OFSS: Skampling for the Flow Size Distribution

Paul Tune
School of Mathematical Sciences
University of Adelaide, Australia
paul.tune@adelaide.edu.au

Darryl Veitch
Melbourne School of Engineering
University of Melbourne, Australia
dveitch@unimelb.edu.au

ABSTRACT

We introduce a new method for flow size estimation, the *Optimised Flow Sampled Sketch*, which combines the optimal properties of Flow Sampling with the computational advantages of a counter array sketch. Using Fisher Information as a definitive basis of comparison, we show that it is superior to alternatives in both model and traffic based comparisons.

Categories and Subject Descriptors

C.2.3 [**Computer Communications Networks**]: Network operations—*Network monitoring*; G.3 [**Mathematics of Computing**]: Probability and statistics—*Probabilistic algorithms*

Keywords

Fisher information; flow size distribution; network measurement; OFSS; skampling

1. INTRODUCTION

The distribution of *flow size*, the number of packets in a flow, is an important metric for numerous applications including traffic modelling, management, and attack detection. In resource constrained environments such as within core Internet routers however, accurate measurement of flow size can be challenging.

The traditional approach to fast approximate measurement has been traffic *sampling*. For example flow size estimation within Cisco's *Netflow* [1] is based on a simple pseudo-random per-packet sampling. In our prior work [6, 10, 11] we examined available sampling techniques in depth and showed that *flow sampling* (FS), where the random sampling decision is made directly on flows, has optimal properties. This conclusion was based on a rigorous evaluation, in terms of Fisher Information, of the inherent abilities of the sampling methods as information collectors. Unfortunately, FS requires a *flow table* supporting collision resolution, which incurs costs in memory (entries must store a flow key in addition to a counter), and in processing and latency (key comparison needed on all insertions). Such costs translate to bottlenecks, for example 'heavy hitter' flows generating unsustainable insertion rates.

Sketches are compact data structures with fast update rules that are used to approximately measure properties over data streams. Sketches have been proposed to measure many metrics including set membership [2], entropy [14], heavy hitters [4], and flow sizes [7].

In this paper we present a hybrid *skampling* approach to data collection for flow size measurement, called the OFSS or *Optimised Flow Sampled Sketch*. The FSS consists of a front-end sampling component, namely FS, feeding a sketching component 'Sk', the counting array sketch of [7]. It includes FS and Sk as special cases. The OFSS is an FSS tuned according to an optimal tradeoff between the information gathering strengths and weaknesses of the FS and Sk components. The result is FS-like statistical performance without flow table costs and bottlenecks. In designing OFSS we build on and extend the Fisher analysis of FS and Sk presented in [12].

It is natural, since OFSS is a 'Sample then Sketch' method, to compare against 'Sketch then Sample'. We accordingly compare against the 'Sketch Guided Sampling' approach of Kumar et al. [8] (SGS). We find SGS to have much worse performance than OFSS, both in terms of information gathering, and implementation. We also compare against an enhanced counter array based method of Ribeiro et al. [9] which also, in an implicit sense, incorporates sampling. We show that this *Eviction Sketch* (ESk) underperforms OFSS by a wide margin at meaningful sampling rates unless it is calibrated very differently than in [9], resulting in considerably higher memory use. For each of SGS and ESk we provide the first Fisher-based analysis.

The goal of this paper is to introduce the OFSS scheme, to give its key properties, and to demonstrate its effectiveness. To that end, we forgo proofs (these will appear elsewhere [13]), and focus on providing the underlying intuition. We do not provide a detailed implementation model, however with only a single hash and a single counter array, OFSS is very simple, cheap enough to implement alongside existing packet sampling based systems. Software will be made publically available to facilitate use of the method, in particular for the calculation of the optimal parameter value p_f^* of OFSS.

The paper is organised as follows. Section 2 gives background on the Fisher Information framework, and defines, recalls and derives key results for FS and Sk, SGS and ESk. Section 3 defines FSS then OFSS and describes their key properties. Section 4 gives the statistical performance com-

IMC'14, November 5–7, 2014, Vancouver, BC, Canada.

ACM 978-1-4503-3213-2/14/11 ...$15.00.
http://dx.doi.org/10.1145/2663716.2663732.

parisons on both model based and real data. We conclude in Section 5.

2. BACKGROUND

We assume that a unique flow key can be extracted from each incoming packet, and that the first packet of a flow (and hence the number of flows) can be identified (e.g. SYN packet in the case of TCP flows).

We write (column) vectors in bold lower-case, matrices in bold upper case, $\mathbf{A}^{\mathrm{T}}$ denotes the transpose of $\mathbf{A}$, and $\mathbf{diag}(\mathbf{x})$ an $m \times m$ diagonal matrix whose diagonal entries are taken from the vector $\mathbf{x} \in \mathbb{R}^m$. We denote the set $\{1, 2, \ldots, S\}$ by $[S]$.

2.1 Modelling Framework, Fisher Information

Consider a measurement interval of duration T containing N_f flows. Of these flows, M_k have size k packets, $1 \le k \le W$, where $W < \infty$ is the maximum flow size. The average flow size is $D = \sum_{k=1}^{W} kM_k/N_f$.

The flow size distribution, the unknown vector parameter we seek information on, is $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_W]^T$ where $\theta_k = M_k/N_f$, and obeys

$$0 < \theta_k < 1 \ , \ k \in [W], \quad \sum_{k=1}^{W} \theta_k = 1 \ . \qquad (1)$$

This is a deterministic model of the data over the measurement interval: randomness enters later through the action of the measurement method itself.

Any estimator of $\boldsymbol{\theta}$ is based on an underlying observable which summarises the traffic, where it here takes the form of a packet count random variable C. The discrete density $c_j(\boldsymbol{\theta})$ of C, $j \ge 0$, depends on the details of the summary method. Viewed as a function of $\boldsymbol{\theta}$ for a fixed value j of the observed data, it is known as the *likelihood*: $f(j, \boldsymbol{\theta}) = c_j(\boldsymbol{\theta})$.

The *Fisher Information* (FIM) is a measure of the information the observable holds about the unknown parameters. The unconstrained FIM is defined as

$$\begin{aligned}\mathbf{J}(\boldsymbol{\theta}) &= \mathbb{E}[(\nabla_{\boldsymbol{\theta}} \log f(j;\boldsymbol{\theta}))(\nabla_{\boldsymbol{\theta}} \log f(j;\boldsymbol{\theta}))^{\mathrm{T}}] \\ &= \sum_{j\ge 0}(\nabla_{\boldsymbol{\theta}} \log f(j;\boldsymbol{\theta}))(\nabla_{\boldsymbol{\theta}} \log f(j;\boldsymbol{\theta}))^{\mathrm{T}} c_j. \qquad (2)\end{aligned}$$

The importance of $\mathbf{J}$ lies in the fact that $\mathbf{J}^{-1}$ is the *Cramér–Rao Lower Bound* (CRLB), which lower bounds the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ of **any** unbiased estimator of $\boldsymbol{\theta}$, i.e. $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} \ge \mathbf{J}^{-1}$ in the positive semidefinite sense.

The constraints on $\boldsymbol{\theta}$ in (1) increase available information. The constrained FIM is given in [5] by

$$\boldsymbol{\mathcal{I}}^{+} = \mathbf{J}^{-1} - \mathbf{J}^{-1}\mathbf{G}\left(\mathbf{G}^{\mathrm{T}}\mathbf{J}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{J}^{-1}. \qquad (3)$$

Here the constraint *gradient* matrix is $\mathbf{G}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}(\mathbf{1}_W^{\mathrm{T}}\boldsymbol{\theta} - 1) = \mathbf{1}_W$, where $\mathbf{1}_W$ is a $W \times 1$ vector of ones.

Of chief interest are the diagonal entries of $\boldsymbol{\mathcal{I}}^{+}$, since $\mathrm{Var}(\hat{\theta}_k) \ge (\boldsymbol{\mathcal{I}}^{+})_{kk}$ for any unbiased estimator. Comparison of these between methods corresponds to comparing the best performance the schemes are capable of supporting, thus reflecting their *comparative efficiency in extracting information from the traffic stream.*

The methods we study involve multiple identically distributed counters, which are approximately independent if N_f is large. The Fisher information arising from the entire measurement interval is then just that of a single counter C multiplied by the number of counters.

2.2 Four Methods and their FIMs

We define four existing methods and describe their FIM and $\boldsymbol{\mathcal{I}}^{+}$ matrices. FS is the statistical method of choice, and Sk a canonical implementation approach. SGS and ESk are alternative skampling methods.

Flow Sampling (FS)
In flow sampling flows are sampled (dropped) independently with probability p_f (resp. q_f). Sampling a flow means that each packet within it is sampled, or none.

Here C represents the size of a randomly selected (typical) flow. For a flow of size k, its density is given by $c_0 = q_f$, $c_k = p_f$, and $c_i = 0$ for all $i \ne \{0, k\}$. From [10, 11], an explicit expression for per-flow $\mathbf{J}$ is given by

$$\mathbf{J}_{\mathrm{FS}} = p_f\,\mathbf{diag}(\theta_1^{-1}, \theta_2^{-1}, \ldots, \theta_W^{-1}) + q_f\mathbf{1}_W\mathbf{1}_W^{\mathrm{T}}, \qquad (4)$$

and $\boldsymbol{\mathcal{I}}^{+} = \mathbf{J}^{-1} - \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}} = \frac{1}{p_f}\left(\mathbf{diag}(\boldsymbol{\theta}) - \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}\right)$, and so

$$(\boldsymbol{\mathcal{I}}_{\mathrm{FS}}^{+})_{kk} = \frac{\theta_k(1-\theta_k)}{p_f} \ , \ k \in [W]. \qquad (5)$$

The great strength of FS compared to other methods is that estimators based on it do not have to try to correct distortions: collected flows are perfect.

Counter Sketch (Sk)
The counter sketch, introduced by Kumar et al. [7], consists of an array of A packet counters, each initialised to zero at the beginning of the measurement interval. Incoming packets are mapped independently and uniformly over the counters, using a hash function mapping a flow key, so that each packet in a flow maps to the same counter, but collisions can occur so that a given counter may sum the packet counts from two or more flows. We define $\alpha = N_f/A$ to be the *flow load factor*, the average number of flows per counter.

Here C is the final packet count in a *typical counter*, at the end of the measurement interval when all N_f flows are in the sketch. By using a Poisson assumption for C, the density can be shown to be (see [13])

$$c_j = \left(\sum_{\mathbf{x}\in\Omega_j}\prod_{k=1}^{W}\frac{\lambda_k^{x_k}}{x_k!}\right)e^{-\sum_{k=1}^{W}\lambda_k}, \quad j \ge 0 \qquad (6)$$

where x_k is the number of flows of size k in the counter, $\lambda_k = M_k/A = \alpha\theta_k$, $\mathbf{x} = [x_1, x_2, \cdots, x_W]$ denotes a *collision pattern*, and Ω_j is the set of flow collision patterns with packet count $C = \sum_{k=1}^{W} kx_k = j$. It follows that the per-counter Fisher matrix is given by

$$(\mathbf{J})_{ik} \ = \ \alpha^2\Big(-1 + \sum_{j=\max(i,k)}^{\infty}(c_{j-i}c_{j-k})/c_j\Big), \qquad (7)$$

and the per-flow information is $\mathbf{J}_{\mathrm{Sk}} = A\mathbf{J}/N_f = \mathbf{J}/\alpha$. The constrained per-flow CRLB, $\boldsymbol{\mathcal{I}}_{\mathrm{Sk}}^{+}$, is given by substituting $\mathbf{J}_{\mathrm{Sk}}$ into (3).

The strength of Sk is that all packets of all flows are efficiently collected, but as the sketch becomes increasingly full as α increases, we expect information-destroying flow collisions to become severe.

Sketch Guided Sampling (SGS)
Traditional *Packet Sampling* (PS) simply samples packets independently with fixed probability p, and is known to destroy information about $\boldsymbol{\theta}$ with remarkable efficiency [10,

11]. Sketch Guided Sampling, introduced in [8], is at heart a generalised packet sampling method indexed by a sampling probability function $p(k)$, $k \in [W]$. A packet will be sampled with probability $p(k)$ if it is the k-th packet encountered so far in its flow. Of course, we do not know the in-flow position of packets, if one did $\boldsymbol{\theta}$ would be already known! The innovation in [8] was to estimate this quantity by a coarse counter sketch, hence SGS is a skampling method.

To simplify comparisons, we assist SGS by replacing its sketch component with oracular knowledge of each packet's in-flow position. Assisted SGS is then a pure sampling method, whose per-flow counter density is characterised by its *sampling matrix* $\mathbf{B}$ as

$$c_j = \sum_{k=1}^{W} b_{jk}\theta_k, \quad 0 \le j \le W, \tag{8}$$

where b_{jk}, is the probability that if the original flow had k packets, only j remain after sampling. The matrix element b_{jk} can be calculated via the recursion $b_{j,k+1} = p(k+1)b_{j-1,k} + q(k+1)b_{j,k}$.

From general sampling results from [11], we can write

$$\mathbf{J} = \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}, \tag{9}$$

where $\mathbf{D} = \mathbf{diag}(\mathbf{B}\boldsymbol{\theta})$, and $\boldsymbol{\mathcal{I}}^{+} = \mathbf{J}^{-1} - \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}$.
Following [8], we use

$$p(k) = p(k;\beta,\epsilon) = \frac{1}{1+\epsilon^2 k^{(2\beta-1)}}, \ 1/2 \le \beta \le 1, \tag{10}$$

a monotonically decreasing function of k. The idea is that this creates a bias toward short flows which counteracts the strong large-flow bias of traditional PS, which corresponds to $\beta = 1/2$.

Note that (true) SGS requires both a flow table and a sketch, and so is more costly to implement than FS.

Eviction Sketch (ESk)
The data collection method proposed in [9] physically enhances Sk by associating to each of its A counters an *ownership* variable taking values in $[L]$, all initialised to L. An incoming packet is given a random priority class $\ell \in [L]$. If ℓ equals the value of the associated ownership variable its counter is incremented as normal, but if it is larger (lower priority than it) the counter is unchanged, and if smaller (higher priority) the counter is reset to 1, and the ownership variable is set to ℓ recording the fact that class ℓ now 'owns' the counter. This random eviction of lower class packets is an implicit packet sampling, reducing the input load α to an effective load $\alpha' < \alpha$, hence ESk is a skampling method.

It is not hard to show that the counter density is

$$c_0 \;=\; e^{-\alpha}, \tag{11}$$
$$c_j \;=\; F(L)\, c_{\mathrm{Sk},j}(\alpha/L), \quad j \ge 0 \tag{12}$$

where $F(L) = (1-e^{-\alpha})/(1-e^{-\alpha/L})$ and $c_{\mathrm{Sk},j}$ is the density (6) with load α/L. It follows that the per-counter Fisher matrix based on C is given by

$$(\mathbf{J})_{ik} \;=\; \frac{\alpha^2}{L^2}\Big(-1 + \sum_{j=\max(i,k)}^{\infty}(c_{j-i}c_{j-k})/c_j\Big), \tag{13}$$

and the per-flow information is $\mathbf{J}_{\mathrm{ESk}} = A\mathbf{J}/N_f = \mathbf{J}/\alpha$. The constrained per-flow CRLB, $\boldsymbol{\mathcal{I}}^{+}_{\mathrm{ESk}}$, is given by substituting $\mathbf{J}_{\mathrm{ESk}}$ into (3).

The degree of sampling offered by ESk, $p = \alpha'/\alpha = F(L)/L$, is only coarsely controllable via the choice of L, and is limited: p hits a limit of $1/L$ for $\alpha \gg L$. For a given α and k, the CRLB of θ_k is minimized at $L = \infty$. Here ESk reduces to FS with $\min(A, N_f)$ sampled flows, however this requires infinite memory. For $L = 1$ ESk reduces to Sk, a pure sketch with $p = 1$.

3. FSS AND OFSS

We define the *Flow Sampling Sketch* (FSS), a family of skampling methods indexed by an internal parameter p_f, describe its properties and derive its Fisher Information. We then introduce the *Optimised Flow Sampling Sketch* (OFSS), which is FSS with p_f chosen to maximise its Fisher Information gathering ability.

3.1 FSS

We begin by contrasting the nature of FS and Sk. The central issue is that of collision resolution. Allowing multiple flows to map to the same counter is the origin of the high speed implementation advantages of Sk, but also of its statistical weakness: collisions create complex ambiguities which literally destroy information (see below). In contrast, the information loss in FS is due solely to the fact that not all flows are used, but the requirement of strict collision resolution makes FS, for any p_f, computationally onerous at high speed.

The key idea of FSS is the following: if α could be kept low, then Sk could work at an 'operating point' with low information loss. A natural way to achieve this is to perform a thinning of the incoming flows to reduce the input α to an effective $\alpha' < \alpha$ seen by the sketch, set to a suitable value which trades off the need for low ambiguity (α' not too big), against adequate data collection as well as not under-utilising expensive memory (α' not too small). It is useful to keep in mind that when $\alpha' = 1$, a proportion $1 - c_0 = 1 - e^{-1} \approx 0.632$ of counters are used.

Definition: FSS(p_f;A) *Each flow is sampled with probability p_f. Packets of sampled flows are inserted into a counter sketch with A counters.*

Thus FSS has an 'FS front end' which sends sampled flows through to an unmodified counter sketch, which sees a reduced *effective* average flow load of

$$\alpha' = p_f\alpha.$$

In information terms, FSS combines the loss of information arising from FS's flow thinning, with a loss due to Sk's collision-induced flow ambiguity.

In terms of implementation, the key advantage of FSS is that the FS component does not require any flow table or collision resolution. Only the flow selection itself remains, but this can be easily implemented in deterministic time on a per-packet basis by hashing flow keys. As described in [12], the hash function can be simultaneously used to index into the sketch.

From the above, we expect FSS to behave exactly as Sk with load α'. However, whereas the number of flows N_f entering FSS is a known constant, the number reaching the counter array is, unlike for Sk, a random variable. Remarkably, it turns out (see [13]) that the counter density nonetheless takes the same form as for Sk (Equation (6)), with the λ_k replaced by $\lambda'_k = \alpha'\theta_k = p_f\lambda_k$. Since the dependence on $\boldsymbol{\theta}$

is unchanged, the per-counter information for FSS is just as for Sk with parameter α', and therefore the information per-incoming-flow is $\mathbf{J}_{\text{FSS}}(\alpha) = p_f \mathbf{J}_{\text{Sk}}(\alpha')$. The per-flow CRLB is therefore

$$\boldsymbol{\mathcal{I}}^+_{\text{FSS}}(\alpha) \quad = \quad \frac{\alpha}{\alpha'}\boldsymbol{\mathcal{I}}^+_{\text{Sk}}(\alpha') = \frac{1}{p_f}\boldsymbol{\mathcal{I}}^+_{\text{Sk}}(\alpha'). \qquad (14)$$

FSS inherits FS's flexibility in adapting to varying traffic levels. If α suddenly increases then p_f could be simply decreased to keep α' in a target range.

3.2 OFSS

The parameters considered fixed in a given measurement are the counter array size A, the number of flows N_f, and $\boldsymbol{\theta}$. The input load to the FSS is therefore fixed at $\alpha = N_f/A$, whereas p_f is an internal parameter free to be tuned to any value in $[0, 1]$.

In defining OFSS our aim is to select p_f to maximise the **total** amount of information captured by FSS, or equivalently, to minimise the CRLB when using all A counters. Thus, for each $\alpha > 0$, we seek to minimise the diagonal elements of

$$\boldsymbol{\mathcal{I}}_T(p_f) = \frac{\boldsymbol{\mathcal{I}}^+_{\text{FSS}}(\alpha)}{N_f} = \frac{\boldsymbol{\mathcal{I}}^+_{\text{FSS}}(\alpha)}{A\alpha} = \frac{1}{A\alpha'}\boldsymbol{\mathcal{I}}^+_{\text{Sk}}(\alpha') \qquad (15)$$

over $p_f \in [0, 1]$, or equivalently, $\alpha' \in [0, \alpha]$.

Equation (15) is a product of two competing terms, each a function of p_f through $\alpha' = p_f\alpha$. The scalar $1/(A\alpha')$ depends on the number of flows delivered to the sketch by the FS component and is monotonically **decreasing** with p_f. The matrix $\boldsymbol{\mathcal{I}}^+_{\text{Sk}}(\alpha')$ corresponds to the information stored per-flow in the counter array. It is monotonically **increasing** with p_f in a positive semidefinite sense [13]. The key question is whether an optimal point exists for any α, that is, if there is a maximum amount of information that can be stored in the sketch, so that excessive load is not just a matter of diminishing returns but actually results in information destruction. To answer it, let $I_k(x) = (\boldsymbol{\mathcal{I}}^+_{\text{Sk}}(x)/x)_{kk}$. From (15), $(\boldsymbol{\mathcal{I}}_T(p_f))_{kk} = I_k(\alpha')/A$.

THEOREM 1. *For each k, $I_k(x)$ has a global minimum at finite $x = \alpha^{**}_k > 0$ for any $\boldsymbol{\theta}$ with $W > 3$.*

The proof is based on the fact that the largest eigenvalue of $\boldsymbol{\mathcal{I}}^+_{\text{Sk}}(x)$ is $O(x^{W-2})$, which defeats the $1/x$ factor for large x if $W > 3$ [13].

The existence of the global minimum implies the existence of a first local minimum $\alpha^*_k > 0$ obeying $\alpha^*_k \leq \alpha^{**}_k$. We are now ready to define OFSS.

Definition: OFSS($k; A$) *OFSS($k; A$) is FSS($p^*_f; A$), where $p^*_f(k;\alpha) \in (0, 1]$ is the value that minimises $(\mathcal{I}_T(p_f))_{kk}$, subject to $\alpha' = \alpha p^*_f \leq \alpha^*_k$.*

Apart from the trivial case of $W = 2$, we do not know of any examples where the first local minima is not equal to the global minimum, and conjecture that $\alpha^*_k = \alpha^{**}_k$ in all cases with $W > 3$.

Even if there are cases when α^*_k is not optimal, it is advantageous to base a definition on it because

$$p^*_f(k;\alpha) = \begin{cases} 1, & \alpha < \alpha^*_k, \\ \frac{\alpha^*_k}{\alpha}, & \alpha \geq \alpha^*_k. \end{cases} \qquad (16)$$

In other words, if $\alpha > \alpha^*_k$, select p_f to cut back the input so that $\alpha' = \alpha^*_k$, otherwise let all flows in. The significance of this property is that one only needs to calculate α^*_k to know $p^*_f(k;\alpha)$ for all α, a significant practical advantage when adapting $p^*_f(k;\alpha)$ to changing traffic conditions, or different resource allocation (A value). Calculation details for α^*_k can be found in [13].

4. PERFORMANCE COMPARISON

We compare the CRLB performance of OFSS against that of FS, Sk, and enhanced versions of our skampling competitors SGS and ESk. For SGS(β, ϵ), we use the assisted form as described above, and also test using $\beta = 0.75$ in addition to the $\beta = 1$ value used in [8]. For ESk(L), we compare not only against the $L = 2$ favoured in [9], but also larger values adapted to α, and ignore the additional memory required to maintain the ownership variables. In data estimation comparisons, we use ML estimators for all methods. Previously a sub-optimal estimator was used to test SGS and ESk.

We compare on a basis of equal counter memory. The methods Sk, FSS, OFSS, ESk are each given an array with A counters, whereas the sampling methods FS, SGS are given flow tables with A entries. This means that if $N_f > A$, only $N'_f = \min(N_f, A)$ flows will actually be delivered to the methods (we ignore flow expiry).

With A and N_f (or equivalently $\alpha = N_f/A$) fixed, methods Sk, ESk(2) and FS(p_f=N'_f/N_f =$\min(1, 1/\alpha)$) are determined, as are OFSS(k) given the target θ_k. For SGS, for each of $\beta = \{0.75, 1\}$ we set ϵ to match the average number of captured packets to that of FS. For ESk, at larger α values we select two values of $L > 2$ to bracket OFSS (at the chosen k^*). Finally, we also compare against FS(1), that is perfect flow collection.

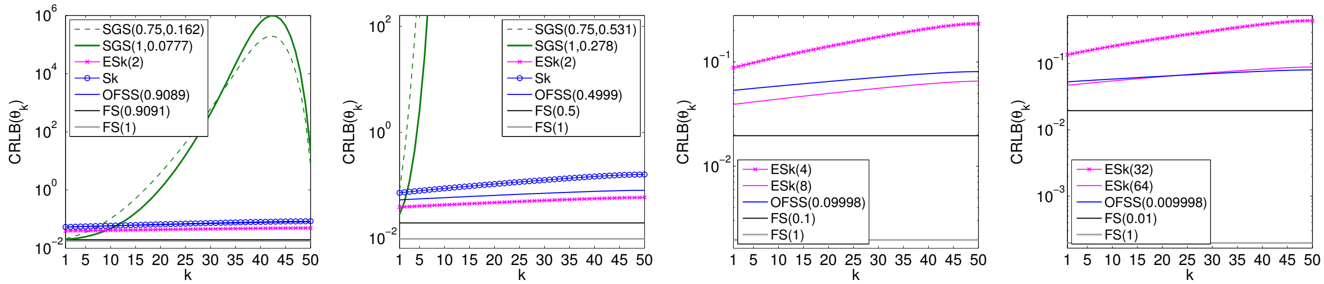
We compare over an operating range of $1 < \alpha \leq 100$, since $\alpha \approx 1$ is theoretically important, and $\alpha = 100$ corresponds to a commonly quoted sampling probability of 0.01. Values as high as $\alpha = 10000$ are now typical in big data contexts. The structure of OFSS is such that its performance will be unaffected by higher loads, whereas other methods suffer in information or implementation terms, or both, as load increases.

Since A enters in only as a multiplicative factor, we plot per-counter variance rather than total variance. Here the objective is to compare methods, the absolute CRLB values depend on θ and are not important.

4.1 CRLB Comparisons

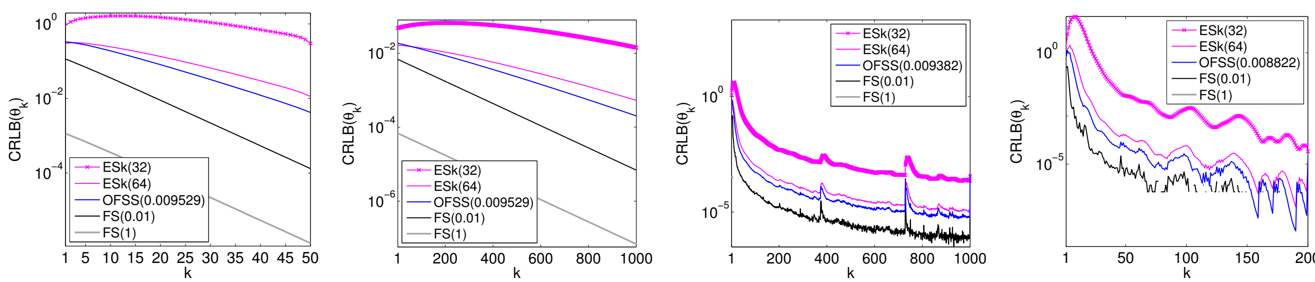
We evaluate $(\boldsymbol{\mathcal{I}}_T(p_f))_{kk}$ numerically from the exact expressions given earlier, initially using a truncated geometric model TG(W, R) for $\boldsymbol{\theta}$, where $R = \theta_1/\theta_W$. The $R = 1$ case is a uniform distribution, which has a heavier tail than any (truncated) Pareto distribution.

Consider Figure 1 where $\boldsymbol{\theta}$ is uniform, W = 50. In plot(a) $\alpha \approx 1$, a load so light that results are close to degenerate: OFSS(p^*_f) $\approx$ FSS(1) = Sk $\approx$ESk(2), FS(p_f) $\approx$ FS(1). Despite this SGS already shows poor behavior. At $\alpha = 2$ it is already orders of magnitude worse than other methods for $k > 3$, and becomes almost impossible to calculate for $\alpha = 100$. This is consistent with the poor behaviour of packet sampling [11], whose structure SGS generalises, but does not fundamentally change. For small α ESk(2) outperforms OFSS, however as α increases both Sk and ESk(2) also have variances orders of magnitude beyond OFSS for all k, and then become difficult to calculate. At $\alpha = 100$ the field therefore narrows to ESk(L) for well chosen $L \gg 2$ versus OFSS. For each of plots (c) and (d) L values are used so that



(a) $\alpha = 1.1$ (b) $\alpha = 2$ (c) $\alpha = 10$ (d) $\alpha = 100$

Figure 1: CRLB Comparisons with uniform $\boldsymbol{\theta}$, $W = 50$. We plot SGS(β, ϵ) when possible; ESk(2) and Sk for $\alpha \leq 2$, else ESk(L) for values bracketing OFSS($p_f^*(1)$), and FS($1/\alpha$). Here $p_f^* \approx 1/\alpha$. since $\alpha_1^* \approx 1$.



(a) $\boldsymbol{\theta}$ from TG(50, 1000) (b) $\boldsymbol{\theta}$ from TG(1000, 1000) (c) $\boldsymbol{\theta}$ from Abilene-III (d) $\boldsymbol{\theta}$ from Leipzig-II

Figure 2: CRLB Comparisons, $\alpha = 100$. [Abilene, Leipzig] truncated at $W = [1000, 200]$.

ESk brackets OFSS. In plot(d) we see that OFSS is within a constant factor of, and the same order of magnitude as, the benchmark FS($1/\alpha$). We see that to (just) defeat OFSS requires ESk with $L = 64$. Assuming a generous 32 bits per counter, this means that $\approx$19% extra memory is needed to achieve this, more for larger α.

Comparing plot(d) with Figure 2(a) shows that similar conclusions hold even after changing the distribution shape quite radically from uniform to TG(50, 1000). For TG(W, R) with $R = \{10, 1000\}$, $W = \{50, 1000\}$, similar results were found. Changing from optimising for θ_1 to other θ_k changes variance values somewhat for OFSS(k) and the matching L slightly for ESk(L), but not the conclusions.

Figure 2 provides a bridge from models to data with load fixed at $\alpha = 100$. From plots(a) to (b) we increase W from $W = 50$ to a more realistic $W = 1000$. The picture is remarkably unchanged. From plot(b) to (c) we move from a very rough traffic model, TG(1000, 1000), to data from the Abilene-III dataset (see Table 1), truncated at $W = 1000$. Again the same model-comparison conclusions hold. Finally, plot(d) uses $\boldsymbol{\theta}$ from the Leipzig-II dataset where $\theta_k = 0$ for many k, resulting in zeros manifesting as gaps in the FS and FS(1) curves. Associated 'spiky' far-tail estimates for OFSS and ESk are a sign of the need for truncation, here we used $W = 200$.

4.2 Estimation Comparisons

We now compare $\hat{\boldsymbol{\theta}}$ estimates, again for $\alpha = 100$, for FS, OFSS, SGS($1, \epsilon$), and ESk(64) using maximum likelihood estimation (see [13] for MLE derivations).

Trace	Link Capacity	N_f	Duration (hh:mm:ss)	D
Leipzig-II	50 Mbps	2,277,052	02:46:01	19.76
Abilene-III	10 Gbps	23,806,285	00:59:49	16.12

Table 1: Summary of the data traces used.

The datasets, summarised in Table 1, are old but adequate for testing the methods. We extract TCP flows according to the standard 5-tuple (with no timeout).

Figure 3 plots $\hat{\boldsymbol{\theta}}$ for Abilene-III, truncated at $W = 2000$, which is approximately the largest value for which $\theta_k > 0$ for all k. The grey curve, FS(1), corresponds to $\boldsymbol{\theta}$ itself. The estimate for SGS: $\hat{\boldsymbol{\theta}}_1 = 1$, and $\hat{\boldsymbol{\theta}}_k = 0$ for all $k > 1$, is as expected very poor, in fact degenerate. All other methods appear to perform quite well, however it is very difficult to assess performance reliably from such plots, in particular

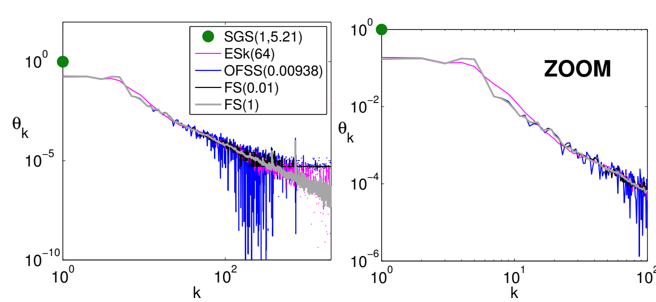


Figure 3: $\hat{\boldsymbol{\theta}}$ for Abilene, $W = 2000$. Here $\alpha = 100$, and OFSS and ESk are optimised for θ_1.

Trace	k^*	p_f^*	L	FS	OFSS	ESk	SGS
Abilene	1	0.009	64	1.6e-3	5.6e-3	7.8e-2	8.9e-1
Leipzig	1	0.009	64	9.4e-2	**2.1e-2**	2.5e-1	**2.8e-1**
Abilene	W	0.005	64	1.6e-3	4.1e-3	7.8e-2	8.9e-1
Leipzig	W	0.013	64	9.4e-2	**2.1e-2**	2.5e-1	**2.5e-1**

Table 2: ℓ_2 error: Abi. (W=2000), Leip. (W=200).

in the far tail because of the high variablility inherent in single point estimates as the data 'runs out'. Smoothing is typically used to improve behaviour in such cases [3].

For a more objective assessment, we employ the ℓ_2 error $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2 = (\sum_{k=1}^{W}(\hat{\theta}_k - \theta_k)^2)^{1/2}$ to summarise performance of each method over all k. Overall, for each trace the results of Table 2 reflect the variance pecking order FS<OFSS<ESk(64)≪SGS from the CRLB analysis averaged over all k, both when optimised for θ_1 and θ_W. Four exceptions are noted in bold. In two of these OFSS is seen to defeat FS. This is due to an implicit smoothing of the sketch component of OFSS improving the fit over the tail where for many indices $\theta_k = 0$ for Leipzig. In the other two cases SGS appears comparable to ESk for Leipzig when $k^* = W$ instead of much worse. This results from the fact that θ_1, θ_2 are fairly large and SGS is able to give rough estimates for them given its deliberate small flow bias, while setting $\hat{\boldsymbol{\theta}}_k = 0$ for all $k > 2$. Note that these exceptions are in part due to the limitations of ℓ_2 as a summary metric. Leipzig-II provides an instructive example of difficulties (for all methods) which arise when the assumption of $\theta_k > 0$ is violated. Abilene-III avoids this by a suitable, and larger, choice of W.

5. CONCLUSION

We have introduced OFSS, a hybrid 'skampling' method for flow size estimation. Its Fisher information gathering ability is of the same order as Flow Sampling (the optimum), but its sketch data structure allows its use in resource constrained applications. It is clearly superior to the alternative SGS and ESk in implementation terms, and in information terms, except for ESk for large enough L. However ESk is more complex and requires supplementary memory which increases both with L and hence with load. OFSS(k^*) is tuned to minimise the CRLB of θ_{k^*}, but for any k^*, it performs well for all θ_k.

Acknowledgment

Partially supported by Australian Research Council's Discovery Projects funding schemes # DP120102834 and # DP110103505.

6. REFERENCES

[1] Cisco NetFlow. http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.

[2] B. H. Bloom. Space-time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[3] N. Duffield, C. Lund, and M. Thorup. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Transactions on Networking*, 13(5):933–946, 2005.

[4] C. Estan and G. Varghese. New directions in traffic measurement and accounting. *ACM Transactions on Computer Systems*, 21(3):270–313, August 2003.

[5] J. D. Gorman and A. O. Hero. Lower bounds for parametric estimation with constraints. *IEEE Trans. Info. Th.*, 36(6):1285–1301, Nov. 1990.

[6] N. Hohn and D. Veitch. Inverting Sampled Traffic. In *Proc. 2003 ACM SIGCOMM Internet Measurement Conference*, pages 222–233, Miami, Oct. 2003.

[7] A. Kumar, M. Sung, J. Xu, and J. Wang. Data streaming algorithms for efficient and accurate estimation of flow size distribution. In *Proceedings of ACM SIGMETRICS / IFIP WG 7.3 Performance 2004*, New York, June 2004.

[8] A. Kumar and J. Xu. Sketch Guided Sampling: Using online estimates of flow size for adaptive data collection. In *Proceedings of IEEE Infocom 2006*, Barcelona, Spain, April 2006.

[9] B. Ribeiro, D. Towsley, and T. Ye. A Resource-minimalist Flow Size Histogram Estimator. In *Proc. ACM/SIGCOMM Internet Measurement Conf.*, pages 285–290, Vouliagmeni, Greece, Oct. 20-22 2008.

[10] P. Tune and D. Veitch. Towards Optimal Sampling for Flow Size Estimation. In *Proc. ACM SIGCOMM Internet Measurement Conf.*, pages 243–256, Vouliagmeni, Greece, Oct. 20-22 2008.

[11] P. Tune and D. Veitch. Fisher Information in Flow Size Distribution Estimation. *IEEE Transactions on Information Theory*, 57(10):7011–7035, Oct. 2011.

[12] P. Tune and D. Veitch. Sampling vs Sketching: An Information Theoretic Comparison. In *IEEE Infocom 2011*, pages 2105–2113, Shanghai, China, April 10-15 2011.

[13] P. Tune and D. Veitch. OFSS: Skampling for the Flow Size Distribution: Technical Report. Technical report, Dept. E&EE, The University of Melbourne, 2014. copy available upon request.

[14] H. Zhao, A. Lall, M. Ogihara, O. Spatscheck, J. Wang, and J. Xu. A data streaming algorithm for estimating entropies of OD flows. In *Proc. 7th ACM SIGCOMM Internet Measurement Conf.*, pages 279–290, New York, NY, USA, 2007. ACM.