Norm, Point, and Distance Estimation Over Multiple Signals Using Max–Stable Distributions

Stilian A. Stoev[†]   Marios Hadjieleftheriou[††]   George Kollios[‡]   Murad S. Taqqu[‡]

[†]University of Michigan, Ann Arbor, [††]AT&T Labs-Research [‡]Boston University

sstoev@umich.edu, marioh@research.att.com, gkollios, taqqu@cs.bu.edu

Abstract

Consider a set of signals $f_s : \{1, \ldots, N\} \to [0, \ldots, M]$ appearing as a stream of tuples $(i, f_s(i))$ in arbitrary order of i and s. We would like to devise one pass approximate algorithms for estimating various functionals on the dominant signal f_{max}, defined as $f_{max} = \{(i, \max_s f_s(i)), \forall i\}$. For example, the "worst case influence" which is the F_1–norm of the dominant signal [7], general F_p–norms, and special types of distances between dominant signals. The only known previous work in this setting are the algorithms of Cormode and Muthukrishnan [7] and Pavan and Tirthapura [18] which can only estimate the F_1–norm over f_{max}. No previous work addressed more general norms or distance estimation. In this work, we use a novel sketch, based on the properties of max–stable distributions, for these more general problems. The max–stable *sketch is a significant improvement over previous alternatives in terms of simplicity of implementation, space requirements, and insertion cost, while providing similar approximation guarantees. To assert our statements, we also conduct an experimental evaluation using real datasets.*

1 Introduction

Consider a network traffic monitoring application. An interesting measure is the maximum possible network utilization which would occur if the transmissions from different source IP addresses were coordinated. This measure is also known as the max–dominance norm and it appears in a variety of other applications as discussed in more depth by Cormode and Muthukrishnan [7]. Essentially, the *max* operation over a large set of input distributions is a natural measure for computing "worst case influence". Previous work on norm estimation has focused mostly on individual distributions, or on processing multiple distributions individually, with space linear to the number of input signals. Here, we study cumulative trends in the presence of multiple distributions, and we present fast approximate algorithms for efficient computation of arbitrary F_p–norms, point and distance computations on the dominant signal of a multiplicity of signals in the data streaming model.

Formally, our setting consists of multiple signals (or data streams) $f_s : \{1, \ldots, N\} \to [0, M]$, $1 \leq s \leq S$ with different distributions, where every signal is defined over a very large domain $[N]$, so that it is not feasible to store it or process it in real time. Each signal f_s can be viewed as a set of items $(i, f_s(i))$, $i \in [N]$, where each i appears only once per signal f_s. Consider the dominant signal defined as $f_{max} = \{(i, \max_s f_s(i)), \forall i\}$. Straightforwardly, one can compute a variety of statistical measures over the dominant signal. For example, the max–dominance norm which is defined as $\sum_i \max_s f_s(i)$ [7], can be viewed as the F_1–norm of the dominant signal. Similarly, the energy of the signal is given by the F_2–norm $(\sum_i \max_s f_s(i)^2)^{1/2}$. Here, we assume that signal values are observed as a stream of tuples $(i, f_s(i))$ arriving in arbitrary order in i and s. We also assume that s is not made known explicitly to the algorithms that we consider (and is used in our notation only for convenience).

For example, in the aforementioned network monitoring application, under our setting the domain of items i corresponds to source IP addresses, while the different signals s correspond to disjoint measurement intervals. The values $f_s(i)$ correspond to the total number of bytes transmitted by IP i within interval s. It is easy to see that computing the F_1–norm of the dominant signal yields the hypothetical worst–case utilization of the network in case of coordinated maximum transmission rates from all known IPs, within the same measurement interval. Computing the F_2–norm of the dominant signal yields the energy of the cumulative distribution which is potentially useful for detecting denial of service attacks. Furthermore, computing distances is essential for change detection applications. Finally, reconstruction of the large values of the signals is useful for identifying the heavy hitters.

To the best of our knowledge, the only previous work for approximate computation of statistical measures on the

dominant signal over a set of input signals has concentrated only on estimating the max–dominance norm — the algorithms by Cormode and Muthukrishnan [7] and Pavan and Tirthapura [18]. A natural way for computing a larger variety of functionals over the dominant signal is by directly using *max–stable* distributions [20]. Here, we use a novel type of synopsis called *max–stable* sketch, which is based on the nice max–stable properties of the Fréchet distribution.

The max–stable sketch can estimate the F_α–norm of the input signals for any $\alpha \in \mathbb{R}^+$, within ϵ error and probability $1 - \delta$ using $O(1/\epsilon^2 \ln 1/\delta \log N/\delta \log M)$ space. In addition, the sketch can be used to recover large values of the dominant signal *exactly*, with high probability. Finally, it can also be used for estimating a special distance between two dominant signals. Notice finally, that other solutions for computing F_α–norms (e.g., AMS sketches [1]) cannot be applied on the max–incremental setting that we consider here. Another advantage of the max–stable sketch is simplicity of implementation, especially in comparison with other alternatives for computing max–dominance norms which require a large number of floating point division and multiplication operations, which might be infeasible on certain hardware, as we will argue in Section 5. Finally, our algorithm has the useful property that it can be applied both in the distributed and the merged stream setting (as defined in [12]) without any modifications.

2 Related Work

A large body of work has appeared recently on designing approximate, probabilistic algorithms for summarizing signals on the data streaming model [17, 8, 6, 13, 16, 10, 11]. These algorithms focus on individual distributions, or on processing multiple distributions individually. In this work, we present algorithms for approximate computation of statistics on the cumulative distribution over a multiplicity of signals.

The idea of using stable distributions has been used successfully in the literature in the past. Cohen [5] used min–stable distributions for estimating subset–sums. Indyk [15], advocated the use of sum–stable distributions for estimating the F_p–norm of a given signal, for $0 < p \le 2$. Cormode and Muthukrishnan [7] exploited the sum–stable distribution for designing approximate algorithms for max–dominance norm estimation over multiple signals. Their algorithm requires $O((1/\epsilon^2 \log M + 1/\epsilon^3 \log n \log \log n) \log 1/\delta)$ space and $O(1/\epsilon^4 \log f_s(i) \log n \log 1/\delta)$ processing time per element, where n is the total number of tuples seen in the combined stream. In comparison, we use algorithms based on max–stable distributions that provide better theoretical bounds, both in terms of space and processing time per item. Also, the Cormode and Muthukrishnan algorithm is max–

dominance norm specific, while the max–stable sketch offers a wider range of features. Finally, their sketch is mostly theoretical in nature, and a viable implementation of this algorithm is problematic in practice, as will be seen in Section 5.

Pavan and Tirthapura [18] recently proposed an efficient algorithm for estimating the F_0–norm of a signal (commonly known as the distinct–count). Their algorithm uses some special properties of the Universal Hash functions of Carter and Wegman [4], to provide a sketch that can handle very efficiently insertions of a range of items $[a, b]$ in $O(\log (b - a))$ time, where $a, b \in \mathbb{N}$. Since the max–dominance norm computation can be reduced to a distinct counting problem using ranges of values, the range–efficient sketch by Pavan and Tirthapura is a natural candidate for max–dominance estimation, providing a solution that uses $O(1/\epsilon^2 (\log N + \log M) \log 1/\delta)$ space, $O(1/\epsilon^2 \log \log N \log f_s(i) \log 1/\delta)$ worst case processing time per element and $O(\log (f_s(i)/\epsilon) \log 1/\delta)$ amortized time. Nevertheless, once again this solution can be used only for estimating the max–dominance norm of a set of signals. In addition, it has a variety of other practical drawbacks which will be discussed in detail in Section 5.

Another work closely related to ours is by Gibbons and Tirthapura [12], in which (ϵ, δ)–approximation algorithms are described for estimating the bitwise or of the union of indicator streams consisting of $0, 1$ elements. These algorithms are then used for approximating the F_0-norm of the union of streams of integer valued domains. Finally, extensions are discussed for handling the scenario studied in [15, 9], in which data items appear as (label, value) pairs like in our setting. The same algorithm can be applied for computing $\sum_i \max \{a_i, b_i\}$, $a_i, b_i \in [0, M]$. However, as pointed out in [7], this algorithm requires $\Omega(M)$ time to process each item (where M can be very large) and therefore this approach is not practical. In addition, here we consider both integer and real valued domains, while the work in [12] is limited to integer valued domains only.

3 The Max–Stable Sketch

We present here the basic theory behind the max–stable sketch. A full analysis and detailed proofs of all theorems appear as a technical report [21].

3.1 Initial Construction

Consider a non–negative signal $f : \{1, \ldots, N\} \to [0, M]$. The α–max-stable sketch of f is defined as:

$$E_j(f) := \max_{1 \le i \le N} f(i) Z_j(i), \ 1 \le j \le K, \qquad (1)$$

where the random variables $Z_j(i)$ are max–stable independent standard α–Fréchet.

A random variable Z is said to be max–stable if, for any $a,\ b > 0$, there exist $c > 0$ and $d \in \mathbb{R}$, such that

$$\max\{aZ', bZ''\} \stackrel{d}{=} cZ + d, \qquad (2)$$

where Z' and Z'' are independent copies of Z and $\stackrel{d}{=}$ means equal in distribution.

A random variable Z is said to be standard α–Fréchet if

$$\mathbb{P}\{Z_j(i) \le x\} = \Phi_\alpha(x) := \left\{ \begin{array}{ll} e^{-x^{-\alpha}} & ,\ x > 0 \\ 0 & ,\ x \le 0, \end{array} \right. \qquad (3)$$

for arbitrary $\alpha > 0$. One can check by using independence that equation (2) holds for any standard α–Fréchet Z. Let $Z, Z(1), \ldots Z(n)$ be iid standard α–Fréchet, and let $f(i) \ge 0$. Then, for any $x > 0$:

$$\mathbb{P}\{ \max_{1 \le i \le N} f(i) Z(i) \le x \} =$$
$$\prod_{1 \le i \le N} \mathbb{P}\{Z(i) \le x/f(i)\} =$$
$$\exp\{ - \sum_{i=1}^N f(i)^\alpha x^{-\alpha} \}, \qquad (4)$$

and thus

$$\xi := \max_{1 \le i \le N} f(i) Z(i) \stackrel{d}{=} \left(\sum_i f(i)^\alpha \right)^{1/\alpha} Z = \|f\|_{\ell_\alpha} Z.$$

That is, the weighted maxima ξ is an α–Fréchet variable with scale coefficient equal to $\|f\|_{\ell_\alpha}$ (i.e., the F_α–norm of f).

Hence, the max–stability of the $Z_j(i)$'s implies that:

$$E_j(f) \stackrel{d}{=} \|f\|_{\ell_\alpha} Z_1(1), \quad 1 \le j \le K.$$

One can also easily express the median of an α–Fréchet variable Z, with scale coefficient σ. Indeed, $\mathbb{P}\{Z \le \mathrm{med}(Z)\} = 1/2$, and by solving $\exp\{-\sigma^\alpha \mathrm{med}(Z)^{-a}\} = 1/2$, one obtains:

$$\mathrm{med}(Z) = \frac{\sigma}{(\ln 2)^{1/a}}.$$

Now, define the quantity:

$$L_\alpha(f) := (\ln 2)^{1/\alpha} \mathrm{med}\{E_j(f), \ 1 \le j \le K\}.$$

By keeping K independent realizations of the weighted maxima we can show the following:

Theorem 1. *Let $\epsilon \in (0,1)$ and $\delta > 0$. Then:*

$$\mathbb{P}\{ |\frac{L_\alpha(f)}{\|f\|_{\ell_\alpha}} - 1| \le \epsilon \} \ge 1 - \delta, \qquad (5)$$

provided that $K \ge \frac{C}{\epsilon^2} \log(\frac{1}{\delta})$, for some $C > 0$.

Proof. Observe that

$$\frac{L_\alpha(f)}{\|f\|_{\ell_\alpha}} \stackrel{d}{=} (\ln 2)^{1/\alpha} \mathrm{med}\{\xi_j, \ 1 \le j \le K\}, \qquad (6)$$

where ξ_j are independent standard α–Fréchet variables and where $\stackrel{d}{=}$ means equality in distribution. Therefore, the result follows from Lemma 2 of [15], since the derivative of $\Phi_\alpha^{-1}(y) = (\ln(1/y))^{-1/\alpha}$ at $y = 1/2$ is bounded. $\qquad \square$

Hence, $L_\alpha(f)$ can be used as an ϵ–approximation of the F_α–norm of the signal, for arbitrary $\alpha > 0$. Notice that the sum–stable sketches provide F_α–norm approximations only when $0 < \alpha \le 2$.

The power of the max–stable sketch lies in the fact that the α–Fréchet variables can be easily simulated in practice. If $U_j,\ j \in \mathbb{N}$ are independent uniformly distributed variables in $(0, 1)$, then $Z_j := \Phi_\alpha^{-1}(U_j) = (\ln(1/U_j))^{-1/\alpha}$ are independent standard α–Fréchet. Indeed, for all $x > 0$:

$$\mathbb{P}\{(\ln(1/U))^{-1/\alpha} \le x\} =$$
$$\mathbb{P}\{\ln(1/U) \ge x^{-\alpha}\} = \mathbb{P}\{U \le e^{-x^{-\alpha}}\} = e^{-x^{-\alpha}}.$$

One can also get the optimal asymptotic form of the constant C in Theorem 1.

Theorem 2. *Given $\epsilon > 0$ and $\delta > 0$, let $K(\epsilon, \delta)$ be the smallest integer for which (5) holds. Then, as $\epsilon,\ \delta \to 0$, we have*

$$K(\epsilon, \delta) \sim \frac{2}{\alpha^2 (\ln 2)^2} \frac{\ln(1/\delta)}{\epsilon^2}. \qquad (7)$$

Here $a_n \sim b_n,\ n \to \infty$, denotes asymptotic equivalence: $a_n/b_n \to 1,\ n \to \infty$.

Relation (7) gives, *for large K's*, the *exact* asymptotic size of the max–stable sketch which guarantees a prescribed (ϵ, δ)–precision. In practice, this approximation is 'very good' even if $K \approx 30$. Intricate finite–sample analysis is required to obtain the exact value for C in Theorem 1.

Proof. Let $\xi_1, \ldots, \xi_K, \ldots$ be independent standard α–Fréchet r.v.'s. By using the Brownian bridge asymptotics for the empirical distribution function $\widehat{F}_n(x) := n^{-1} \sum_{i=1}^n I\{\xi_i \le x\}$, we obtain, as $K \to \infty$,

$$\sqrt{K}\Big(\mathrm{med}\{\xi_j, \ 1 \le j \le K\} - m_\alpha \Big) \stackrel{d}{\to} \mathcal{N}(0, \sigma_\alpha^2), \quad (8)$$

where $\stackrel{d}{\to} \mathcal{N}(0, \sigma_\alpha^2)$ denotes convergence to the Normal distribution with zero mean and variance σ_α^2. For more details, see Ch. 3.9.4.2, p. 385 in [23]. Here $m_\alpha = (\ln 2)^{-1/\alpha}$ denotes the median of a standard α–Fréchet variable. The variance σ_α^2 has the form:

$$\begin{aligned} \sigma_\alpha^2 &= \frac{\Phi_\alpha(m_\alpha)(1 - \Phi_\alpha(m_\alpha))}{(\Phi_\alpha'(m_\alpha))^2} = \frac{1}{4(\Phi_\alpha'(m_\alpha))^2} \\ &= \frac{1}{\alpha^2 (\ln 2)^{2+2/\alpha}}, \end{aligned} \qquad (9)$$

3

Algorithm 1: Fast Max-Stable Insertion

Input:: A set of K variables, Number of groups G, Item i, Value $f_s(i)$ for arbitrary signal s, Hash function h.

1 Initialize PRG $R(i)$ using i as the seed.
2 $g = h(i) \bmod G$
3 **for** $gK/G \leq c \leq (g+1)K/G$ **do**
4   U = draw the next uniform number from R
     $Z = \ln(1/U)^{-1/\alpha}$
5     $K_c = \max(K_c, f_s(i) \cdot Z)$

---

where $\Phi_\alpha(x)$ and $\Phi'_\alpha(x)$ are the distribution and density functions, respectively, of a standard $\alpha-$Fréchet variable (see (3)).

In view of (6) and (8), we have that

$$\mathbb{P}\{|\frac{L_\alpha(f)}{\|f\|_{\ell_\alpha}} - 1| \leq \epsilon\} \sim 2\Phi(m_\alpha \epsilon \sqrt{K}/\sigma_\alpha) - 1,$$

as $K \to \infty$, where $\Phi(x) = \int_{-\infty}^{x} e^{-u^2/2} du/\sqrt{2\pi}$ is the standard Normal distribution function. By using that, $\ln(1-\Phi(x)) \sim -x^2/2$, as $x \to \infty$, we get

$$\ln(\delta) = \ln(2(1 - \Phi(m_\alpha \epsilon \sqrt{K}/\sigma_\alpha))) \sim -m_\alpha^2 \epsilon^2 K/(2\sigma_\alpha^2),$$

as $K \to \infty$. By using (9), $m_\alpha = (\ln 2)^{-1/\alpha}$ and solving for $K = K(\epsilon, \delta)$, we obtain (7). $\square$

3.2 The Fast Max–Stable Sketch

It is clear that the cost of updating the max–stable sketch is dominated by the need to generate the K α–Fréchet variables corresponding to each item i. Essentially, every insertion needs to update all variables K comprising the sketch. This operation can become very expensive for large sketch sizes, especially in streaming applications where fast insertions are critical. Nevertheless, the cost of insertions can be reduced significantly by partitioning the problem into smaller subsets. In particular, instead of updating all the K counters for every insertion i, we partition the input domain into a number of groups G, and assign a disjoint subset of K/G variables to every group. The partitioning of the input domain can be performed using any universal hash function. Essentially, every group forms an independent max–stable sketch on a smaller domain using only K/G variables. The algorithm for constructing the fast α–max-stable sketch over a set of signals is shown in Algorithm 1.

Clearly, this algorithm reduces the cost of each insertion by a factor G. Now, in order to estimate the F_α–norm of the signal, we estimate the F_α–norm of each group individually as in the original max–stable sketch, let $L_1 =$

$L_\alpha(f_1), \ldots, L_G = L_\alpha(f_G)$ be these estimates, and sum the results as follows:

$$L_\alpha(f) := (L_1^\alpha + \ldots + L_G^\alpha)^{1/\alpha}. \tag{10}$$

Since every item belongs to only one group, it is easy to see that Equation (10) is an estimate of $\|f\|_{\ell_\alpha}$.

We can show the following:

Theorem 3. *Let $\epsilon \in (0,1)$, $\delta > 0$, $K/G \geq C/\epsilon^2 \log(1/\delta)$. Also let $L_1, \ldots, L_G$ be the individual estimates per group, with $|L_i/\|f_i\|_{\ell_\alpha} - 1| \leq \epsilon$, with probability $1 - \delta$. Then:*

$$\mathbb{P}\{(1-\epsilon)^\alpha \leq \frac{\sum_{g \in [G]} L_i^\alpha}{\sum_{i \in [G]} \|f_i\|_{\ell_\alpha}^\alpha} \leq (1+\epsilon)^\alpha\} \geq 1 - G\delta.$$

Hence, the fast α–max-stable sketch with G groups provides $(1 \pm \epsilon)^\alpha$–approximate answers with probability $1 - G\delta$.

Proof. To prove the upper bound, we know that $L_i \leq (1 + \epsilon)\|f_i\|_{\ell_\alpha}$. Hence, $L_i^\alpha \leq (1 + \epsilon)^\alpha \|f_i\|_{\ell_\alpha}^\alpha$, and by taking the sum $\sum_{i \in [G]} L_i^\alpha \leq (1 + \epsilon)^\alpha \sum_{i \in [G]} \|f_i\|_{\ell_\alpha}^\alpha$. Similarly for the lower bound. The probability of failure can be computed directly by applying the union bound. $\square$

In practice, we observed that the fast max–stable sketch has excellent insertion performance, while providing accurate estimates that do not diverge significantly from those of the original sketch. This is also due to the fact that $\lim_{\epsilon \to 0} (1 \pm \epsilon)^\alpha/(1 \pm \epsilon) = \alpha = \text{const}$. Notice also the very important property that for $\alpha = 1$, the error bound of the fast max–stable sketch is equal to the error bounds of the individual group max–stable sketches. Hence, for the max–dominance norm estimation problem of [7] and [18] our sketch works very well in practice. Some experimental results will be presented in Section 6.

3.3 Estimating Distances

The same construction without any modification can be used similarly for approximating distances, as well as for recovering *exactly* relatively large components of f with high probability.

Consider two signals $f, g : \{1, \ldots, N\} \to [0, M]$ and let $E_j(f)$, $E_j(g)$, $j = 1, \ldots, K$ be their α–max-stable sketches, for arbitrary $\alpha > 0$. Observe that the max–stable sketches are non–linear and therefore even if $f(i) \leq g(i)$, $1 \leq i \leq N$, the sketch $E_j(g - f)$ does not equal $E_j(g) - E_j(f)$. Nevertheless, one can introduce a distance between the signals f and g, other than the norm $\|f - g\|_{\ell_\alpha}$ which can be computed by using the sketches $E_j(f)$ and $E_j(g)$.

Consider the functional

$$\|f^\alpha - g^\alpha\|_{\ell_1} := \sum_i |f(i)^\alpha - g(i)^\alpha|.$$

4

One can verify that $\|f^\alpha - g^\alpha\|_{\ell_1}$ is a metric on $\mathbb{R}_+^N$. Due to the non–linearity of the max–stable sketches, this metric, rather than the norm $\|f - g\|_{\ell_\alpha}$, is more natural in our setting [22]. Suppose for example that we have indicator signals, i.e. $f(i) = 1_A(i)$ and $g(i) = 1_B(i)$, for some $A, B \subset \{1, \dots, N\}$. The problem of efficiently estimating the size of the intersection $|A \cap B|$ is known to be hard [19]. Nevertheless, since $\|f^\alpha - g^\alpha\|_{\ell_1} = |A \cap B|$ (independently of $\alpha > 0$), if we can estimate this distance well, we can estimate the size of the intersection. Another application arises in change detection or classification problems. Given a set of signals $f_s(i), 1 \le s \le S$, we want to determine how they group or cluster together. If the only feasibly available information about the signals is their max–sketches, then we can compute the $S \times S$ *distance matrix* $D = (D_\alpha(f_s, f_l))_{1 \le s, l \le S}$. Many clustering and visualization algorithms can be then applied to the matrix D to determine possible associations and similarity patterns between the signals. For example, the class of *multidimensional scaling* algorithms, generate points x_s, $1 \le s \le S$ in an r–dimensional space, with pair–wise distances given by D. The goal is to find low–dimensional representations which nonetheless reveal patterns and structure among the points [2]. These point configurations can be further visualized (automatically or interactively), see e.g. [3].

Now, denote with '$\vee$' the max operation. Observe that:

$$\begin{aligned} \|f^\alpha - g^\alpha\|_{\ell_1} &= \\ &= \sum_i (f(i)^\alpha \vee g(i)^\alpha - f(i)^\alpha) + \\ &+ \sum_i (f(i)^\alpha \vee g(i)^\alpha - g(i)^\alpha) \\ &= 2\|f \vee g\|_{\ell_\alpha}^\alpha - \|f\|_{\ell_\alpha}^\alpha - \|g\|_{\ell_\alpha}^\alpha. \end{aligned}$$

By the max–linearity of max–stable sketches we get $E_j(f \vee g) = E_j(f) \vee E_j(g)$. Therefore, the terms in the last expression can be estimated in terms the estimator $L_\alpha(f)$ above. Namely, we define:

$$D_\alpha(f, g) := 2L_\alpha(f \vee g)^\alpha - L_\alpha(f)^\alpha - L_\alpha(g)^\alpha.$$

Theorem 4. *Let $\epsilon, \eta \in (0, 1)$, $\delta > 0$. If $\|f^\alpha - g^\alpha\|_{\ell_1} \ge \eta\|f \vee g\|_{\ell_\alpha}^\alpha$ then:*

$$\mathbb{P}\left\{ \left| \frac{D_\alpha(f, g)}{\|f^\alpha - g^\alpha\|_{\ell_1}} - 1 \right| \le \mathcal{O}(\epsilon/\eta) \right\} \ge 1 - 3\delta,$$

provided that $K \ge C/\epsilon^2 \log(1/\delta)$, for constant $C > 0$.

A proof sketch is given in [21].

For the example of the two indicator signals $1_A, 1_B$ above, suppose that $\|f^\alpha - g^\alpha\|_{\ell_1} = |A \cap B| \ge \eta\|f \vee g\|_{\ell_\alpha}^\alpha = |A \cup B|$, i.e. $|A \cap B|$ is not too small relative to $|A \cup B|$. Theorem 4 implies that we have a good estimator of the size of the intersection of two sets.

3.4 Estimating the Largest Components

Now, to recover point estimates for signal f we do the following. Given an $i_0 \in \{1, \dots, N\}$, set

$$g_j(i_0) := \frac{E_j(f)}{Z_j(i_0)}, \ j = 1, \dots, K,$$

Then $\widehat{f}(i_0) := \min_{1 \le j \le K} g_j(i_0)$, and we can show the following:

Theorem 5. *Let $\epsilon \in (0, 1)$, $\delta > 0$ and $i_0 \in \{1, \dots, N\}$. If $f(i_0) > \epsilon\|f\|_{\ell_\alpha}$ and $K \ge 1/\epsilon^\alpha \ln(1/\delta)$, then*

$$\mathbb{P}\{\widehat{f}(i_0) = f(i_0)\} \ge 1 - \delta.$$

The proof appears in [21]. Notice that α is chosen at sketch creation time according to the norm to be estimated by the sketch. A trivial solution for recovering the largest values of f would be of course to maintain a list with the top–K values observed. Nevertheless, the ability of max–stable sketches to recover part of the signal is important when a sketch needs to be used for estimating F_α–norms as in our motivating example. Hence, it obviates the need of using other solutions. Current alternatives do not have this capability.

4 The Range–Efficient Sketch

In this section we briefly review the Range–Efficient F_0 sketch proposed in [18], since it is considered to be the state-of-the-art for estimating max–dominance norms in practice, hence forms a good basis for comparison with the fast max–stable sketch.

Consider once again a set of signals $f_s : \{1, \dots, N\} \to \{0, \dots, M\}$ appearing as a stream of tuples $(i, f_s(i))$ in arbitrary order of i and s, where the range of f_s is now a subset of $\mathbb{Z}^*$ instead of $\mathbb{R}^*$. The max–dominance norm is the F_1–norm of the dominant signal, that is $\sum_i \max_s f_s(i)$ [7]. The problem of estimating the max–dominance norm can be reduced to that of estimating the F_0–norm on a stream of ranges $\mathcal{R}$, by converting every insertion $(i, f_s(i))$ into the range $[(i-1) \cdot M, (i-1) \cdot M + f_s(i))$, where every range $[a, b]$ appearing on the stream implies that all items $a, a+1, \dots, b$ appear on the stream. Essentially we map every item into a disjoint interval of the integer domain. Then, it is easy to verify that the max–dominance norm equals the number of distinct elements observed in $\mathcal{R}$ [18]. In order to estimate the F_0–norm on stream $\mathcal{R}$ efficiently, a range-efficient algorithm is needed for processing ranges in time sub–linear to their length.

The range–efficient F_0 sketch of [18] is a sampling based approach that maintains a sample of disjoint ranges that

have appeared on the stream. Every time a new range appears, if the range intersects with any other ranges in the sample, only the union of those ranges is kept. Otherwise, the new range is sampled with some set probability. When the sample becomes full we re-sample the existing ranges using a higher rate in order to discard some entries. Initially, the sampling level is $l = 0$, and the sampling rate is $1/2^l = 100\%$. After the sample becomes full we increase l by one and re-sample. The F_0–norm of the stream is estimated as the sum of the lengths of the ranges contained in the sample, scaled according to the current sampling level.

In order to to be able to insert ranges in sub–linear time, Pavan and Tirthapura proposed to sample a given range according to the following principle: Given range $[a, b]$ use a linear hash function $h(x) = dx + c \, mod \, P$ [4] to convert the range into an arithmetic progression with common difference d (where P is a prime number larger than $10NM$). After hashing all elements of $[a, b]$ using $h(x)$ we get the progression $h(a), h(a) + d, h(a) + 2d, \dots, h(a) + (b - a - 1)d$. We now sample a given interval if any element of the progression is contained in interval $L = [0, P/2^l)$ defined by the sampling level. The original problem is now reduced to the problem of finding if an arithmetic progression in modulo P arithmetic intersects with interval L. Pavan and Tirthapura propose a divide and conquer technique that can solve the new problem in time logarithmic to the length of the series.

The space complexity of estimating the max–dominance norm using the range–efficient sketch becomes $O(1/\epsilon^2 (\log M + \log N) \log 1/\delta)$ and the worst case processing time per item $O(1/\epsilon^2 \log \log N \log (b - a) \log 1/\delta)$, while the amortized time is $O(\log \frac{b-a}{\epsilon} \log \frac{1}{\delta})$.

5 Implementation Issues

In this section we evaluate the existing approximate algorithms for max–dominance estimation from a practical perspective, and motivate the need for a simpler construction which is much easier to implement and does not require a large number of floating point operations.

The method of Cormode and Muthukrishnan [7] uses p–sum-stable sketches with very small $p > 0$. The p–sum-stable distributions involved in these sketches have infinite moments of all orders greater than p and in practice take extremely large values. This poses a number of practical challenges in storing and in fact precisely generating these random sketches. In contrast, our method does involve heavy–tailed random variables but they are not *extremely heavy–tailed* and can be shown to have good computational properties [21]. Furthermore, Fréchet distributions can be simulated much more efficiently than sum–stable distributions, as was shown in Section 3.1. Therefore, in practice our

method is expected to be more robust than the one in [7]. Moreover, asymptotically the storage and per item processing times of our method are significantly smaller as well.

It is worth also noting here that for larger α we expect the max–stable sketch to yield even better results, since the α–Fréchet distribution becomes even less heavy–tailed. Errors are introduced mainly due to heavy tails appearing in division by small numbers. These errors are more pronounced for small α since the extreme heavy–tails lead to numbers beyond the machine precision with non–negligible probability.

The sketch proposed by Pavan and Tirthapura [18] has various drawbacks from a technical perspective. Even though the algorithm they propose for efficiently computing the intersection of a given range with an interval has sub–linear asymptotic cost with respect to the length of the range, in practice the constant time computations that need to be performed in order to compute boundary conditions in modulo P arithmetic require integer multiplications and divisions with extremely large values (significantly larger than P, where P is larger than $10NM$ in the first place). Hence, for a practical implementation of this algorithm one has to revert to using a large number of floating point divisions and multiplications, as well as recursive procedures for computing products in modulo P arithmetic. In some specialized hardware with limited precision and no FPU available (e.g., FPGA enabled NICs), these issues will render the algorithm very expensive to compute (and very hard to implement), and in some cases more expensive than a simple linear cost algorithm (depending on the size of the input). Furthermore, even using state-of-the-art hardware the algorithm can still overflow the FPU in some special cases.

Furthermore, the fact alone that every insertion $(i, f_s(i))$ is converted into a range with potential lower bound equal to MN necessitates the use of at least 64-bit integer arithmetic for large Ns (e.g., $2^{32} - 1$ for IPv4 addresses). In specialized hardware as well as most modern 32-bit processors this requirement slows down computations ever further. (Notice here that for very large input domains one might be tempted to map the active domain to the range $[1, N']$. This is not feasible since N' is one of the quantities we are implicitly trying to estimate to begin with.) Notice also, that even though the algorithm requires manipulation of floating point numbers, by nature it can work only for integer valued input. In contrast, the max–stable sketch can handle real valued domains as well. Finally, observe that in order to use the range–efficient sketch for max–dominance norm estimation, the maximum value M has to be available in advance in order to properly produce the desired ranges. This is not always the case in practice. For example, in network monitoring applications it cannot be accurately decided in advance what the maximum traffic generated will be.

Finally, another problem of the sampling based approach is the fact that the sampling level l cannot be increased indefinitely. When $l > \log_2 p$ the maximum level is reached and the algorithm fails. Essentially, at that point the sketch has to be reconstructed. This condition will occur frequently if a large number of the input ranges are disjoint, and hence the reservoir keeps becoming full very often. Such scenarios are not rare in practice. In contrast, our algorithm does not suffer from such extreme failures.

6 Experimental Evaluation

In order to validate the proposed techniques we performed a qualitative analysis of the max–stable and range–efficient sketches with real datasets. We implemented both algorithms using C++. For the range–efficient sketch we opted at using 64-bit integer arithmetic and tried to minimize the number of floating point operations that need to be performed. Nevertheless, since we perform our experimental evaluation on a Pentium(R) 4 CPU which is equipped with an advanced FPU, the floating point computations did not affect the insertion performance of the sketch. Our implementation of the sketch is now freely available LGPL code as part of the Sketch Library [14].

For our datasets we used one day worth of network traffic generated by the AT&T Shannon Labs. The dataset contains close to 101K distinct IP addresses, and close to 16.3M traces of per second, per IP aggregated traffic. In our setting, we would like to know what is the "worst case influence" of a coordinated transmission at the any–time highest observed rate of all IPs on the network backbone of the Labs, for varying intervals of observation ranging from a few minutes to hours. Hence, the domain of items here is the set of distinct IP addresses, while one signal consists of the set of traces within a specific observation interval. Furthermore, the observed values are the total number of bytes transmitted by an IP within the observation interval.

6.1 Max–Dominance Norm Estimation

For this set of experiments we estimate the max–dominance norm and compare against the exact answer to evaluate the estimation accuracy of the sketches. We compare the range–efficient sketch with the fast 1–max–stable sketch.

In the first experiment we run the sketches for 10 minute observation intervals and an increasing amount of traffic, from 4 up to 24 hours. We constantly evaluate the max–dominance norm against the current exact answer. Overall, we run in parallel 50 copies of every sketch and take averages at the end. We fix the sketch sizes to at most 5 KBytes per run. Notice that an exact solution would need approximately 790 KBytes. For the fast max–stable sketch

we set the group size G equal to 10. The average relative error along with the 5-th and 95-th percentiles are shown in Figure 1(a). The insertion cost is shown in Figure 1(b). The range–efficient sketch (re) has better accuracy but two times higher insertion cost, compared to the fast max–stable sketch (f-ms). In addition the max–stable sketch has considerably smaller variance in most cases.

In the rest of our experiments we always report averages over 50 runs and keep the default values to 10 minute observation intervals, 24 hours of data, 5 KByte sketch sizes, and 10 groups per fast max–stable sketch.

Next, we evaluate the accuracy and processing cost of the sketches as a function of sketch size. Results are shown in Figures 2(a) and 2(b). As expected, the larger the sketch size the better the accuracy and the higher the insertion cost. Notice that the cost for the fast max–stable sketch increases linearly, while the estimation accuracy reaches eventually the accuracy of the range–efficient sketch.

In Figure 3 we show the effect of increasing number of groups on the insertion cost of the fast max–stable sketch. Note that the group parameter is not relevant to the range–efficient sketch which is shown in the same plot only for comparison purposes. Clearly, the larger the number of groups the better the performance of the sketch, which verifies the theoretical results.

Finally, we evaluate the performance of the sketches as a function of the length of the observation interval, ranging from 1 up to 60 minutes. The largest the interval, the largest the traffic generated by each IP but the smaller the number of signals that need to be processed. Results are shown in Figure 4. We observe that the accuracy of the sketches remains at the same level, while the processing cost decreases accordingly.

6.2 General F_α–norm Estimation

For completeness we show here some results on estimating the energy, and higher norms, of the dominant signal over the real network traffic dataset. Once again, we use fast α–max–stable sketches for given α. Figure 5(a) shows the accuracy of the 2–max–stable sketch as a function of size in KBytes. Figure 5(b) shows the results for varying α. Observe that the max–stable sketch has better accuracy for increasing α which validates our conjecture that less heavy–tailed distributions can be approximated more accurately.

6.3 Largest Component Reconstruction

We also test the efficiency of the sketch for recovering the highest components of the dominant signal. Once again we use real network datasets and reconstruct the top–K highest components of the signal, computing the average relative error of the reported values, as well as the total
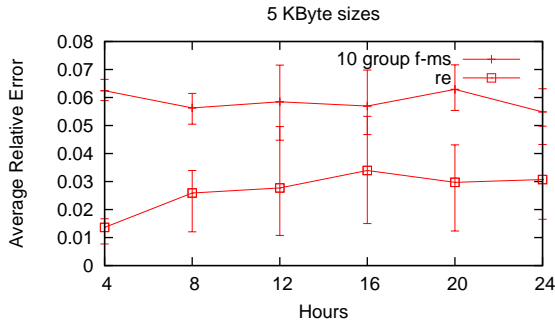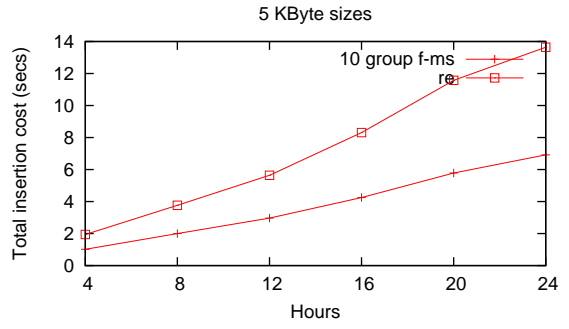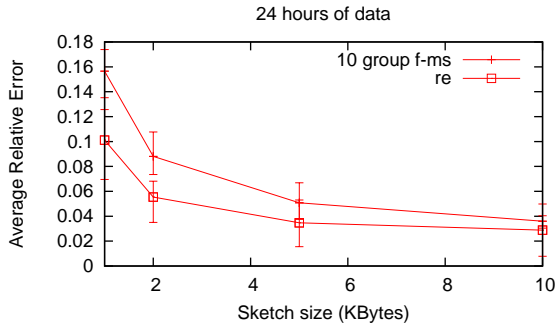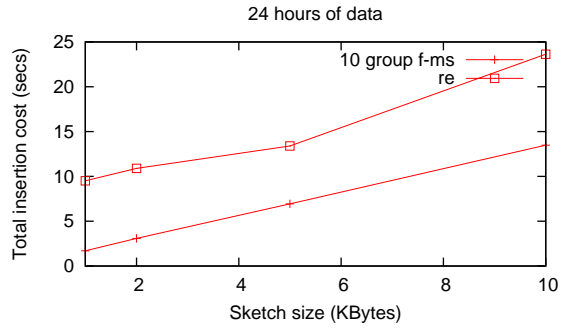
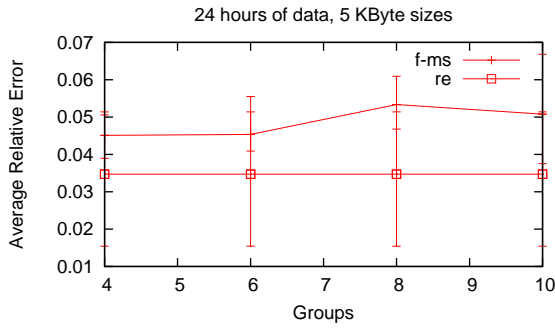**Figure 1.** F_1–norm performance as a function of hours of operation.
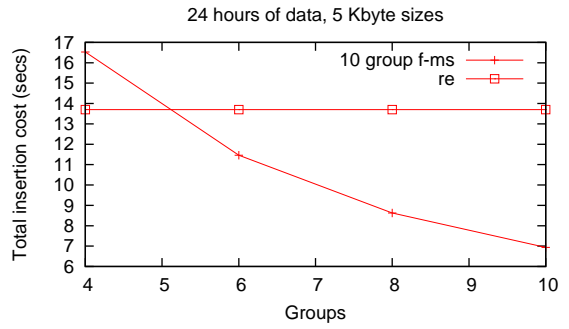
(a) Relative error.

(b) Insertion cost.

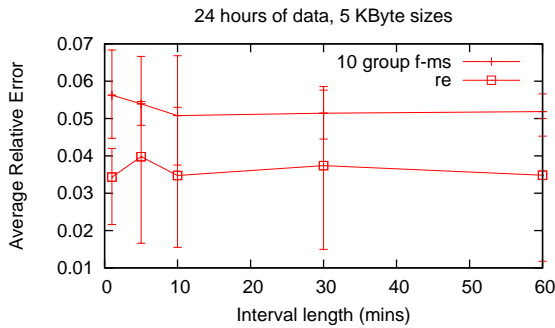**Figure 2.** F_1–norm performance as a function of sketch size.
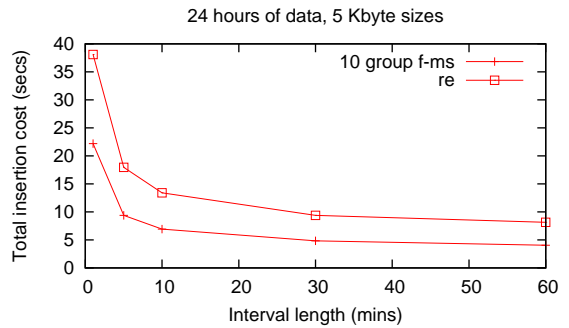
(a) Relative error.

(b) Insertion cost.

Figure 3. F_1–norm performance as a function of the number of groups G.

(a) Relative error.

(b) Insertion cost.

Figure 4. F_1–norm performance as a function of the length of the observation interval.

(a) Relative error.

(b) Insertion cost.

number of exactly reconstructed components. The results are shown in Figure 6. The max–stable sketch can recover exactly a very large percentage of the largest components throughout all queries that we run. For the rest of the components it is clear that the sketch has excellent estimation accuracy.

6.4 Distance Estimation

Finally, we test the accuracy of the max–stable sketch for distance computations. For this experiment we use network traffic datasets for two different days (on the 4th of July and on the 6th of July of 2006). The results are shown in Figure 7. The quality of the estimations drop somewhat to 11% of the exact value. This is expected due to the increased error reported by Theorem 4.

7 Conclusion

The max–stable sketch is an efficient (ϵ, δ)–aproximation algorithm for computing any F_p–norm on the dominant signal of a set of signals. The same sketch can also be used for recovering the largest components of the dominant signal exactly and with high probability, as well as for approximating special types of distances between dominant signals, which can be useful for change detection applications and more. An added benefit of our sketch is simplicity of implementation (as is exhibited in Algorithm 1), especially in contrast with other alternatives that can be used only for a small subset of the functionality that the max–stable sketch provides. We also evaluated our sketch experimentally using real datasets and observed very good query and update performance in practice.

References

[1] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *Proc. of ACM Symposium on Theory of Computing (STOC)*, pages 20–29, 1996.

[2] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Series in Statistics. Springer, New York, 1997.

[3] A. Buja and D. Swayne. Visualization methodology in multidimensional scaling. *Journal of Classification*, 19(1):7–43, 2002.

[4] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[5] E. Cohen. Self–estimation framework with applications to transitive closure and reachability. *Journal of Computer and System Sciences*, 55(3):441–453, 1997.

[6] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. Effective computation of biased quantiles over data streams. *Proc. of International Conference on Data Engineering (ICDE)*, pages 20–31, 2005.

[7] G. Cormode and S. Muthukrishnan. Estimating dominance norms of multiple data streams. In *Proc. of the European Symposium on Algorithms (ESA)*, 2003.

[8] A. Dobra, M. N. Garofalakis, J. Gehrke, and R. Rastogi. Processing complex aggregate queries over data streams. In *Proc. of ACM Management of Data (SIGMOD)*, pages 61–72, 2002.

[9] J. Feigenbaum, S. Kannan, M. J. Strauss, and M. Viswanathan. An approximate L^1–difference algorithm for massive data streams. In *SICOMP*, pages 131–151, 2002.

[10] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

[11] S. Ganguly, M. Garofalakis, and R. Rastogi. Processing set expressions over continuous update streams. In *Proc. of ACM Management of Data (SIGMOD)*, pages 265–276, 2003.

[12] P. B. Gibbons and S. Tirthapura. Estimating simple functions on the union of data streams. In *Proc. of the ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 281–291, 2001.

[13] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In *Proc. of ACM Management of Data (SIGMOD)*, pages 58–66, 2001.

[14] M. Hadjieleftheriou. The Sketch Library. `http:// sourceforge.net/ projects/ sketches`.

[15] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 189–197, 2000.

[16] G. S. Manku and R. Motwani. Approximate Frequency Counts over Data Streams. In *Proc. of Very Large Data Bases (VLDB)*, pages 346–357, 2002.

[17] S. Muthukrishnan. Data streams: algorithms and applications, 2003.

[18] A. Pavan and S. Tirthapura. Range-efficient computation of F_0 over massive data streams. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 32–43, 2005.

[19] A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.

[20] G. Samorodnitsky and M. S. Taqqu. *Stable Non-Gaussian Processes: Stochastic Models with Infinite Variance*. Chapman and Hall, 1994.

[21] S. Stoev and M. Taqqu. Max–stable sketches: exact point queries and ℓ_α–norm estimation for non–negative signals. Technical Report 433, Department of Statistics, University of Michigan, 2006.

[22] S. Stoev and M. S. Taqqu. Extremal stochastic integrals: a parallel between max–stable processes and α–stable processes. *Extremes*, 8:237–266, 2005.

[23] A. W. van der Vaart and J. A. Wellner. *Weak convergence and empirical processes*. Springer Series in Statistics. Springer-Verlag, New York, 1996. With applications to statistics.
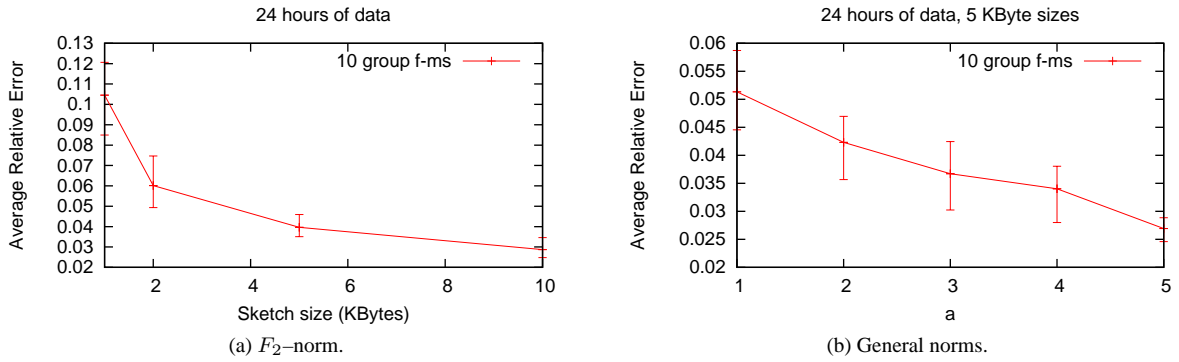
(a) F_2–norm.

(b) General norms.

Figure 5. Estimation accuracy for general F_α–norms.

(a) Relative error.

(b) Percent of reconstructed components.
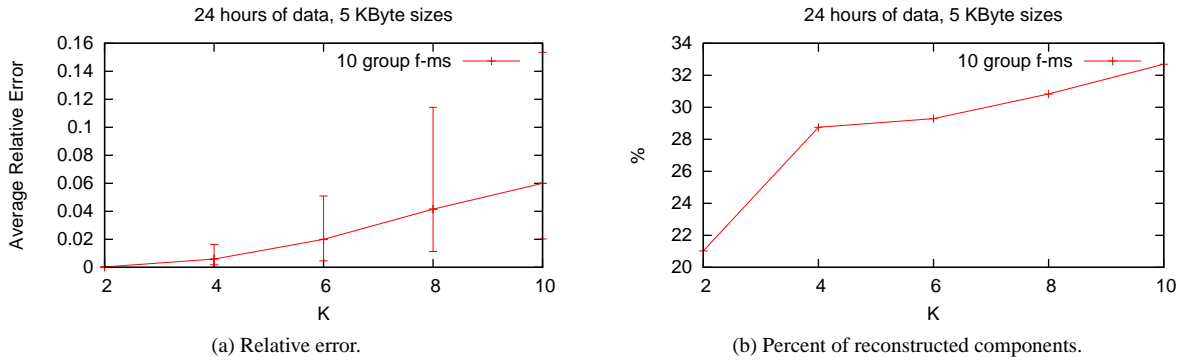
Figure 6. Performance for largest component reconstruction.
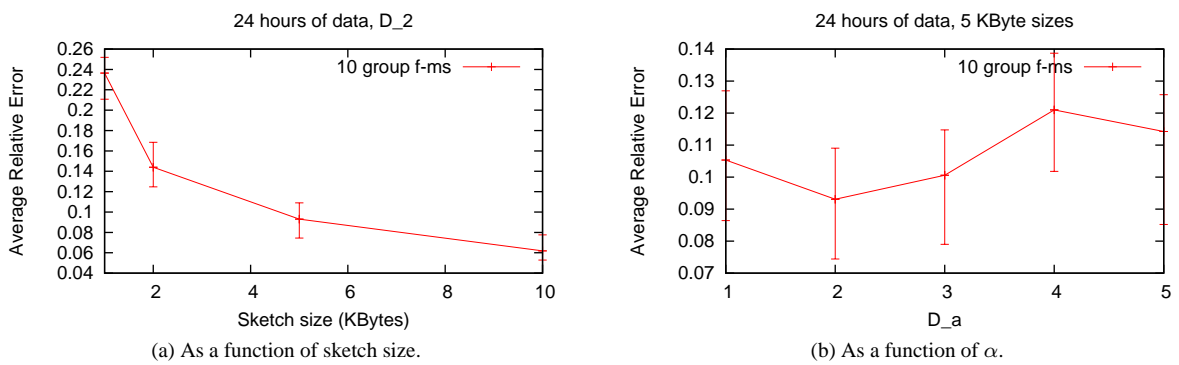

(a) As a function of sketch size.

(b) As a function of α.

Figure 7. Performance for distance estimation.