-sensitive schemes P+R$_k$, THR$_k$, and SYS$_k$, would all pick w_1 with probability proportional to size, that is, with probability $w_1/(w_1+w_2) > w_1/(2w_2)$.

1.6 The birth of priority sampling.

As a historical remark, we note that the way we came up with priority sampling was that we first came up with the priority based implementation of threshold sampling. This was needed for a reservoir implementation of independent interest. Then, as a simple hack to get exactly k samples, we replaced τ^{THR} with the $(k+1)$th priority as a variable threshold τ^{PRI}. Analyzing this new sampling scheme, we discovered to our great surprise that it provided unbiased estimators and had many other interesting properties. This paper is only the start, introducing priority sampling along with some of its basic properties. There have already been follow-up papers by ourselves and others [Cohen et al. 2007; Cohen and Kaplan 2007; Duffield et al. 2005b; Szegedy 2006; Szegedy and Thorup 2007; Thorup 2006], deepening the theory and broadening the practice. We will describe these directions in the concluding remarks.

1.7 Outline of the Paper

The rest of the paper is organized as follows. In Section 2 we present a basic theory for priority sampling. First of all, we show that priority sampling provides unbiased estimators as stated in (1). In addition we will show how we can estimate the variance of our subset sum estimates. This relies on a striking property of priority sampling, namely, that the covariance between different weight estimates is zero. Zero covariance is impossible for any kind of Horvitz-Thompson estimator using exactly k samples.

In Section 3, we present experiments comparing priority sampling with the other sampling schemes from Section 1.5 on real data from the Internet. We demonstrate an order of magnitude gain in accuracy in subset sum estimates compared with uniform sampling and probability proportional to size sampling with replacement. We also present experiments on synthetic heavy-tailed distributions

In Section 4, we analyze the performance of the different sampling schemes in some simple cases in order to gain further understanding of the experiments.

In Section 5 we discuss the optimality of priority sampling in relation to that of threshold and systematic threshold sampling. We conjecture an extremely strong near-optimality; namely that for any weight sequence, there exists no specialized scheme for sampling k items with unbiased weight estimators that gets smaller sum of individual variances than priority sampling with $k+1$ items. This conjecture was settled recently by Szegedy [2006][1].

In Section 6, we compare in detail how reservoir sampling can be implemented with the different schemes. In particular this includes our reservoir implementation of threshold sampling.

In Section 7 we finish with some concluding remarks including a description of later follow-up works.

2. BASICS OF PRIORITY SAMPLING

In this section, we will show that priority sampling yields unbiased estimates of subset sums as stated in (1). The proof is simpler and more combinatorial than the standard proofs for the known unit case [Arnold and Balakrishnan 1988; David 1981]. We will also show how to form unbiased estimators of secondary weights. Finally, we consider variance estimation. We show that there is no covariance between the weight estimates of different items, and that we can get unbiased estimates of the variance of any subset sum estimate.

Recall that we consider items with positive weights $w_0, ..., w_{n-1}$. For each item $i \in [n]$, we generate an independent uniformly distributed random number $\alpha_i \in (0,1]$, and a priority $q_i = w_i/\alpha_i$. Priority q_i is higher than q_j, denoted $q_i \succ q_j$, if either $q_i > q_j$, or $q_i = q_j$ and $i < j$. A priority sample S of size k consists of the k items of highest priority. The threshold τ is the $(k+1)^{\text{st}}$ highest priority. Then $i \in S \iff q_i \succ \tau$. Each $i \in S$ gets a weight estimate $\widehat{w}_i = \max\{w_i, \tau\}$. Also, for $i \notin S$, we define $\widehat{w}_i = 0$. Now (1) states that $\mathsf{E}[\widehat{w}_i] = w_i$.

We will prove that (1) holds for an item i no matter which values the other $\alpha_j,\ j \neq i$ take. Fixing these values, we fix all the other priorities $q_j,\ j \neq i$. Let τ' be the k^{th} highest of these other priorities. We can now view τ' as a fixed number. More formally, our analysis is conditioned on the event $A(\tau')$ of τ' being the k^{th} highest among the priorities $q_j,\ j \neq i$, and we will prove

$$\mathsf{E}[\widehat{w}_i | A(\tau')] = w_i. \tag{5}$$

Proving (5) for any value of τ' implies (1). The essential observation is as follows.

LEMMA 2.1. *Conditioned on $A(\tau')$, item i is picked with probability $\min\{1, w_i/\tau'\}$, and if picked, $\tau = \tau'$.*

[1] It may seem funny to raise here a conjecture that has already been settled, but our conjecture goes back to SIGMETRICS'04 when this work was first announced and the conjecture was included in [Duffield et al. 2005c]. Moreover, we show here that the conjecture is equivalent to a certain technical statement, and it is this technical statement that Szegedy proved true.

PROOF. We pick $\alpha_i \in (0,1]$ uniformly at random, thus fixing $q_i = w_i/\alpha_i$. If $q_i \prec \tau'$, there are at least k priorities higher than q_i, so $i \notin S$. Conversely, if $q_i \succ \tau'$, then τ' becomes the $(k+1)$th priority among all priorities, so $\tau' = \tau$, and then $i \in S$. Finally,

$$\Pr[i \in S | A(\tau')] = \Pr[q_i \succ \tau'] = \Pr[\alpha_i < w_i/\tau'] = \min\{1, w_i/\tau'\}.$$

□

From Lemma 2.1, we get

$$\begin{aligned} \mathsf{E}[\widehat{w}_i | A(\tau')] &= \Pr[i \in S | A(\tau')] \times \mathsf{E}\left[\widehat{w}_i | i \in S \wedge A(\tau')\right] \\ &= \min\{1, w_i/\tau'\} \times \max\{w_i, \tau'\} \\ &= w_i. \end{aligned}$$

The last equality follows by observing that both the min and the max take their first, respectively their second value, depending on whether or not $w_i \geq \tau'$. This completes the proof of (5), hence of (1).

2.1 Zero weight items and sampling it all

We note here that priority sampling, as defined above, works even in the presence of zero weights. First we note that $w_i = 0 \iff q_i = w_i/\alpha_i = 0$ while $w_i > 0 \iff q_i = w_i/\alpha_i > w_i > 0$. It follows that zero weight items can only be sampled if all positive weight items have been sampled. Moreover, if we do sample a zero weight item i, we have $\tau \prec q_i = w_i = 0$, so $\tau = 0$, and then $\widehat{w}_j = w_j$ for all items j. Having noted that zero weight items do not cause problems, we will mostly ignore them.

Above we have assumed $k < n$, but we note a natural view of a priority sample of everything, that is, with $k = n$. We define an $(n+1)^{th}$ priority $\tau = q_n = 0$, as if we had an extra zero weight $w_n = 0$. Then $q_i \succ \tau = q_n$ for all $i \in [n]$, so all items get sampled. Moreover $\widehat{w}_i = \max\{w_i, \tau\} = w_i$, so the weight estimate is equal to the original weight.

2.2 Secondary variables

Suppose that each item i has a secondary variable x_i. We can then use (1) to give unbiased estimators of corresponding secondary subset sums. More precisely, we set $\widehat{x}_i = \widehat{w}_i x_i / w_i$. That is $\widehat{x}_i = \max\{w_i, \tau\} x_i / w_i = \max\{1, \tau/w_i\} x_i$ if i is sampled; 0 otherwise. Then (1) implies $\mathsf{E}[\widehat{x}_i] = x_i$.

An application could be to deal with negative and positive weights x_i. We could define the priority weights as their absolute values, that is, $w_i = |x_i|$, and use these non-negative weights in the priority sample.

Another application could be if we had several different variables for each item. Instead of making an independent priority sample for each variable, we could construct a compromise weight. For example, for each item, the weight could be a weighted sum of all the associated variables.

2.3 Variance estimation for a single item

We now provide a simple variance estimator

$$\widehat{v}_i = \begin{cases} \tau \max\{0, \tau - w_i\} & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases},$$

and show that it is unbiased, that is,

$$\mathsf{E}\,[\widehat{v}_i] = \mathsf{Var}[\widehat{w}_i]. \tag{6}$$

As in the proof of (1), we define $A(\tau')$ to be the event that τ' is the k^{th} highest among the priorities $q_j,\ j \neq i$. We will prove

$$\mathsf{E}\,[\widehat{v}_i | A(\tau')] = \mathsf{E}[\widehat{w}_i^2 | A(\tau')] - w_i^2. \tag{7}$$

From Lemma 2.1, we get

$$\begin{aligned} \mathsf{E}[\widehat{v}_i | A(\tau')] &= \Pr[i \in S | A(\tau')] \times \mathsf{E}\,[\widehat{v}_i | i \in S \wedge A(\tau')] \\ &= \min\{1, w_i/\tau'\} \times \tau' \max\{0, \tau' - w_i\} \\ &= \max\{0, w_i\tau' - w_i^2\}. \end{aligned}$$

On the other hand,

$$\begin{aligned} \mathsf{E}[\widehat{w}_i^2 | A(\tau')] &= \Pr[i \in S | A(\tau')] \times \mathsf{E}\left[\widehat{w}_i^2 | i \in S \wedge A(\tau')\right] \\ &= \min\{1, w_i/\tau'\} \times \max\{w_i, \tau'\}^2 \\ &= \max\{w_i^2, w_i\tau'\}. \end{aligned}$$

This establishes (7) and hence (6).

2.4 Zero covariance

Assuming that we sample more than one item, we will show that the covariance between our weight estimates is zero , that is, for $k > 1$ and $i \neq j$,

$$\mathsf{E}\,[\widehat{w}_i \widehat{w}_j] = w_i w_j. \tag{8}$$

If $k = 1$, we have $\mathsf{E}\,[\widehat{w}_i \widehat{w}_j] = 0$ since we cannot sample both i and j.

Note that (8) is somewhat counter-intuitive in that if we sample i then this reduces the probability that we also sample j. However, the assumption that i is sampled affects the threshold τ and thereby the weight estimate $\widehat{w}_j$ and it turns out that the different effects cancel out.

We will prove (8) via the following common generalization of (8) and (1) holding for any $I \subset [n],\ |I| \leq k$:

$$\mathsf{E}\left[\prod_{i \in I} \widehat{w}_i\right] = \prod_{i \in I} w_i. \tag{9}$$

If $|I| > k$, we have $\mathsf{E}\left[\prod_{i \in I} \widehat{w}_i\right] = 0$ since at most k items are sampled with $\widehat{w}_i > 0$.

The proof of (9) generalizes that of (1). Inductively on the size of I, we will prove that (9) holds no matter what values all the other $\alpha_j,\ j \notin I$ take. The equality is trivially true in the base case where $I = \emptyset$ and the products equals one.

Thus, for all $j \notin I$, fix all $\alpha_j \in (0, 1]$ and priorities $q_j = w_j/\alpha_j$. Fix τ'' to be the $(k - |I| + 1)^{\text{th}}$ highest of these priorities $q_j\ j \notin I$. This priority exists because $k \leq |I| < n$. Next for $i \in I$, we pick $\alpha_i \in (0, 1]$ and set $q_i = w_i/\alpha_i$. We can now have at most $(k - |I|) + |I|$ priorities below τ'', so τ'' is at least as big as our new threshold τ.

Consider the case that I has a weight $w_h \geq \tau''$. Fix $\alpha_h \in (0,1]$ arbitrarily. Then $q_h > w_j \geq \tau'' \geq \tau$, so item h is sampled with $\widehat{w}_h = \max\{w_h, \tau\} = w_h$. Hence $\mathsf{E}\left[\prod_{i\in I}\widehat{w}_i\right] = w_h \mathsf{E}\left[\prod_{i\in I\setminus\{h\}}\widehat{w}_i\right]$. We have now fixed all α_j, $j \notin I \setminus \{h\}$, and by induction, $\mathsf{E}\left[\prod_{i\in I\setminus\{h\}}\widehat{w}_i\right] = \prod_{i\in I\setminus\{h\}} w_i$. This completes the proof of (9) in the case that I has some weight as big as τ''.

Next consider the case that all weights from I are smaller than τ''. Let q_ℓ be the lowest priority from I. If $q_\ell \prec \tau''$, then there are at least $(k-|I|+1)+|I\setminus\{\ell\}| = k$ priorities higher than q_ℓ, so $q_\ell \notin S$, and $\widehat{w}_\ell = 0 = \prod_{i\in I}\widehat{w}_i$. Thus, if $q_\ell \prec \tau''$, there is no contribution to $\mathsf{E}\left[\prod_{i\in I}\widehat{w}_i\right]$.

Conversely, if $\tau'' \prec q_\ell$, then all priorities from I are bigger than τ''. In this case there are exactly $(k-|I|)+|I| = k$ priorities higher than τ'', so τ'' becomes our threshold τ. Then each $i \in S$ are sampled. Since $w_i \leq \tau'' = \tau$, we get $\widehat{w}_i = \max\{w_i, \tau\} = \tau$. Hence $\prod_{i\in I}\widehat{w}_i = \tau''^{|I|}$. Since no weights in I is higher than τ'', the probability that all their priorities are bigger is $\prod_{i\in I}(w_i/\tau'')$. Thus, the contribution to $\mathsf{E}\left[\prod_{i\in I}\widehat{w}_i\right]$ is $\tau''^{|I|}\prod_{i\in I}(w_i/\tau'') = \prod_{i\in I} w_i$. This completes the proof of (9) in the remaining case where all weights from I are smaller than τ''.

2.5 Variance estimation over subsets

We can now use our variance estimator from Section 2.3 to estimate the variance over any subset. By (8) and (6) we get an unbiased estimator of the variance of any subset sum estimate simply by summing the variance estimators from the subset, that is, if $k > 1$ for any subset $I \subseteq [n]$,

$$\mathsf{Var}[\sum_{i\in I}\widehat{w}_i] = \sum_{i\in I}\mathsf{Var}[\widehat{w}_i] = \mathsf{E}\left[\sum_{i\in S\cap I}\widehat{v}_i\right]. \tag{10}$$

In fact, (10) also holds if $k = 1$, but this is because $\mathsf{Var}[\sum_{i\in I}\widehat{w}_i] = \infty$ for any non-empty subset I. We shall return to this point later in Section 4.1.

3. EXPERIMENTS

We tested priority sampling against the other sampling schemes from Section 1.5 on both real and synthetic data.

3.1 Internet flow records

The real data are from the Internet application we discussed in Section 1.4. The stream consisted of 85,680 flow records exported from an Internet gateway router. The distribution of bytes reported in the flow records was quite heavy-tailed with a single record containing 78% of the total weight. Such heavy tails are common in real life [Adler et al. 1998; Park et al. 1996].

In Figure 3 we consider subsets defined by different types of applications; namely ftp, multimedia, sessions, and dns. On the left hand side the weights of the sets are plotted in order of decreasing weight. The interesting thing is that the subsets are of very different nature. For example, there are less than 1% ftp flows, yet they represent most of the weight. Conversely, nearly half the flows are dns, yet they represent less than 0.1% of the total weight.

The middle plot shows the estimates for increasing sample sizes k. We note that for threshold sampling, k represents the expected number of samples and not the actual number. Also, with weighted sampling with replacement, there may be many duplicates among the k samples. In fact the percentage of duplicates is plotted in Figure 4. For all sampling schemes except systematic threshold sampling, increasing k means adding more samples to the same set, that is, the $k+1$ samples contains the k samples. However, there is no natural way of doing this with systematic sampling. This is why it is only for systematic threshold sampling that we can have an isolated point as we see it in the multimedia case for $k \approx 15$.

Finally, on the right hand side we plotted the root mean square (RMS) relative error over 1000 runs. That is, we made a 1000 runs with increasing sample sizes, each like the one in the middle plot. For each sampling scheme and each value of k, we got 1000 estimates $\widehat{W}_I$ of the weight W_I of a subset. The relative error is $\frac{|\widehat{W}_I - W_I|}{W_I}$, and we computed the square root of the mean over all 1000 estimates of the square of the relative error. The RMS relative error is hence our measured relative standard deviation.

In Figure 5 we consider estimates of all entries in an 8×8 traffic matrix. Each entry then represents the subset of traffic going from a given source to a different destination. Whereas Figure 3 considered the performance on very different types of subsets, the point here is to consider the average performance over the 64 entries, thinking of these as more random subsets. What is plotted is the sum of errors over all entries divided by the total traffic. These data are all based on a single run, that is, for each sampling scheme and value of k, it is the same sample that is used to estimate all 64 entries.

3.2 Pareto distributions

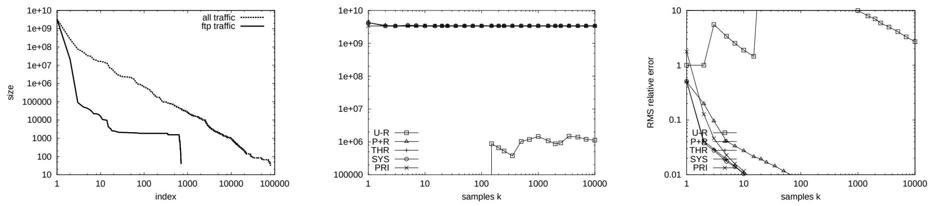
As synthetic data we use Pareto distributions displaying different degrees of heavy-tailedness. The Pareto(β) distribution of a random variable X has cumulative distribution function $P[X < x] = 1 - x^{-\beta}$ for x greater than 1. The distribution is more heavy tailed when β is small. It has infinite variance for $\beta \le 2$ and infinite mean for $\beta \le 1$.

In order to study the behavior of the schemes as the distribution got more and more heavy-tailed, we started with a ground set of 10,000 weights each picked uniformly at random from $[0, 1]$. Each weight was selected independently with probability 1/10 for a subset. Next, for $\beta = 1.75, 1.25, 0.75$, we converted the data to Pareto(β) with the map $w_i \mapsto (1 - w_i)^{1/\beta}$. This way it is only the distribution of the ground set and subset that changes while the ranks of the subset elements within the ground set remain unchanged. The results are presented in Figure 6. After uniform, we display the Paretos in order of decreasing β so they become less and less uniform and more and more heavy-tailed. As in Figure 3, we have 3 plots for each case, first a distribution plot, then the result of a single run, and finally the RMS relative error measured over 1000 runs.

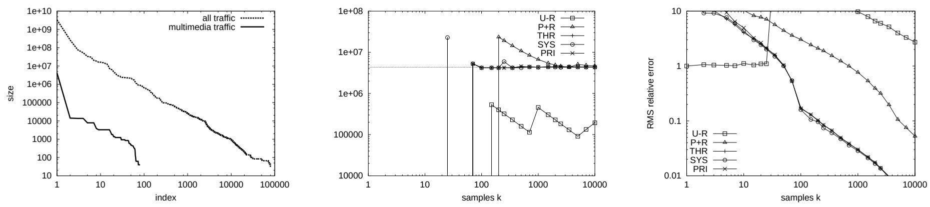
3.3 Discussion

The quality of a sampling scheme is the number of samples it takes before the estimates converge towards the true value. It takes only a quick look at the results

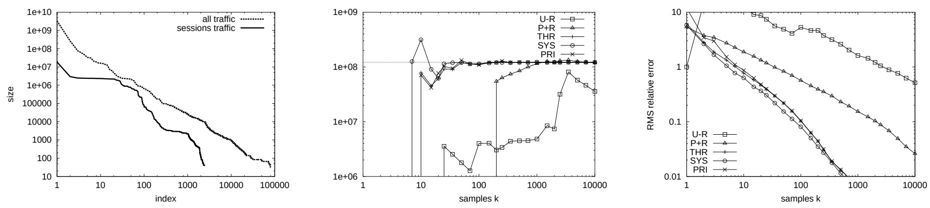
Ftp. This subset has 727 items and weight 3,394,832,734



Multimedia. This subset has 81 items and weight of 4,339,064.



Sessions. This subset has 2,600 items and weight of 121,625,397.



Dns. This subset has 40,767 items and weight 4,083,277.

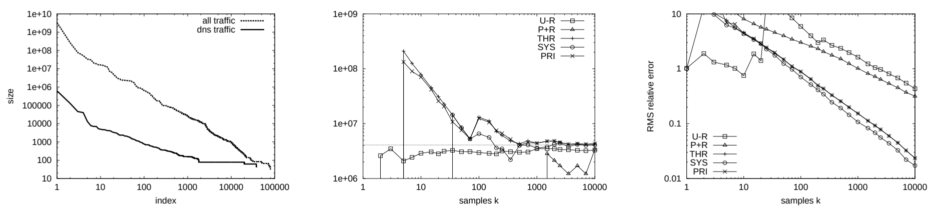


Fig. 3. Real network data. A stream 85,680 flow records was collected from an Internet gateway router. Their total weight was 4,265,677,642. Different traffic classes were selected as subsets. Left: Weight distribution for total traffic and dns traffic. Middle: Weight estimates as a function of sample size. Right: RMS relative error of weight estimates as function of sample size averaged over 1000 runs.

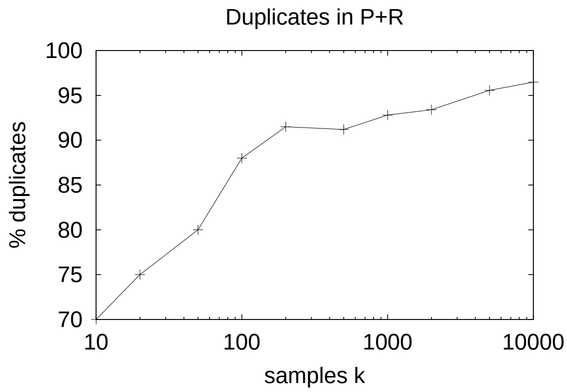


Fig. 4. Percentage of duplicates with P+R as function of number of samples.

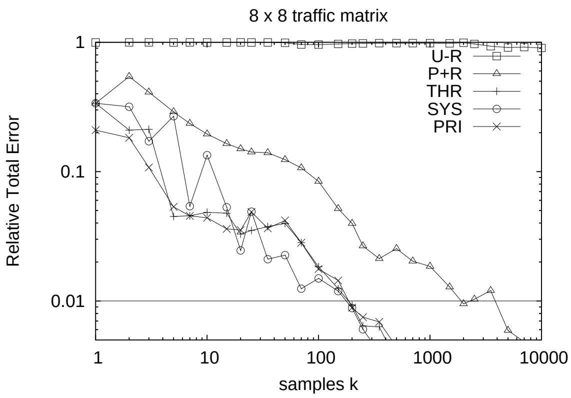
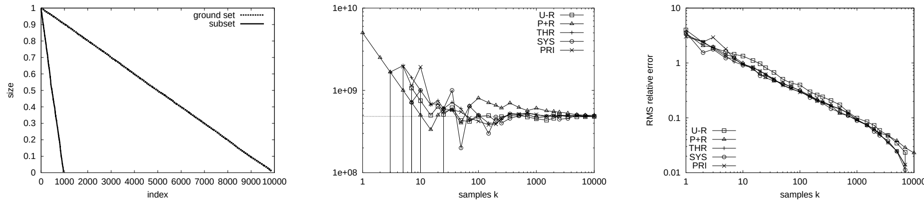


Fig. 5. Estimating a traffic matrix with different sampling strategies. We divide the total error over all entries with the total traffic.

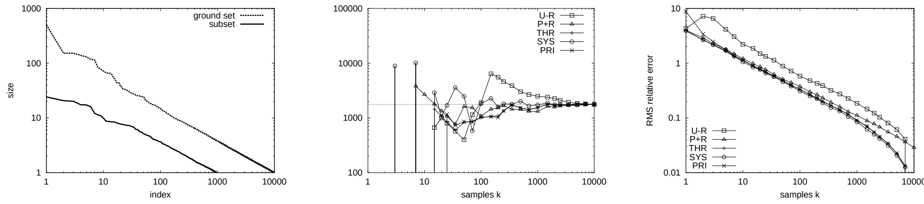
in Figures 3, 5, and 6 to see that priority sampling (PRI) performs better than the standard alternatives: uniform sampling (U−R) and probability proportional to size sampling with replacement (P+R). Also, we see that its performance is very similar to that of the idealistic benchmarks: thresholds sampling (THR) and systematic threshold sampling (SYS). These two schemes are considered idealistic in our context because they are not suitable in a streaming context for maintaining a reservoir of k samples (c.f. Section 1.2). The similarity between PRI, THR, and SYS is so strong that it is often hard to tell them apart in the plots. Below follows a more detailed comparison.

We will now in more detail compare priority sampling (PRI) scheme with the standard alternatives uniform sampling without replacement (U−R) and probability proportional to size sampling with replacement (P+R). First we focus on the single sample paths in the middle plots of Figures 3 and 6, showing single estimates with increasing sample sizes. For each of the three schemes, we are adding samples to the same set when k increases. When comparing the resulting curves, there are

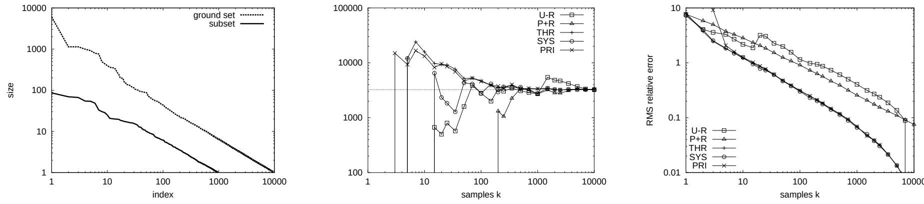
Uniform. Subset has 975 items and weight 483.45. Groundset has 10,000 items and weight 5005.02.



Pareto(1.75). Subset has 975 items and weight 2,123. Ground set has 10,000 items and weight 30,467.



Pareto(1.25). Subset has 975 items and weight 3,633. Groundset has 10,000 items and weight 267,178.



Pareto(0.75). Subset has 975 items and weight 37,212. Groundset has 10,000 items and weight 774,271,000.

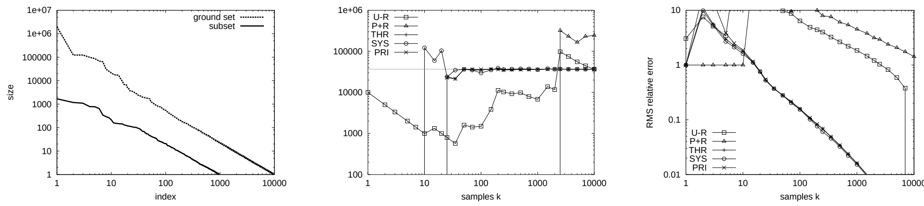


Fig. 6. Synthetic data. We start with a groundset of 10,000 weights chosen uniformly at random from $[0, 1]$. Each was picked with probability 1/10 for a random subset which got 975 items. For $\beta = 1.75, 1.25, 0.75$, we converted the data to Pareto(β) with the map $w_i \mapsto (1 - w_i)^{1/\beta}$. Left: Weight distribution for ground set and selection. Middle: Weight estimates as a function of sample size. Right: RMS relative error of weight estimates as function of sample size averaged over 1000 runs

two points to consider. One is how many samples it takes before we get one from a given application. This is the point at which we get our first non-zero estimate. Second we consider how quickly we converge after this point.

Number of samples needed to hit an application. With uniform sampling, the number of samples expected before we get one from a given application is roughly the total number of flows divided by the number of application flows. In that regard, ftp traffic is clearly the worst.

With probability proportional to size sampling with replacement, the expected number is roughly the total traffic divided by the application traffic. The worst application here is dns traffic which was the best application for uniform sampling.

Priority sampling is like probability proportional to size sampling with replacement but it avoids making duplicates of dominant items. If the dominant items are outside the application, we waste at most one sample on each. The impact is clear for dns traffic where we get the first sample about 30 times earlier with priority sampling than we did with probability proportional to size sampling with replacement. A more direct illustration of the problem is found in Figure 4 where we see how the fraction of distinct samples drops in probability proportional to size sampling with replacement.

Turning to the synthetic experiments in Figure 6, the subset always has the same roughly 10% of the items, so we expect a first hit after around 10 samples. In Pareto(0.75), uniform gets the subset already in the first sample. The probability of that event is 1/10, but given that we have 4 independent experiments, it is not so unlikely that one case comes out early. As the distribution gets more heavy-tailed, the subset gets a smaller and smaller fraction of the total weight, and hence it takes longer and longer before probability proportional to size sampling gets its first sample, the most extreme being Pareto(0.75) where the first hit is after more than 2000 samples. For contrast, priority sampling gets its first hit already after about 15 samples.

Convergence after first hitting an application. After we have started getting samples from an application, uniform sampling may still have problems with convergence. This typically occurs if the weight distribution within the application is heavy-tailed. Once again, ftp traffic is the worst application, this time because it has a dominant flow with more than 99% of its traffic. Until this flow is sampled, we expect to underestimate. If it is sampled early, we will hugely overestimate, although this is unlikely. The typical heavy-tail behavior is that the estimate grows as we catch up with more and more dominant items. We see this phenomena most clearly for the ftp traffic. Another good illustration is in the synthetic data where we expect to get the first hit after the same number uniform samples, yet the convergence with uniform sampling gets slower as the distribution gets more heavy tailed.

With probability proportional to size sampling with replacement and with priority sampling, we generally get quick convergence as soon as we start having samples from an application. Neither scheme has any problems with skewed weight distributions within a subset. For example, we see that probability proportional to size sampling with replacement starts slower than uniform on dns traffic, yet it ends up converging faster. If we consider the very best case for uniform sampling; namely dns traffic, we see that uniform is very quick at getting positive estimates, yet it continues to underestimate the total. This is because the uniform sampling misses out on the large dns items. Priority sampling has a slower start, but converges much better after about 300 samples. The dns example is also interesting in that the priority sample starts of with large over estimates because it is "lucky" to get an early dns sample.

The traffic matrix. Figure 5 shows the average relative error over $8 \times 8 = 64$ entries. We note first the poor performance of uniform sampling. In fact, it is only luck that the error with uniform is remains below 100%. This is because we miss the dominant items and get under-estimates that can never be by more than 100%. We could instead have gotten a dominant item early, leading to a huge over-estimate, but this is much less likely.

Comparing priority sampling with probability proportional to size sampling with replacement the faster convergence of priority sampling is very clear. For example, priority sampling gets down around a 1% error with about 150 samples whereas probability proportional to size sampling with replacement needs about 3000 samples, and the probability proportional to size sampling falls further behind with smaller errors because it gets more and more duplicates.

Root mean square (RMS) relative error. We now consider the root mean square relative error in the plots to the right in Figures 3 and 6. The RMS plots show our measured estimates of the relative standard deviation.

The real standard deviation is often dominated by rare large overestimates, so a question is if $r = 1000$ runs is enough for these rare events to show up. Consider the Internet data in Figure 3, we see that the curve for uniform sampling often starts low around 1, and then jumps up before we get to 100 samples. We claim that the curve is only a reasonable approximation to the relative standard deviation after the jump. To be more precise, we have $n = 85,680$ flows. With sample size k, the weight estimate of a sampled item is n/k times bigger than the original weight. Suppose a subset X has a dominant item $i \in X$ whose weight w_i is a fraction ε of the total subset weight W_I. If i is contained in a sample of size k, we get a subset estimate $\widehat{W_I}$ which is at least $w_i n/k$. With $r = 1000$ runs, we expect one of them to sample w_i with $k \approx n/r \approx 85$. If so, that run overestimates w_i by a factor $n/k \approx r$, hence W_I by at least a factor $r\varepsilon$. In the RMS, this alone contributes a factor $\sqrt{(r\varepsilon)^2/r} = \varepsilon\sqrt{r} \approx 30\varepsilon$. With ftp and multimedia we see from the plots of the data, we have a large dominant weight, hence ε close to 1, explaining the very high jump. With sessions and dns, the dominance is less pronounced, and the jump much smaller. In either case, we only trust the RMS curve after around 100 samples.

In the synthetic data in Figure 3, we have only 10,000 items, so a dominant item is expected to appear within $k = 10$ samples, hence earlier than before. In the synthetic data, the dominant subset weight is generally less dominant, and it is really only in the most heavy-tailed case that we see a pronounced jump for uniform sampling.

For probability proportional to size sampling with replacement, we can do a corresponding calculation. This time the rare event is that we get an early sample of any item from a subset. Let W be the total weight and W_I be the weight of a subset I. With $r = 1000$ runs, the subset should turn up among the first samples when $W/W_I < r$. If the subset is much lighter, it should start appearing after around $k = W/(rW_I)$ samples. The only case of such a very light subset is with Pareto(0.75) where $W/(rW_I) = 774,271,000/(1000 \cdot 37,212) \approx 20$, and indeed we see probability proportional to size sampling jump around this number of samples.

With priority sampling, we expect to reach items from a given subset at least as soon as probability proportional to size sampling as we do not have wasted duplicate samples. Indeed there is no case of a jump with priority sampling after $k = 2$.

Now, considering the curves after $k = 100$, it is very clear that priority sampling beats both of the standard alternatives, that is, uniform sampling and probability proportional to size sampling, on the real Internet data in Figure 3. Concerning the synthetic data in 6, we see that priority sampling continues to beat uniform sampling, but that it only beats probability proportional to size sampling in the more heavy-tailed cases of Pareto(1.25) and Pareto(0.75). In the less heavy-tailed cases, priority sampling ties with probability proportional to size sampling, except that priority sampling wins in the end. This is because the probability proportional to size sampling only starts having many duplicates when k gets close to n.

3.4 Priority sampling versus idealistic benchmarks

In our experiments, it appears that priority sampling (PRI) has a performance very similar to the idealistic benchmarks: threshold sampling (THR) and systematic threshold sampling (SYS). The middle plots based on a single run are a bit noisy with no clear general winner. The right hand side RMS plots have more stable curves. There we see that the three curves are almost always indistinguishable, except that SYS has a slight advantage on the real Internet data in Figure 3 for dns and perhaps a bit for sessions. The overall conclusion is, however, that the performances are very close.

The above closeness was not so unexpected. Recall that the threshold sampling threshold τ^{THR} is the fixed value such that the expected number of larger priorities is k. The priority sampling threshold τ^{PRI} is the priority which has exactly k larger priorities. Thanks to this similarity, we were hoping that priority sampling in its estimate quality would perform similarly to threshold sampling. Since systematic threshold sampling use the same marginal sampling probabilities on the items as threshold sampling, it was also expected that it would have similar performance. The fact that systematic threshold sampling does slightly better on subsets with a large number of items may have to do with negative covariance where threshold sampling has zero covariance. We shall return to this point in Section 5.4.

4. ANALYTIC COMPARISON OF VARIANCE IN SOME SIMPLE CASES

In this section, we will compare the different sampling schemes on some simple cases where we can analyze the variance, so as to gain some intuition for what is going on. This includes proving the infinite variance with a single priority sample claimed previously in Section 2.5. The results of the analysis will be used to formulate some general optimality theorems and conjectures in the next section.

We will focus on the variance for individual items. For threshold and priority sampling, this does give us the full picture, for with zero covariance, the variance of a subset sum estimate, is the sum of the variances of the items in the set.

Since threshold and systematic threshold sampling have the same marginal distribution for each item, they have the same individual variances, so below, when we analyze threshold sampling, we are really considering both of these schemes.

Most of the sampling schemes we deal with use so-called Horvitz-Thompson estimators, and it is useful to introduce some special notation for these: if w is a weight and $p \in [0, 1]$ a sampling probability, we let $\widehat{w}(p)$ denote the random variable that is w/p with probability p; 0 otherwise. Then

$$\begin{aligned} \mathsf{E}\left[\widehat{w}(p)\right] &= w \\ \mathsf{E}\left[(\widehat{w}(p))^2\right] &= p(w/p)^2 = w^2/p \\ \mathsf{Var}\left[\widehat{w}(p)\right] &= w^2/p - w^2 = w^2\frac{1-p}{p}. \end{aligned}$$

It is also convenient to define the function

$$v(w, \tau) = w \max\{0, \tau - w\}.$$

Then, with fixed threshold τ^{THR}, the variance for item i is

$$\begin{aligned} \mathsf{Var}[\widehat{w}_i^{THR}] &= \mathsf{Var}\left[\widehat{w}_i(\min\{1, w_i/\tau^{THR}\})\right] \\ &= w_i^2(1/\min\{1, w_i/\tau^{THR}\} - 1) \\ &= w_i \max\{0, \tau - w_i) \\ &= v(w_i, \tau^{THR}). \end{aligned}$$

With our new priority sampling, the threshold changes, and the variance of item i is

$$\mathsf{Var}\left[\widehat{w}_i\right] = \int_{\tau'=0}^{\infty} f(\tau')v(w_i, \tau')\, d\tau', \tag{11}$$

where $f(\tau')$ is the probability density function for τ' to be the k^{th} threshold amongst the items $j \neq i$. With τ' thus defined, by Lemma 2.1, item i is picked if $q_i = w_i/\alpha_i > \tau'$ with $\widehat{w}_i = \tau'$; 0 otherwise. This imitates the fixed threshold scheme with $\tau' = \tau^{THR}$. Thus (11) follows from the previous calculation with a fixed threshold. A similar calculation was made in Section 2.3 but without the reference to threshold sampling.

Sometimes it is easier with a more direct calculation. Summing over all $j \neq i$, we integrate over choices of α_j, multiply with the probability that $q_j = w_j/\alpha_j$ is the kth highest priority from $[n] \setminus \{i\}$, and multiply with the variance $v(w_i, q_j)$. That is,

$$\mathsf{Var}\left[\widehat{w}_i\right] = \sum_{j\in[n]\setminus\{i\}} \int_0^1 \Pr[|\{h \in [n] \setminus \{i, j\}|q_h \succ q_j\}| = k - 1]\, v(w_i, q_j)\, d\alpha_j. \tag{12}$$

4.1 Infinite variance with single priority sample

We will show that if we only make a single priority sample with $k = 1$, then the variance of any weight estimate is infinite. The proof is based on (12). We assume $i = 0$. For a lower-bound, in the sum, we only need to consider one other item

$j = 1$. Also, when integrating over α_1, we only consider very small values of α_1. More precisely, define $\varepsilon = w_1/(2W)$ where W is the sum of all weights. If $\alpha_1 < \varepsilon$, we have $q_1 = w_1/\alpha_1 > 2W$, and then

$$\begin{aligned}
\Pr[|\{h \in [n] \setminus \{i,j\} | q_h \succ q_j\}| = k-1] &= \Pr[|\{h \in \{2,...,n-1\} | q_h \succ q_1\}| = 0] \\
&> 1 - \sum_{h \in \{2,\ldots,n-1\}} \Pr[q_h > 2W] \\
&= 1 - \sum_{h \in \{2,\ldots,n-1\}} (w_h/2W) \\
&> 1/2.
\end{aligned}$$

Moreover, for $\alpha_1 < \varepsilon = w_1/(2W)$, we have $q_1 = w_1/\alpha_1 > 2W$, and hence

$$v(w_i, q_j) = v(w_0, q_1) = w_0 \max\{0, q_1 - w_0\} > w_0 w_1/(2\alpha_1).$$

Thus, by (12), we have

$$\mathsf{Var}\,[\widehat{w}_0] > \int_0^{\varepsilon} 1/2 \cdot w_0 w_1/(2\alpha_1)\, d\alpha_1 = \infty.$$

We note that none of the other sampling schemes considered can get infinite variance.

Next, we argue that the variance is bounded if we make at least two priority samples. Again, we focus on the variance for item $i = 0$. Also, it suffices to show that the integral in (12) is finite for each value of j, that is, we want to show that

$$V_{i,j} = \int_0^1 \Pr[|\{h \in [n] \setminus \{i,j\} | q_h \succ q_j\}| = k-1]\, v(w_i, q_j)\, d\alpha_j$$

is bounded. Now, for $k \geq 2$,

$$\begin{aligned}
\Pr[|\{h \in [n] \setminus \{i,j\} | q_h \succ q_j\}| = k-1] &\leq \Pr[|\{h \in [n] \setminus \{i,j\} | q_h \succ q_j\}| \geq 1] \\
&\leq \sum_{h \in [n] \setminus \{i,j\}} \Pr[q_h \succ q_j] \\
&= \sum_{h \in [n] \setminus \{i,j\}} \Pr[w_h/\alpha_h > w_j/\alpha_j] \\
&= \sum_{h \in [n] \setminus \{i,j\}} \min\{1, w_h \alpha_j / w_j\} \\
&\leq \sum_{h \in [n] \setminus \{i,j\}} (w_h \alpha_j / w_j) \\
&< W \alpha_j / w_j.
\end{aligned}$$

Moreover,

$$v(w_i, q_j) = w_i \max\{0, w_j/\alpha_j - w_i\} \leq w_i w_j / \alpha_j,$$

so we get that

$$V_{i,j} < \int_0^1 W\alpha_j/w_j \cdot w_i w_j/\alpha_j\, d\alpha_j = \int_0^1 W w_i\, d\alpha_j = W w_i.$$

Hence

$$\mathsf{Var}\left[\widehat{w}_i\right] = \sum_{j\in[n]\setminus\{i\}} V_{i,j} < n\,W w_i,$$

so indeed the variance is bounded. Since the covariance is zero, it also follows that estimates of weights of subsets are bounded. Thus we have proved

PROPOSITION 4.1. *If we make a single priority sample, then all weight estimates have infinite variance. With more than one priority samples, all weight estimates have finite variance.*

By contrast, with all the other sampling schemes, the variance estimates are finite as soon as we make at least one sample.

4.2 Unit weights

We will now study identical unit weights, focusing on the first item $i = 0$. We will compute the exact variance for each of the sampling schemes considered.

U−R. For uniform sampling without replacement, item 0 is picked with probability $p_0^{U-R} = k/n$, so

$$\mathsf{Var}\left[\widehat{w}_0^{U-R}\right] = \frac{1-p_0^{U-R}}{p_0^{U-R}} = \frac{n-k}{k}.$$

P+R. For probability proportional to size sampling with replacement, item 0 is picked with probability $p_0^{P+R} = 1-(1-1/n)^k$, hence with

$$\mathsf{Var}\left[\widehat{w}_0^{P+R}\right] = \frac{1-p_0^{P+R}}{p_0^{P+R}} = \frac{(1-1/n)^k}{1-(1-1/n)^k}.$$

For $k \ll n$, the variance approaches $\frac{n-k}{k}$ from above. However, for $k = n$, the variance approaches $1/(e(1-e^{-1}) = 0.58...$

THR/SYS. In the fixed threshold scheme from [Duffield et al. 2005a], we set $\tau^{THR} = n/k$. Then

$$\mathsf{Var}\left[\widehat{w}_0^{THR}\right] = v(w_0, \tau^{THR}) = w_1 \max\{0, \tau^{THR} - w_i\} = \frac{n-k}{k}. \tag{13}$$

PRI. For priority sampling, we will evaluate (12) exactly. By symmetry, it suffices to consider the case where q_1 becomes the threshold, and then multiply by a factor $n-1$. We use that

$$\Pr[q_h \succ q_1] = \Pr[\alpha_h < \alpha_1] = \alpha_1$$

and

$$v(w_0, q_1) = w_0 \max\{0, q_1 - w_0\} = (1/\alpha_1 - 1).$$

Hence

$$\begin{aligned}
\mathsf{Var}\left[\widehat{w}_0\right] &= (n-1)\int_{\alpha_1=0}^{1} \Pr[|\{h \in \{2,...,n-1\} | q_h \succ q_1\}| = k-1]\, v(w_0, q_1)\, d\alpha_1 \\
&= (n-1)\int_{\alpha_1=0}^{1} \Pr[B(n-2,\alpha) = k-1]\,(1/\alpha_1 - 1)\, d\alpha_1
\end{aligned}$$

$$= (n-1)\binom{n-2}{k-1}\int_{\alpha_1=0}^{1} \alpha_1^{k-2}(1-\alpha_1)^{n-k}\, d\alpha_1$$
$$= (n-1)\binom{n-2}{k-1}\frac{(k-2)!\,(n-k)!}{(n-1)!}$$
$$= \frac{n-k}{k-1}.$$

Discussion. For unit weights, uniform sampling without replacement and threshold sampling gets the same variance on single item weight estimates; namely $\frac{n-k}{k}$. When k is not too small, priority sampling gets nearly the same variance; namely $\frac{n-k}{k-1}$. Weighted sampling with replacement starts doing well, but gets worse and worse as k grows. In particular, for any $k \geq n$, it has positive variance while all the other schemes have zero variance since they have no replacement.

4.3 Large and small weights

In this section we illustrate what happens when different weights are involved. We consider the case where we have ℓ large weights of weight N and $n > \ell$ small unit weights. The large weights are first, that is, $w_0 = \cdots = w_{\ell-1} = N$ while $w_\ell = \cdots = w_{n+\ell-1} = 1$. We let $W = \ell N + n$ denote the total weight. We will use w_0 as a representative for the large items and w_n as a representative for the small items. Besides studying the individual variances, we will study the *total variance* ΣV measured as the sum of the individual variances. The variances from the different sampling schemes will be collected in Table I.

As notation we use $x \approx y \iff x = (1 \pm o(1))y$ and $x \ll y \iff x = o(y)$. We view ℓ, n, and N as unbounded. We assume $\ell \ll n \ll \sqrt{N}$ and that $k \ll n$. These assumptions will help simplifying the analysis.

U−R. For uniform sampling without replacement, the large item 0 is picked with probability $p_0^{U-R} = k/(n+\ell)$, hence with

$$\mathsf{Var}\left[\widehat{w}_0^{U-R}\right] = N^2\frac{1-p_0^{U-R}}{p_0^{U-R}} = N^2\frac{n+\ell-k}{k} \approx N^2\frac{n}{k}.$$

For small item n, we have the same sampling probability, $p_n^{U-R} = k/(n+\ell)$, so we get

$$\mathsf{Var}\left[\widehat{w}_n^{U-R}\right] = \frac{1-p_n^{U-R}}{p_n^{U-R}} \approx \frac{n}{k}.$$

Then $\Sigma V \approx \ell N^2 n/k$.

P+R. For probability proportional to size sampling with replacement, the large item 0 is picked with probability $p_0^{P+R} = 1-(1-N/W)^k$ hence with

$$\mathsf{Var}\left[\widehat{w}_n^{P+R}\right] = N^2\frac{1-p_0^{P+R}}{p_0^{P+R}} \approx N^2\frac{1-(1-N/W)^k}{(1-N/W)^k} = N^2(1/(1-N/W)^k - 1).$$

In particular, this is $\Theta(N^2)$ for $k = \Theta(\ell)$. Yet it saves a factor n/k over uniform sampling with replacement in the case of large weights.

For probability proportional to size sampling with replacement, the small item $i = n$ is picked with probability $p_n^{P+R} = 1-(1-1/W)^k \approx k/W \approx k/(\ell N) \ll 1$, hence with

$$\mathsf{Var}\left[\widehat{w}_n^{P+R}\right] = \frac{1-p_n^{P+R}}{p_n^{P+R}} \approx N\ell/k.$$

Then $\Sigma V^{P+R} \approx N^2\ell/(1/(1-N/W)^k-1) + Nn\ell/k$.

THR/SYS. For the fixed threshold scheme, if $k \leq \ell$, we set $\tau^{THR} = W/k > N$. Then for large item 0,

$$\mathsf{Var}\left[\widehat{w}_0^{THR}\right] = v(w_0, \tau^{THR}) = N(W/k - N) \approx \begin{cases} N^2\frac{\ell-k}{k} & \text{if } k < \ell \\ Nn/\ell & \text{if } k = \ell. \end{cases}$$

For light item n with $k \leq \ell$, we get

$$\mathsf{Var}\left[\widehat{w}_n^{THR}\right] = v(w_n, \tau^{THR}) = (W/k - 1) \approx N\ell/k.$$

On the other hand, for $k > \ell$, we pick a threshold below N; namely $\tau^{THR} = (n-\ell)/(k-\ell)$. Then for large item 0,

$$\mathsf{Var}\left[\widehat{w}_0^{THR}\right] = 0$$

while for a light item n, it is

$$\mathsf{Var}\left[\widehat{w}_n^{THR}\right] = v(w_n, \tau^{THR}) = (n-\ell)/(k-\ell) - 1 \approx n/(k-\ell).$$

For $k < \ell$, the total variance is dominated by the large items, adding up to $\Sigma V^{THR} \approx N^2\ell\frac{\ell-k}{k}$. For $k = \ell$, the variance over the large items is approximately Nn but this is also the variance over the small items, so $\Sigma V^{THR} \approx 2Nn$. For $k > \ell$, we only have variance for the small items, so $\Sigma V^{THR} \approx n^2/(k-\ell)$.

We note that with $k > 1$, the variance with threshold sampling is always smaller than with probability proportional to size with replacement. Since both use Horvitz-Thompson estimators, this follows directly because the sampling probability with threshold sampling is larger for any items.

PRI. First we consider the large item 0. Let m be the event that we have exactly m small items with priorities bigger than N. Then

$$\mathsf{Var}\left[\widehat{w}_0\right] = \sum_{m=0}^{n} \Pr[A(m)] \left(\mathsf{E}\left[\widehat{w}_0^2 | A(m)\right] - w_0^2\right)$$

Trivially $\Pr[A(m)] = \Pr[B(n, 1/N) = m]$. Moreover, we saw in Section 2 that $\widehat{w}_0$ is unbiased no matter the remaining priorities, including all priorities of small items. Hence $\mathsf{E}\left[\widehat{w}_0 | A(m)\right] = w_0$, so

$$\mathsf{E}\left[\widehat{w}_0^2 | A(m)\right] - w_0^2 = \mathsf{E}\left[\widehat{w}_0^2 | A(m)\right] - \mathsf{E}\left[\widehat{w}_0 | A(m)\right]^2 = \mathsf{Var}\left[\widehat{w}_0 | A(m)\right].$$

Therefore

$$\mathsf{Var}\left[\widehat{w}_0\right] = \sum_{m=0}^{n} \Pr[B(n, 1/N) = m]\, \mathsf{Var}\left[\widehat{w}_0 | A(m)\right].$$

Consider a small item i. Conditioned on having a big priority $q_i > N$, item i acts like a large item. Conversely, conditioned on having a small priority $q_i < N$, item i has no impact on the weight estimate of large items. Thus, in the event $A(m)$, the variance of item 0 is as if we had $\ell + m$ large items and no small items. If $\ell + m \leq k$, the threshold is at most N, and then there is no variance. If $\ell + m > k$, the analysis from the all unit weights case shows that

$$\mathsf{Var}[\widehat{w}_0 | A(m)] = N^2 \frac{\ell + m - k}{k-1}.$$

Thus

$$\begin{aligned} \mathsf{Var}[\widehat{w}_0] &= \sum_{m=0}^{n} \Pr[A(m)] \, \mathsf{Var}[\widehat{w}_0 | A(m)] \\ &= \sum_{m=\max\{0,k-\ell+1\}}^{n} \Pr[B(n, 1/N) = m] \, N^2 \frac{\ell + m - k}{k-1}. \end{aligned}$$

Since $N \gg n^2$, the first term dominates. If $k < \ell$, we get $m = 0$, and then

$$\mathsf{Var}[\widehat{w}_0] \approx \Pr[B(n, 1/N) = 0] \, N^2 \frac{\ell - k}{k-1} \approx N^2 \frac{\ell - k}{k-1}.$$

If $k \geq \ell$, we get $m = k - \ell + 1$, and then

$$\begin{aligned} \mathsf{Var}[\widehat{w}_0] &\approx \Pr[B(n, 1/N) = k - \ell + 1] \, N^2/(k-1) \\ &\approx \binom{n}{k - \ell + 1} N^{1-k+\ell}/k \\ &\approx \begin{cases} Nn/k & \text{if } k = \ell \\ n^2/(2k) & \text{if } k = \ell + 1 \\ o(n/k) & \text{if } k > \ell + 1. \end{cases} \end{aligned}$$

We now consider the small item n. We are going to prove that $\mathsf{Var}[\widehat{w}_n] \approx N\ell/(k-1)$ if $k \leq \ell$, $\mathsf{Var}[\widehat{w}_n] \approx n \ln N$ if $k = \ell + 1$, and $\mathsf{Var}[\widehat{w}_n] \approx n/(k - \ell - 1)$ if $k > \ell + 1$.

We consider two different contributions to the variance depending on whether the threshold τ is greater than N. If $\tau > N$, we further distinguish depending on whether $q_n > N$. If $\tau > N$ and $q_n \leq N$, then $\widehat{w}_n = 0$ so $(\widehat{w}_n - w_n)^2 = 1$. The variance contribution of this event is therefore at most 1, and this will not impact the final small item variance.

Next consider the case that $\tau > N$ and $q_n > N$. The probability that $q_n > N$ is $1/N$. Let $A'(m)$ denote that event that we have m small items $i \neq n$ with $q_i > N$. Conditioned on $q_n > N$, we have $\tau > N$ if and only if $m \geq k - \ell$. In this case, the variance contribution is $\mathsf{E}\left[\widehat{w}_n^2\right] - 1$. However, conditioned on $q_n > N$, q_n is distributed like a priority of a large item, and hence $\widehat{w}_n$ behaves like the weight estimate of large item among $\ell + m + 1$ large items. Thus $\mathsf{E}\left[\widehat{w}_n^2 | q_n > N \wedge A'(m)\right] = N^2 \frac{\ell+m+1}{k-1}$, so we get a variance contribution of

$$\begin{aligned} &\Pr[q_n > N] \sum_{m=\max\{0,k-\ell\}}^{n-1} \Pr[B(n-1, 1/N) = m] \, (N^2 \frac{\ell + m + 1}{k-1} - 1) \\ &\approx 1/N \cdot \Pr[B(n-1, 1/N) = \max\{0, k - \ell\}] \, N^2 \frac{\ell + \max\{0, k - \ell\}}{k-1} \end{aligned}$$

$$\approx \begin{cases} N\ell/(k-1) & \text{if } k \le \ell \\ n & \text{if } k = \ell+1 \\ o(1) & \text{if } k > \ell+1. \end{cases}$$

When $k \le \ell$, the above estimate of $N\ell/(k-1)$ will dominate the small item variance. For $k > \ell$, we will find larger contributions below.

Finally we consider the case where $\tau \le N$. For light item n to be sampled, we need $k > \ell$, and then this is like the all unit weights case, except that we only sample $k' = k - \ell$ items. Hence we can apply the integral from the all unit weight case, but with the restriction that $\alpha \ge 1/N$. We then get a variance contribution of

$$(n-1)\int_{\alpha=1/N}^{1} \Pr[B(n-2,\alpha) = k'-1]\,(1/\alpha - 1)\,d\alpha$$
$$= (n-1)\binom{n-2}{k'-1}\int_{\alpha=1/N}^{1} \alpha^{k'-2}(1-\alpha)^{n-k'}\,d\alpha.$$

For $k' \ge 2 \iff k > \ell+1$, the impact of starting the integral at $1/N$ instead of at 0 is not significant, so as in the all unit weights case, we get an variance contribution which is approximately

$$\frac{n-k'}{k'-1} = \frac{n-k+\ell}{k-\ell-1} \approx \frac{n}{k-\ell-1}.$$

This is the dominant small item variance contribution for $k > \ell+1$.

For $k' = 1 \iff k = \ell+1$, we get a variance contribution of

$$(n-1)\int_{\alpha=1/N}^{1}\binom{n-2}{k'-1}\alpha^{k'-2}(1-\alpha)^{n-k'}\,d\alpha \approx n\int_{\alpha=1/N}^{1}\alpha^{-1}\,d\alpha = n\ln N.$$

This is the dominant small item variance contribution for $k = \ell+1$.

We now consider the total variance with priority sampling. For $k < \ell$, the total variance is dominated by the large items, adding up to $N^2\ell\frac{\ell-k}{k-1}$. For $k = \ell$, the variance over the large items is approximately Nn but this is also the variance over the small items, so $\Sigma V^{PRI} \approx 2Nn$. For $k = \ell+1$, we get a contribution of $n^3/(2k)$ from the large items and $n^2 \ln N$ from the small variance. Finally, for $k > \ell+1$, the variance is dominated by the small items who have a total variance of $n^2/(k-\ell-1)$.

This completes the analysis of priority sampling for large and small weights. A comparison of all the sampling schemes is summarized in Table I.

Discussion. With reference to Table I, the problem with uniform sampling is that it does a terrible job on the large weights, performing about n/ℓ times worse than the other schemes. On the other hand, it gives the best performance on the small items. However, the advantage over threshold and priority sampling becomes insignificant when $k \gg \ell$. This illustrates that if the number of large items is small compared with the number of samples, then threshold and priority sampling do very well even on the small items.

The problem in probability proportional to size sampling with replacement is that it does poorly compared with threshold and uniform sampling when the number of samples exceeds the number of large items. This is for both large and small items,

	$1 \le k < \ell$	$k = \ell$	$k = \ell + 1$	$k > \ell + 1$
large item				
U−R	$N^2 n/k$			
P+R	$N^2/(1/(1-N/W)^k - 1)$			
THR/SYS	$N^2 \frac{\ell-k}{k}$	Nn/ℓ	0	
PRI	$N^2 \frac{\ell-k}{k-1}$	Nn/ℓ	$n^2/(2k)$	$o(n/k)$
small item				
U−R	n/k			
P+R	$N\ell/k$			
THR/SYS	$N\ell/k$		$n/(k-\ell)$	
PRI	$N\ell/(k-1)$		$n \ln N$	$n/(k-\ell-1)$
Total variance				
U−R	$N^2 \ell n^2/k$			
P+R	$N^2\ell/(1/(1-N/W)^k - 1) + Nn\ell/k$			
THR/SYS	$N^2\ell \frac{\ell-k}{k}$	$2nN$	$n^2/(k-\ell)$	
PRI	$N^2\ell \frac{\ell-k}{k-1}$	$2nN$	$n^2\ell/(2k) + n^2 \ln N$	$n^2/(k-\ell-1)$

Table I. Overview of approximate variance with k samples, in the case of ℓ large items of size N and $n - \ell$ items of small weight. We assume $1 \ll \ell \ll n \ll \sqrt{N}$ and $1 \ll k \ll n$.

illustrating the problem with duplicates. In particular, for $k \gg \ell$, we see that the small item variance with probability proportional to size with replacement is nearly a factor W/n bigger than that for threshold and priority sampling. This difference also shows up in the total variance.

Finally, comparing threshold and priority sampling, we see that priority sampling has positive variance for large items when $k > \ell$ whereas threshold sampling has no variance. However, this variance of priority sampling is small compared to the size N of a large weight, so this is a case where priority sampling is doing well anyway. It is more interesting to see what happens with the small items. The major differences are in the two boundary cases when $k = 1$ and when $k = \ell + 1$. The former case has infinite variance as discussed previously. For $k = \ell + 1$, we see that priority sampling does worse by a factor of $\ln N$. However, these boundary cases are quite special and it is therefore not surprising that this kind of difference did not show up in any of our experiments.

Thus, in our analysis, priority sampling performs much better than the standard sampling schemes U−R and P+R, and for most cases of k, it is close to the idealistic benchmarks THR and SYS.

5. VARIANCE OPTIMALITY

We are now going to discuss the variance optimality of threshold, systematic threshold, and priority sampling, and contrast it with the non-optimality of uniform sampling and probability proportional to size sampling with replacement. First we show that threshold sampling minimizes the variance relative to the expected number of samples. As in the previous section, we are here considering the *total variance* measured as the sum of variances for the individual items. Next we discuss how the optimality of threshold sampling should imply a certain near-optimality for priority sampling, capturing this in a concrete conjecture. With threshold and priority sampling we have zero covariance, and then the total variance is the sum of variances over any partitioning of the items into sets to be estimated. Considering cases with non-zero covariance like systematic threshold sampling, we finish this section discussing the relationship between our total variance and variances over random subsets of different sizes.

5.1 Threshold sampling

In this subsection, we prove that threshold sampling minimizes the total variance relative to the expected number of samples. Some of this is partly known, at least implicitly, but here we need the results in a concrete strong form. The optimality will also hold for systematic threshold sampling which has exactly the same total variance as threshold sampling because the estimate of each item follows the same marginal distribution.

In Poisson sampling, for each item i with weight w_i, we have to decide on a sampling probability p_i. If i is not picked, the weight estimate is zero, that is, $\widehat{w}_i(p_i) = 0$. To get an unbiased estimator, if item i is picked, it gets the Horvitz-Thompson weight estimate $\widehat{w}_i(p_i) = w_i/p_i$. Generally, we want to sample few items, yet keep the variance low. This motivates an objective of the form

$$\text{minimize } p_i + \alpha \, \mathsf{Var}\left[\widehat{w}_i(p_i)\right].$$

Here

$$\mathsf{Var}\left[\widehat{w}_i(p_i)\right] = \mathsf{E}\left[(\widehat{w}_i(p_i))^2\right] - w_i^2 \text{ where } \mathsf{E}\left[(\widehat{w}_i(p_i))^2\right] = p_i(w_i/p_i)^2 = w_i^2/p_i.$$

Thus we want to minimize $p_i + \alpha\, w_i^2/p_i$. For $p_i \in [0,1]$, the unique solution is to set

$$p_i = \min\{1, \sqrt{\alpha} w_i\}.$$

Setting $\sqrt{\alpha} = 1/\tau^{THR}$, we get $p_i = \min\{1, \sqrt{\alpha} w_i\} = \min\{1, w_i/\tau^{THR}\}$, which is the sampling probability of threshold sampling with threshold $\tau^{THR} = 1/\sqrt{\alpha}$. Conversely, this means that for any choice of τ^{THR}, the fixed threshold scheme picks the $p_i = \max\{1, w_i/\tau^{THR}\}$ so as to

$$\text{minimize } p_i + (1/\tau^{THR})^2 \, \mathsf{Var}\left[\widehat{w}_i(p_i)\right]. \tag{14}$$

Summing over the whole stream of items, we

$$\text{minimize } \sum_{i\in[n]} \left(p_i + (1/\tau^{THR})^2 \, \mathsf{Var}\left[\widehat{w}_i(p_i)\right]\right) = \sum_{i\in[n]} p_i + 1/(\tau^{THR})^2 \, \mathsf{Var}\left[\sum_{i\in[n]} \widehat{w}_i(p_i)\right]. \tag{15}$$

To get a specified number of k expected samples, we choose τ^{THR} such that

$$\sum_{i\in[n]} p_i = \sum_{i\in[n]} \min\{1, w_i/\tau^{THR}\} = k.$$

This scheme is denoted THR_k.

LEMMA 5.1. *Among Poisson sampling scheme with an expected number of at most k samples,* THR_k *minimizes the total variance.*

PROOF. If some scheme used the same number or fewer expected samples than THR_k and simultaneously got smaller sum of variances, then this scheme would reduce (15), contradicting that (15) is minimized by THR_k. □

We are now going to strengthen Lemma 5.1 to cover all possible sampling schemes. For a given set of input weights $w_1, ... w_n$, we think abstractly of a sampling and estimation scheme as a probability distribution Φ over functions $\widehat{w}$ mapping items i into estimates $\widehat{w}_i$. We want the estimators to be unbiased in the sense that $\mathsf{E}_{\widehat{w}\leftarrow\Phi}[\widehat{w}_i] = w_i$. For a given $\widehat{w} \in \Phi$, the number of samples is the number of non-zeros.

PROPOSITION 5.2. *For any weight sequence $w_0, ..., w_{n-1}$ and positive k, there is no sampling scheme Φ with unbiased estimators and an expected number of at most k samples which has smaller total variance than* THR_k.

PROOF. Let Φ be a sampling scheme with unbiased estimation of the w_i and an expected number of at most k samples. We will first convert Φ into a Poisson sampling scheme and then use Lemma 5.1 to conclude that THR_k has no bigger total variance.

As a first transformation, for each item i independently, we create a random variable w'_i whose distribution is the marginal distribution of $\widehat{w}_i$ in Φ, that is, $\Pr[w'_i = x] = \Pr_{\widehat{w}\leftarrow\Phi}[\widehat{w}_i = x]$ for any $x \in \mathbb{R}$. Then $\Pr[w'_i > 0] = \Pr_{\widehat{w}\leftarrow\Phi}[\widehat{w}_i > 0]$ and $\mathsf{Var}[w'_i] = \mathsf{Var}_{\widehat{w}\leftarrow\Phi}[\widehat{w}_i]$, so the transformation has neither changed the total variance, nor the expected number of non-zeros.

We now create a new random variable w''_i as a function of w'_i that is 0 when w'_i is 0, and equals $\bar{w}_i = \mathsf{E}[w'_i \,|\, w'_i > 0]$ when $w'_i > 0$. Then $\mathsf{E}[w''_i] = \mathsf{E}[w'_i] = w_i$, so the random variables w''_i represent a Poisson sampling scheme providing unbiased estimation of the w_i: each item i is sampled independently with estimate $\bar{w}_i$ when $w'_i > 0$. Moreover, for each i, $\Pr[w''_i > 0] = \Pr[w'_i > 0] = \Pr_{\widehat{w}\leftarrow\Phi}[\widehat{w}_i > 0]$ while $\mathsf{Var}[w''_i] \le \mathsf{Var}[w'_i] = \mathsf{Var}_{\widehat{w}\leftarrow\Phi}[\widehat{w}_i]$.

The Poisson sampling scheme represented by the w''_i thus has the same expected number of samples/non-zeros as Φ which was at most k, so from Lemma 5.1 it follows that the total variance of the w''_i is at least as big that of THR_k. However, the total variance with Φ is at least as big as that of the w''_i, so we conclude that the total variance of Φ is at least as big as that of THR_k. □

Previous Results on Optimality.. When τ^{THR} is bigger than the largest weight, the sampling probabilities are all proportional to size. In this case it is standard that the variance is minimized relative to the expected number of samples (see, e.g., [Särndal et al. 1992, p. 86]). It is also a standard idea (see, e.g., [Särndal et al. 1992, p. 90]) to include very large weights with probability one. This is normally defined iteratively, saying that if the total weight is W and the desired number of samples is k, we include all weights beyond W/k with probability 1, and sample from the rest. Recursively, we may find more weights that should be sampled with probability 1 among the rest. However, it appears to be [Duffield et al. 2005a] that names threshold sampling defining it generally in terms of a given threshold parameter τ^{THR}. Also, the proof of (15) is from [Duffield et al. 2005a]. In [Duffield et al. 2005a] the focus was somewhat different, and it was never made explicit that threshold sampling minimizes the total variance relative to the expected number of samples as stated here in Lemma 5.1. The simple relation to general sampling in Proposition 5.2 is also new.

5.2 Conjectured near-optimality of priority sampling

We saw above that threshold sampling minimizes the total variance relative to the expected number of samples. Technically speaking, priority sampling is very similar to threshold sampling. The technical difference is that threshold sampling uses the threshold τ^{THR} such that we expect k bigger priorities, while priority sampling uses the actual $(k+1)^{st}$ priority as the threshold τ^{PRI}.

We would like to argue that priority sampling inherits some of the variance optimality of threshold sampling. For unit items, threshold sampling got an item variance of $\frac{n-k}{k}$ while priority sampling got an item variance of $\frac{n-k}{k-1}$. The difference is infinite for $k = 0$. Another more interesting example is the large and small weights from Section 4.3. With ℓ large weights and $k = \ell + 1$ samples, the total variance with priority sampling was a factor $\ln N$ worse than that of threshold sampling, and this factor can be arbitrarily large. However, instead of comparing an expected number of k threshold samples with k priority samples, we suggest giving priority sampling one extra sample, e.g., comparing the bad case priority sampling with $\ell + 1$ samples with the worse case of threshold sampling with an expected number of ℓ samples. We conjecture that priority sampling with one extra sample always beats threshold, that is,

Conjecture 5.3. *For any weight sequence $w_0, ..., w_{n-1}$ and positive integer k, threshold sampling with an expected number of k samples gets a total variance which is no smaller than with a priority sample of size $k + 1$.*

One consequence of Conjecture 5.3 is that if we only have resources for a certain number k of samples, then we are much better off using priority sampling than using threshold sampling; for with threshold sampling, we would have to aim for a much smaller expected number of samples, e.g., $k - 2\sqrt{k}$, in order to make the probability of getting more than k samples small.

By Proposition 5.2, Conjecture 5.3 is equivalent to saying that priority sampling with one extra sample beats any possible sampling scheme, that is,

Conjecture 5.4. *For any weight sequence $w_0, ..., w_{n-1}$ and positive integer $k \leq n$, there is no sampling scheme with unbiased estimators and an expected number of at most k samples which has smaller total variance than with a priority sample of size $k + 1$.*

As evidence for Conjecture 5.3, we note from Section 4.2 that it holds true for the unit case. Also, for the large-small example in Section 4.3, the analysis implies Conjecture 5.3 within a factor $(1 \pm o(1))$. Finally, we note that the conjecture conforms nicely with the closeness of priority sampling and threshold sampling in the experiments from Section 3.

Recently, Szegedy [2006] has settled Conjecture 5.3, and then, by Proposition 5.2, Conjecture 5.4 follows. Thus priority sampling is variance optimal modulo one extra sample.

5.3 Anti-optimality of uniform sampling and probability proportional to size sampling with replacement

In the previous subsection, we discussed optimality of priority sampling based on giving it one extra sample. No such general optimality is possible for uniform sampling or probability proportional to size sampling with replacement. More precisely consider the case of large and small items from the previous section, and suppose the number of samples is much larger than the number of large items. This corresponds to $\ell \ll k$ in Table I. In that case, threshold sampling had a total variance of $\approx n^2/k$ the total variance of uniform sampling is $\approx N^2 \ell n^2/k$ and the total variance of probability proportional to size sampling is $\approx Nn\ell/k$. In both cases we have a dependence of the size N of the largest item which cannot be absorbed, e.g., by allowing 10 times as many samples.

5.4 Variance over larger subsets and systematic threshold sampling

Above we have discussed variance optimality of threshold and priority sampling with respect to the total variance measured as the sum

$$\Sigma V = \sum_i \mathsf{Var}[\widehat{w}_i].$$

However, the intended application of priority sampling is the estimation of sums over subsets of arbitrary sizes, not just singleton sets. In [Szegedy and Thorup 2007] it is suggested that one studies the average variance over sets of a given size m. We denote this quantity $V_{m:n}$. Then $V_{1:n} = \Sigma V/n$. With zero covariance as in threshold and priority sampling, we have $V_{m:n} = \frac{m}{n}\Sigma V$. Now $V_{m:n}$ could be smaller if we have negative covariance. In [Szegedy and Thorup 2007] it is shown that $V_{m:n} \geq \frac{n-m}{n-1}\frac{m}{n}\Sigma V$ for any sampling scheme with unbiased estimators, with equality if and only if the estimate of the total sum is exact, that is, $\sum_{i\in[n]} \widehat{w}_i = \sum_{i\in[n]} w_i$. Systematic threshold sampling has the same optimal total variance ΣV as threshold sampling, and it estimates the total exactly, so it minimizes $V_{m:n}$ for any set size m, improving over threshold sampling by a factor $\frac{n-1}{n-m}$. Modulo one extra sample, this is also the largest scope for improvement over priority sampling.

We note that the factor $\frac{n-1}{n-m}$ is only significant when m is not too far from n. In practice, we do not view the factor as much of an obstacle. For example, in a partitioning such as a traffic matrix in our Internet application, or a division into traffic classes, there can be at most one subset of size bigger than $n/2$, and generally, the larger subset is the one that is expected to have the smallest relative error, hence the least of our worries. More formally, with zero covariance, the average variance $V_{m:n}$ and the average sum both grow linearly in the subset size m. For sets achieving both of these averages, this means that the relative standard deviation falls as $1/\sqrt{m}$.

If we are very interested in estimating the weight of large subsets containing a large fraction of all items, then we may consider the following adjusted weight estimation: besides maintaining the priority sample, we maintain a counter W for the exact total weight $\sum_{i\in[n]} w_i$. If $\widehat{W} = \sum_{i\in S} \widehat{w}_i$ is the priority estimate of the total weight, we derive an adjusted weight estimate $\widehat{w}_i^{adj}$ from the priority weight estimate $\widehat{w}_i$ as follows.

$$\widehat{w}_i^{adj} = (W/\widehat{W})\widehat{w}_i.$$

In particular, we get that the adjusted estimate of the total $\sum_{i\in S}\widehat{w}_i^{adj}$ is exactly W. Such adjustments are standard (see, e.g., [Särndal et al. 1992, p. 87]) but the result is biased, and we will not pursue this option any further in this paper.

Even though systematic threshold sampling is optimal with respect to the average variance $V_{m:n}$, it is not a good choice in a streaming context. Besides implementation issues that we will discuss in the next section, systematic threshold sampling may have very strong correlations between items in a subset depending on how they are placed in the stream. Normally, it is recommended that the items are appropriately shuffled [Särndal et al. 1992, p. 92], but that is not possible in reservoir sampling (c.f. Section 1.2) from a stream. With threshold and priority sampling there is no such dependence. As demonstrated in [Thorup 2006], it is possible to get good confidence bounds with priority sampling and threshold sampling so that we statistically know when we have good estimates for a subset. The correlation between items in systematic threshold sampling prevents us from providing good confidence intervals, so even if systematic threshold sampling gives better variance on the average, we have no way of knowing if we get these good estimates for a concrete subset.

In all our experiments, there is only one case where systematic threshold sampling has a lead over threshold and priority sampling, and that is a marginal lead for the dns traffic which is the most numerous traffic class (see Figure 3).

6. RESERVOIR SAMPLING

In this section, we will discuss and compare how the different sampling schemes can be used for reservoir sampling from a stream of weighted items as described in Section 1.2. If we have seen items $0, ..., n-1$, we should have a sample $S \subseteq [n]$. The individual samples are denoted $S[0], .., S[k-1]$. Our solution for threshold sampling (THR) is a small result of independent interest.

U−R. Reservoir uniform sampling goes back to Fan et al. [1962]. Let $S^{U-R} \subseteq [n]$ be the current sample. While $n \le k$, we have $S[i] = i$ for $i = 0, ..., n-1$. When item $n > k$ arrives, we pick a random number $j \in [n+1]$. If $j < k$, we set $S^{U-R}[j] := n$. Finally, we set $n := n+1$. All this takes constant time for each item.

We note that the weight estimates are only maintained implicitly via n. If $j \in S^{U-R}$, then $\widehat{w}_j = \frac{n}{k} w_j$ where n is the current number of items.

A faster solution by Vitter [1985] does not consider all items, but generates directly the random number of items to be skipped before reaching an item to be included in the sample. However, in our context it is OK to spend a little time on each item streaming by.

P+R. A reservoir version of probability proportional to size sampling with replacement was studied by Chaudhuri et al.[1999]. Besides maintaining a sample $S^{P+R} \subseteq [n]$, we maintain the total current weight $W = \sum_{i\in[n]} w_i$. When item n arrives, for $j = 0, ..., k-1$, we pick a random number $\alpha \in (0,1]$. If $\alpha \le \frac{w_n}{W+w_n}$, we set $S^{P+R}[j] := n$. When done with all samples, we set $W := W + w_n$. Note that if we had $w_n \geq W$, we would expect to change at least half the samples, so for exponentially increasing weight sequences, we spend $\Theta(k)$ time on each item. In [Chaudhuri et al. 1999], it is claimed that their algorithm spends constant time on each item, but that is only true if k is viewed as a constant. In our Internet application, we typically use $k > 1,000$.

Using the current value of the total weight W, we can compute the weight estimates of the sampled items as described in Section 3.

PRI. Priority sampling is trivially implemented using a standard priority queue (see, e.g., [Cormen et al. 2001]). Recall that for each item i, we generate a random number $\alpha_i \in (0,1]$ and a priority $q_i = w_i/\alpha_i$. A priority queue Q maintains the $k+1$ items of highest priority. The k highest form our sample S, and the smallest q_i in Q is our threshold τ.

It is convenient to start filling our priority queue Q with $k+1$ dummy items with weight and priority 0. When a new item arrives we simply place it in Q. Next we remove the item from Q with smallest priority. With a standard comparison based priority queue, we spend $O(\log k)$ on each item, but exploiting a floating point representation, we can get down to $O(\log\log k)$ time for each item [Thorup 2002] (this counts the number of floating point operations, but is independent of the precision of floating point numbers). This is substantially better than the $\Theta(k)$ time we spend on probability proportional to size sampling with replacement, but a bit worse than the constant time spent on uniform sampling without replacement. We shall later show how to get down to constant time if we relax the notion of reservoir sampling a bit.

THR. As a small result of independent interest, we show here how to perform threshold reservoir sampling. In this case, the reservoir will always contain an expected number of k samples from the items seen thus far.

In [Duffield et al. 2005a], the threshold τ^{THR} was determined before items where considered. The threshold was adapted to the traffic to get a desired amount of samples, yet bursts in traffic lead to bursts in the sample. Here, as a new contribution to threshold sampling, we present a reservoir version of threshold sampling which at any time maintains a sample S^{THR} of expected size k.

As items stream by, we generate priorities as in priority sampling. At any point, n is the number of items seen so far. We maintain a threshold τ^{THR} that would give an expected number k of items, that is,

$$\sum_{i\in[n]} \min\{1, w_i/\tau^{THR}\} = k. \tag{16}$$

Also, we maintain the corresponding threshold sample, that is,

$$S^{THR} = \{i \in [n] | q_i > \tau^{THR}\}.$$

The sample S^{THR} is stored in a priority queue. When a new item n arrives it is first added to S^{THR}. Next we have to increase τ^{THR} so as to satisfy (16) with $n' = n+1$. Finally, we remove all the items from S^{THR} with priorities lower than τ^{THR}. Thanks to the priority queue, each such item is extracted in $O(\log k)$ time.

We still have to tell how we compute the threshold. Together with the sample, we store the set L of all items i with weight $w_i \geq \tau^{THR}$. Also, we store the total weight U of all smaller items. We note that the set L is contained in S^{THR}. Now,

$$\sum_{i\in[n]} \min\{1, w_i/\tau^{THR}\} = |L| + U/\tau^{THR}.$$

The items i in L are stored in a priority queue ordered not by priority p_i but by weight w_i. When item n arrives we do as follows. If $w_i \geq \tau^{THR}$, we add i to L; otherwise we add its weight w_n to U.

Next we increase τ^{THR} in an iterative process. Let $\tau^* = U/(k - |L|)$ and let w_j be the smallest weight in L. If L was empty, $w_j = \infty$. If $\tau^* < w_j$, we set $\tau^{THR} = \tau^*$, and we are done. Otherwise, we set $\tau^{THR} = w_j$, remove j from L, add w_j to U, and repeat.

In the above process, each item is inserted and deleted at most once from each priority queue. Also, at any time, the expected size of each priority queue is at most k, so the total expected cost per item is $O(\log k)$. Exploiting a floating point representation of priorities, this can be reduced to $O(\log\log k)$ time. Thus we get the same time complexity as for priority sampling, but with a more complicated algorithm.

SYS. Systematic threshold sampling is unfortunately not useful for reservoir sampling. Recall the definition of the sample. For some threshold τ^{SYS}, we want to pick item i with the same sampling probability $p_i = \min\{1, w_i/\tau^{SYS}\}$, just as in threshold sampling. The sampling decisions are performed by picking a single uniformly random number $x \in [0, 1]$, and include i in S if and only if for some integer j, we have

$$\sum_{h<i} p_i \leq j + x < \sum_{h\leq i} p_i.$$

With threshold sampling we saw how we could maintain the threshold τ^{SYS}, increasing it as new items arrive. However, when we change τ^{SYS}, we change all the sampling probabilities, and typically, this will completely change the set of items picked. For contrast, with threshold sampling, the increased τ^{SYS} would only result in the deletion of some old samples, and never in the inclusion of a previously discarded item.

There is, however, an alternative implementation of systematic threshold sampling which works when the threshold can only be increased by doubling. As in threshold sampling, we maintain a priority queue L over large items i with $w_i > \tau^{SYS}$. The large items are all in the sample. The random number x is now picked in $[0, \tau^{SYS}]$. Considering now the smaller items only, item i is picked if for some integer j,

$$\sum_{h<i} w_j \leq j\tau^{SYS} + x < \sum_{h\leq i} w_i.$$

This leads to the same sampling probabilities as above. For an efficient implementation, we maintain the sum W^S of all the small items. When a new item i arrives, if $w_i \geq \tau^{SYS}$, it is put in the priority queue of large items. Otherwise, we sample i if for some integer j, we $W^S \leq j\tau^{SYS} + x < W^S + w_i$. Afterward, we increment W^S by w_i. If we end up with $k+1$ samples, we double the threshold τ^{SYS}.

When we double τ^{SYS}, with probability 1/2 we add the old value of τ^{SYS} to x. Now x is a uniform random number in the doubled interval $[0, \tau^{SYS}]$. The effect is to keep every other small item, starting with the first item if τ^{SYS} was not added. In addition, there may be items in the priority queue that are now small, and these have to be appended to the list of small items and sampled as described above. The amortized cost per item is that of a priority queue operation, hence the same as for priority sampling, except that for priority sampling it was worst-case.

Note that when we double τ, we may halve the sample size, e.g., if all items are small. Hence we are not fully utilizing the capacity of the reservoir for k samples.

Discussion. Above we showed how the different sampling schemes could be used for reservoir sampling. Uniform sampling was the easiest to implement, using only constant time in the worst case. Second came priority sampling using $O(\log k)$ time with a simple priority queue. Using more complicated algorithms, we could also get this time bound for threshold sampling and systematic threshold sampling, though only expected for threshold sampling and only amortized for systematic threshold sampling. From the view point of time complexity, probability proportional to size was worst, spending $O(k)$ time on each item.

Another problem for threshold and systematic threshold sampling is that they do not make good use of a fixed size reservoir. Threshold sampling only gives an expected number of samples, and systematic threshold sampling can only control the sample size within a factor 2. In all the analysis of the previous sections, we considered the off-line version of systematic threshold sampling which picked exactly k items. Priority sampling with k samples performed very similarly, and would outperform systematic threshold sampling with $k/2$ items.

Relaxed reservoir sampling. Before leaving reservoir sampling, we note that the processing time of an item can be reduced to a constant if we temporarily allow for a doubling of the number of samples in the reservoir. In priority sampling, suppose we during the processing have space for $2k + 2$ items. Instead of using a priority queue, we use a buffer B for up to $2k+2$ items. The buffer is guaranteed to contain the $k + 1$ items of highest priority. New items are placed in the buffer. When the buffer gets full, a cleanup is performed to reduce the occupancy to $k + 1$. Using a standard selection algorithm [Cormen et al. 2001], we find the $(k + 1)^{\mathrm{st}}$ highest priority in B, and all items of lower priority are deleted, all in $O(k)$ time. The cleaning is executed once for every $k + 1$ arrivals, hence at constant cost $O(1)$ per item processed. Using two buffers in tandem, cleaning one while filling the other, we can process each item in constant worst-case time. From the buffers we can generate a priority sample of size k in $O(k)$ time. Similar ideas can be applied to the other sampling schemes.

7. CONCLUDING REMARKS

We have introduced priority sampling as a new simple scheme for weight sensitive sampling for later estimation of subset sums. The scheme is particularly well suited for reservoir sampling in a high volume streaming context. Priority sampling is special in that all item estimates are correlated via the $(k + 1)^{st}$ priority. Despite the correlations, we proved that priority sampling provides unbiased item estimates with no covariance. This combination is not achieved by any previous scheme.

Next we evaluated priority sampling experimentally, finding that it in quality outperformed traditional schemes like uniform sampling and probability proportional to size sampling with replacement. In quality priority sampling matched threshold sampling and systematic threshold sampling which are viewed as idealistic benchmarks, not really applicable for reservoir sampling with fixed capacity. To further understand the phenomena we presented an analytic comparison of the different schemes on some simple cases.

We then embarked on a study of the variance optimality of the different schemes. We focused on the total variance measured as the sum of individual item variances. This measure was particularly relevant for threshold and priority sampling which both have zero covariance. We saw that threshold sampling was optimal if one is satisfied with an expected number of k samples, and conjectured that priority sampling did as well given exactly $k + 1$ samples.

Finally, we made a more detailed comparison of the performance of the different sampling schemes in the context of reservoir sampling from a stream.

The above has already formed the basis of several follow-up works. In particular, our conjecture is now settled by Szegedy [2006], implying that priority sampling given one extra sample beats any other sampling scheme on any possible input. This is of course only for the above total variance measure. However, Szegedy and Thorup [2007] showed that this implies near-optimality with respect to the average variance over all subsets of a given size, as long as this size is not too close to the total set size.

Thorup has shown that we can generate confidence intervals from a priority sample, complimenting the subset sum estimates with likely upper and lower bounds qualifying the precision of the estimates [Thorup 2006]. No such confidence bounds can be given for systematic threshold sampling even though its estimates may be better on the average for large enough subsets.

Duffield et al. [2005b] have shown how to combine different priority estimates of the same value, based on estimates of the variance of each estimate, so as to minimize the overall variance.

Moving in a quite different direction, Alon et al. [2005] have considered how one can preprocess a large data base so that one can later generate a priority sample from any selected subset. Note that this is the opposite order than the one considered in this paper where we first generate a sample, and later are told what subset sum to estimate from this sample. Also in this data base application did priority sampling outperform previous work from [Hellerstein et al. 1997].

A fundamentally new feature of priority sampling is that the $(k+1)^{st}$ priority as is used in determining the estimate of any sampled item. This is very different from traditional Horvitz-Thompson estimators where the estimate of a sampled item is always the same; namely the original weight divided by the sampling probability for that item. However, the proof that the priority estimates are unbiased conditions on the k^{th} priority among the remaining items, and subject to this condition, the estimates can be viewed as Horvitz-Thompson estimators. Cohen and Kaplan [2007] and Cohen et al. [2007] have generalized this basic idea and used it to create other types of sampling and estimation schemes with different kinds of priorities which they call ranks.

A direction not explored is that of pseudo-random permutations used in connection with unweighted estimation of, say, set intersections [Cohen 1997]. For weighted items i, we could let $\alpha_i \in (0.1)$ be a pseudo-random function of i. We would then get a pseudo-random weight sensitive permutation from the ordering of the priorities $q_i = w_i/\alpha_i$.

Summing up, we have introduced priority sampling as a new simpler and better scheme for weight sensitive sampling of k items for later estimation of subset sums. Besides its immediate usefulness for reservoir sampling, priority sampling has already turned into the basis of a rich area of research branching off in multiple directions from the work presented in this paper.

Acknowledgment

We would like to thank a referee from *J. ACM* for thorough refereeing with concise suggestions for clear improvements, including the fix of several typoes in the formulas.

REFERENCES

Adler, R., Feldman, R., and Taqqu, M. 1998. *A Practical Guide to Heavy Tails*. Birkhauser.

Alon, N., Duffield, N., Lund, C., and Thorup, M. 2005. Estimating arbitrary subset sums with few probes. In *Proc. 24th ACM Symp. on Principles of Database Systems (PODS)*. 317–325.

Arnold, B. and Balakrishnan, N. 1988. *Relations, Bounds and Approximations for Order Statistics, Lecture Notes in Statistics, vol. 53*. Springer, New York.

Brewer, K. and Hanif, M. 1983. *Sampling With Unequal Probabilities, Lecture Notes in Statistics, vol. 15*. Springer, New York.

Chaudhuri, S., Motwani, R., and Narasayya, V. 1999. On random sampling over joins. In *Proc. ACM SIGMOD Conference*. 263–274.

Cohen, E. 1997. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. Syst. Sci. 55,* 3, 441–453.

Cohen, E., Duffield, N., Kaplan, H., Lund, C., and Thorup, M. 2007. Sketching unaggregated data streams for subpopulation-size queries. In *Proc. 26th ACM Symp. on Principles of Database Systems (PODS)*. 253–262.

Cohen, E. and Kaplan, H. 2007. Bottom-k sketches: Better and more efficient estimation of aggregates (poster). In *Proc. ACM IFIP Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance)*. 353–354.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. 2001. *Introduction to algorithms*, 2nd ed. MIT Press, McGraw-Hill. ISBN 0-262-03293-7, 0-07-013151-1.

David, H. 1981. *Order Statistics*, 2 ed. Wiley, New York.

Duffield, N., Lund, C., and Thorup, M. 2004. Flow sampling under hard resource constraints. In *Proc. ACM IFIP Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance)*. 85–96.

Duffield, N., Lund, C., and Thorup, M. 2005a. Learn more, sample less: control of volume and variance in network measurements. *IEEE Transactions on Information Theory 51,* 5, 1756–1775.

Duffield, N., Lund, C., and Thorup, M. 2005b. Optimal combination of sampled network measurements. In *Proc. ACM SIGCOMM Internet Measurement Conference (IMC)*. 91–104.

Duffield, N., Lund, C., and Thorup, M. 2005c. Sampling to estimate arbitrary subset sums. Tech. Rep. cs.DS/0509026, Computing Research Repository (CoRR). `http://arxiv.org/abs/cs.DS/0509026`.

Fan, C., Muller, M., and Rezucha, I. 1962. Development of sampling plans by using sequential (item by item) selection techniques and digital computers. *J. Amer. Stat. Assoc. 57*, 387–402.

Hellerstein, J., Haas, P., and Wang, H. 1997. Online aggregation. In *Proc. ACM SIGMOD Conference*. 171–182.

Johnson, T., Muthukrishnan, S., and Rozenbaum, I. 2005. Sampling algorithms in a stream operator. In *Proc. ACM SIGMOD*. 1–12.

Knuth, D. 1969. *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*. Addison-Wesley.

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., and Weaver, N. 2003. Inside the slammer worm. *IEEE Security and Privacy Magazine 1,* 4, 33–39.

Muthukrishnan, S. 2005. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science 1,* 2.

Park, K., Kim, G., and Crovella, M. 1996. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. 4th IEEE Int. Conf. Network Protocols (ICNP)*.

Särndal, C.-E., Swensson, B., and Wretman, J. 1992. *Model Assisted Survey Sampling*. Springer.

Sunter, A. B. 1977. List sequential sampling with equal or unequal probabilites without replacement. *Applied Statistics 26*, 261–268.

Szegedy, M. 2006. The DLT priority sampling is essentially optimal. In *Proc. 38th ACM Symposium on the Theory of Computing (STOC)*. 150–158.

Szegedy, M. and Thorup, M. 2007. On the variance of subset sum estimation. In *Proc. 15th European Symposium on Algorithms (ESA), LNCS 4698*. 75–86.

Thorup, M. 2002. Equivalence between priority queues and sorting. In *Proc. 43nd IEEE Symposium on Foundations of Computer Science (FOCS)*. 125–134. To appear in *J. ACM*.

Thorup, M. 2006. Confidence intervals for priority sampling. In *Proc. ACM IFIP Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance)*. 252–253.

Vitter, J. 1985. Random sampling with a reservoir. *ACM Trans. Math. Softw. 11,* 1, 37–57.

Received Month Year; revised Month Year; accepted Month Year