Functional Monitoring without Monotonicity*

Chrisil Arackaparambil, Joshua Brody, and Amit Chakrabarti

Department of Computer Science, Dartmouth College, Hanover, NH 03755, USA

Abstract. The notion of *distributed functional monitoring* was recently introduced by Cormode, Muthukrishnan and Yi [4] to initiate a formal study of the communication cost of certain fundamental problems arising in distributed systems, especially sensor networks. In this model, each of k sites reads a stream of tokens and is in communication with a central coordinator, who wishes to continuously monitor some function f of σ, the union of the k streams. The goal is to minimize the number of bits communicated by a protocol that correctly monitors $f(\sigma)$, to within some small error. As in previous work, we focus on a threshold version of the problem, where the coordinator's task is simply to maintain a single output bit, which is 0 whenever $f(\sigma) \leq \tau(1-\varepsilon)$ and 1 whenever $f(\sigma) \geq \tau$. Following Cormode et al., we term this the $(k, f, \tau, \varepsilon)$ functional monitoring problem.

In previous work, some upper and lower bounds were obtained for this problem, with f being a frequency moment function, e.g., F_0, F_1, F_2. Importantly, these functions are *monotone*. Here, we further advance the study of such problems, proving three new classes of results. First, we provide nontrivial monitoring protocols when f is either H, the empirical Shannon entropy of a stream, or any of a related class of entropy functions (Tsallis entropies). These are the first nontrivial algorithms for distributed monitoring of non-monotone functions. Second, we study the effect of non-monotonicity of f on our ability to give nontrivial monitoring protocols, by considering $f = F_p$ with deletions allowed, as well as $f = H$. Third, we prove new lower bounds on this problem when $f = F_p$, for several values of p.

Keywords: Communication complexity, distributed algorithms, data streams, sensor networks.

1 Introduction

Energy efficiency is a key issue in sensor network systems. Communication, which typically uses power-hungry radio, is a vital resource whose usage needs to be minimized [7]. Several other distributed systems have a similar need for minimizing communication. This is the primary motivation for our present work, which is a natural successor to the recent work of Cormode, Muthukrishnan

* Work supported in part by an NSF CAREER Award CCF-0448277 and NSF grant EIA-98-02068.

S. Albers et al. (Eds.): ICALP 2009, Part I, LNCS 5555, pp. 95–106, 2009.

and Yi [4], who introduced a clean formal model to study this issue. The formalization, known as *distributed functional monitoring*, involves a multi-party communication model consisting of k *sites* (the sensors, in a sensor network) and a single central *coordinator*. Each site asynchronously receives "readings" from its environment, formalized as a *data stream* consisting of *tokens* from a discrete universe. The union of these streams defines an overall input stream σ that the coordinator wishes to monitor continuously, using an appropriate protocol involving private two-way communication channels between the coordinator and each site. Specifically, the coordinator wants to continuously maintain approximate knowledge of some nonnegative real-valued function f of σ. (We assume that f is invariant under permutations of σ, which justifies our use of "union" above, rather than "concatenation.")

As is often the case in computer science, the essence of this problem is captured by a threshold version with Boolean outputs. Specifically, we have a threshold $\tau \in \mathbb{R}_+$ and an approximation parameter $\varepsilon \in \mathbb{R}_+$, and we require the coordinator to continuously maintain an output bit, which should be 0 whenever $f(\sigma) \leq \tau(1-\varepsilon)$ and 1 whenever $f(\sigma) \geq \tau$.[1] Following [4], we call this the $(k, f, \tau, \varepsilon)$ functional monitoring problem. This formulation of the problem combines aspects of streaming algorithms, sketching and communication complexity.

Motivation. Plenty of recent research (e.g., in network and database settings) has studied such continuous monitoring problems, for several special classes of functions f (see, e.g., [2,6,5,12]). However, *formal* worst-case bounds on communication cost were first addressed in detail in [4]. We continue to address this here. Philosophically, the study of such monitoring problems is a vast generalization of Slepian-Wolf-style distributed source coding [13] in much the same way that communication complexity is a vast generalization of basic source coding in information theory. Furthermore, while the problems and the model we consider here are strongly reminiscent of streaming algorithms, there are notable additional challenges: for instance, maintaining an approximate count of the total number of tokens received is a nontrivial problem in our setting, but is trivial in the streaming model. For a more detailed discussion of prior research, we refer the reader to [4] and the references therein.

Our Results and Comparison with Prior Work. Our work studies $(k, f, \tau, \varepsilon)$ functional monitoring for two natural classes of functions f: the empirical Shannon entropy H (and its generalization: Tsallis entropy) and the frequency moments F_p. For an input stream σ of tokens from the universe $[n] := \{1, 2, \ldots, n\}$, let f_i denote the number of appearances of i in σ, where $i \in [n]$. For $p \geq 0$, the pth frequency moment $F_p(\sigma)$ is defined to be $\sum_{i=1}^{n} f_i^p$. Note that p can be non-integral or zero: indeed, using the convention $0^0 = 0$ makes $F_0(\sigma)$ equal to the number of distinct tokens in σ. These functions F_p capture important statistical properties

[1] Clearly, a solution to the value monitoring problem solves this threshold version, and the value monitoring problem can be solved by running, in parallel, several copies of a solution to this threshold version with geometrically spaced thresholds.

of the stream and have been studied heavily in the streaming algorithms literature [1,9]. The stream σ also implicitly defines a probability distribution over $[n]$, given by $\Pr[i] = f_i/m$, where m is the length of σ. For various applications, especially ones related to anomaly detection in networks, the entropy of this distribution — also called the empirical entropy of the stream — is a measure of interest. Abusing notation somewhat, we denote this as $H(\sigma)$, when the underlying entropy measure is Shannon entropy: thus, $H(\sigma) = \sum_{i=1}^{n}(f_i/m)\log(m/f_i)$.[2] We also consider the family of functions $T_\alpha(\sigma) = (1 - \sum_{i=1}^{n}(f_i/m)^\alpha)/(\alpha - 1)$, which are collectively known as Tsallis entropy [14] and which generalize Shannon entropy, as shown by considering the limit as $\alpha \to 1$.

We study the effect of *non-monotonicity* of f on the $(k, f, \tau, \varepsilon)$ problem: the bounds of Cormode et al. [4] crucially exploited the fact that the functions being monitored were monotone nondecreasing. We obtain three new classes of results. First, we provide nontrivial monitoring protocols for H, and the related functions T_α. For this, we suitably extend recent sketching algorithms such as those due to Bhuvanagiri and Ganguly [3] and Harvey et al. [8]. These are the first nontrivial algorithms for monitoring non-monotone functions.[3] Our algorithms, which are simple and easily usable, can monitor continuously until the end of the stream, even as the $f(\sigma)$ crosses the threshold multiple times. This is the desired behavior when monitoring non-monotone functions.

Secondly, we prove lower bounds for monitoring $f = F_p$ with *deletions* allowed: i.e., the stream can contain "negative tokens" that effectively delete earlier tokens. In contrast with the good upper bounds in [4] for monitoring F_p *without* deletions (a monotone problem), we show that essentially no good upper bounds are possible. Using similar techniques, we also give a lower bound for monitoring H that is necessarily much milder, and in the same ballpark as our upper bound.

Thirdly, we prove new lower bounds for the monotone problems $f = F_p$, without deletions, for various values of p. These either improve or are incomparable with previous bounds [4]; see Table 1 for a side-by-side comparison.

Notation, etc. We now define some notation that we use at various points. We use $|\sigma|$ to denote the length of the stream σ and $\sigma_1 \circ \sigma_2$ to denote the concatenation: σ_1 followed by σ_2. We typically use $S_1, \ldots, S_k$ to denote the k sites, and C to denote the coordinator, in a $(k, f, \tau, \varepsilon)$ functional monitoring protocol. We tacitly assume that randomized protocols use a public coin and err with probability at most 1/3 at each point of time. These assumptions do not lose generality, as shown by appropriate parallel repetition and the private-versus-public-coin theorem of Newman [11]. We use m to denote the overall input length (i.e., number of tokens) seen by the protocol under consideration. We state our communication bounds in terms of m, k and ε, and sometimes τ.

[2] Throughout this paper we use "log" to denote logarithm to the base 2 and "ln" to denote natural logarithm.

[3] Muthukrishnan [10] gives an upper bound for monitoring a non-monotone function, but with additive error.

Table 1. Summary of our results (somewhat simplified) and comparison with previous work [4]. Dependence on τ is not shown here, but is stated in the relevant theorems.

Problem	**Previous Results**	**Our Results**
H, deterministic	$\widetilde{O}(m)$, trivially	$\Omega(k\varepsilon^{-1/2}\log m)$
H, randomized		$\widetilde{O}(k\varepsilon^{-3}\log^4 m)$, $\Omega(\varepsilon^{-1/2}\log m)$
F_p, dels., determ.		$\Omega(m)$
F_p, dels., rand.		$\Omega(m/k)$
F_1, deterministic	$O(k\log(1/\varepsilon))$, $\Omega(k\log(1/(\varepsilon k)))$	$\Omega(k\log(1/\varepsilon))$
F_0, randomized	$\widetilde{O}(k/\varepsilon^2)$, $\Omega(k)$	$\Omega(1/\varepsilon)$, $\Omega(1/\varepsilon^2)$ if round-based
$F_p, p > 1$, rand.	$\widetilde{O}(k^2/\varepsilon + (\sqrt{k}/\varepsilon)^3)$, $\Omega(k)$, for $p = 2$	$\Omega(1/\varepsilon)$, $\Omega(1/\varepsilon^2)$ if round-based

2 An Algorithm for Monitoring Entropy

We discuss randomized algorithms for monitoring Shannon entropy, H, and Tsallis entropies, T_α. These provide the first nontrivial communication upper bounds for the monitoring of non-monotone functions. At a high level, our algorithms monitor changes (in the L_1 sense) in the empirical probability distribution defined by the input streams. For probability distributions μ, ν on the set $[n]$, we write $\|\mu - \nu\|_1 = \sum_{i=1}^{n} |\mu(i) - \nu(i)|$. We use the following three technical lemmas, whose proofs are left to the full version of the paper.

Lemma 1. *Let σ and σ' be streams of tokens from $[n]$, and μ and ν denote the empirical distributions induced by σ and $\sigma \circ \sigma'$ respectively. Let $m = |\sigma|$. If $|\sigma'| \le m$, then, $|H(\sigma \circ \sigma') - H(\sigma)| \ \le \ \|\nu - \mu\|_1 \log(2m)$.*

Lemma 2. *Let $\sigma, \sigma', \mu, \nu$ and m be defined as in Lemma 1. Then, for all $\alpha > 1$, $|T_\alpha(\sigma \circ \sigma') - T_\alpha(\sigma)| \ \le \ \|\nu - \mu\|_1 \cdot \min\{\log(2m), \alpha/(\alpha - 1)\}$.*

Lemma 3. *Let $\sigma, \sigma', \mu, \nu$ and m be defined as in Lemma 1. Then if $|\sigma'| < \ell$, then $\|\nu - \mu\|_1 < 2\ell/m$.*

We also need an *entropy sketching scheme*, such as the one provided by the following result, due to Harvey, Nelson and Onak [8].

Fact 1. *Let $\varepsilon > 0$. There is an algorithm that maintains a data structure (called a "sketch") $\mathcal{S}_H(\sigma)$, based on an input stream σ, such that (1) based on $\mathcal{S}_H(\sigma)$, we can compute an estimate $\hat{H}(\sigma) \in [H(\sigma) - \varepsilon, H(\sigma) + \varepsilon]$, (2) we can suitably combine $\mathcal{S}_H(\sigma_1)$ and $\mathcal{S}_H(\sigma_2)$ to obtain $\mathcal{S}_H(\sigma_1 \circ \sigma_2)$, and (3) $\mathcal{S}_H(\sigma)$ can be stored using $\widetilde{O}(\varepsilon^{-2}\log m \log n \log(mn))$ bits.[4] Here, the $\widetilde{O}$ notation hides factors polynomial in* $\log\log m$ *and* $\log(1/\varepsilon)$. □

[4] The $\widetilde{O}(\varepsilon^{-2}\log m)$ bound in [8] is on the number of words of storage, each $O(\log(mn))$ bits long, and does not include $O(\log n)$ space for a pseudorandom generator.

The Algorithm. We proceed in multiple *rounds*. At the end of the ith round, let ρ_{ij} be the overall stream seen at site S_j, let $\sigma_i = \rho_{i1} \circ \cdots \circ \rho_{ik}$, and let $m_i = |\sigma_i|$. In round 0, sites directly forward input tokens to the coordinator C, who ends the round after seeing a $c_0 := 100$ items. Then, C uses $\mathcal{S}_H$ from Fact 1 to get an estimate $\hat{H}(\sigma_0)$ of $H(\sigma_0)$ with an additive error of $\hat{\varepsilon} := \varepsilon\tau/4$.

For rounds $i > 0$, C and $S_1, \ldots, S_k$ simulate a $(k, F_1, \tau_i, \frac{1}{2})$ monitoring algorithm, such as the one from [4], using error $\frac{1}{2}$ and threshold $\tau_i := \min\{m_{i-1}, m_{i-1}\lambda_i/(2\log(2m_{i-1}))\}$, where $\lambda_i = \tau(1 - \frac{\varepsilon}{4}) - \hat{H}(\sigma_{i-1})$ if $\hat{H}(\sigma_{i-1}) < \tau(1 - \frac{\varepsilon}{2})$, and $\lambda_i = \hat{H}(\sigma_{i-1}) - \tau(1 - \frac{3\varepsilon}{4})$ otherwise. λ_i is the slack of the estimate $\hat{H}(\sigma_{i-1})$ from τ (or $\tau(1-\varepsilon)$ in the latter case), while allowing for error of the estimates. The choice of τ_i ensures that the simulated F_1 monitoring algorithm notifies the coordinator by outputting 1 when too many items (as determined from the technical lemmas) have been received in the round. When this happens, C signals each S_j that round i is ending, whereupon S_j sends it $\mathcal{S}_H(\rho_{ij})$. Then, C computes $\mathcal{S}_H(\sigma_i)$, updates its estimate $\hat{H}(\sigma_i)$, and outputs 1 iff $\hat{H}(\sigma_i) \geq \tau(1 - \frac{\varepsilon}{2})$.

Theorem 1. *The above is a randomized algorithm for $(k, H, \tau, \varepsilon)$ functional monitoring that communicates $\widetilde{O}(k\varepsilon^{-3}\tau^{-3}\log^3 m \log n \log(mn))$ bits.*

Proof. We first analyze the correctness. In round 0, it is trivial for the coordinator to output the correct answer. Now, for round $i > 0$, suppose the coordinator outputs 0 at the end of round $i - 1$. Then, we must have $\hat{H}(\sigma_{i-1}) \leq \tau(1 - \frac{\varepsilon}{2})$, whence $H(\sigma_{i-1}) < \tau(1 - \frac{\varepsilon}{4})$ by the bound on the sketching error. By the correctness of the F_1 monitoring algorithm, we receive at most τ_i items during round i. Therefore by Lemmas 1 and 3, when going from σ_{i-1} to σ_i, the total entropy will be less than τ throughout round i. Hence, the coordinator is free to output zero through the end of round i. If the coordinator instead outputs 1 at the end of round $i - 1$, we are guaranteed to remain above $\tau(1 - \varepsilon)$ similarly.

To bound the communication cost, we need to estimate both the number of rounds, and the number of bits exchanged in each round. It is easy to see that for each round i, $\lambda_i \geq \varepsilon\tau/4$. Suppose the stream ends during round $r + 1$. Then,

$$\begin{aligned} m \;\geq\; m_r \;\geq\; m_{r-1} + \tau_r/2 \;&\geq\; m_{r-1}\left(1 + \min\{1/2, \tau\varepsilon/(16\log(2m_{r-1}))\}\right) \\ &\geq\; m_{r-1}\left(1 + \min\{1/2, \tau\varepsilon/(16\log(2m))\}\right) \;=\; m_{r-1}\beta \quad \text{(say)}, \end{aligned}$$

where the second inequality follows from the guarantee of the F_1 monitoring algorithm. Iterating the above recurrence for m_r, we get $m \geq c_0\beta^r$, whence $r \leq \log(m/c_0)/\log\beta = O(\max\{\log m, \log^2 m/(\tau\varepsilon)\})$, where the final bound uses $\ln(1 + x) \geq x/(x + 1)$ for all $x > 0$. In each round, we use $O(k\log m)$ bits to send τ_i to the sites and $O(k)$ bits for the F_1 algorithm. These terms are dominated by the sizes of the sketches that the sites send. Using the size bound from Fact 1 and the above bound on r, we can bound the total communication by $\widetilde{O}(k\varepsilon^{-3}\tau^{-3}\log^3 m \log n \log(mn))$, for m large enough (i.e., if $\log m \geq \tau\varepsilon$). □

Our algorithm for monitoring Tsallis entropy is similar. Lemma 2 bounds T_α just as Lemma 1 bounds H, and a suitable sketch $\mathcal{S}_{T_\alpha}$, analogous to $\mathcal{S}_H$, can be obtained from [8]. We postpone the details to the full paper.

Theorem 2. *There is a randomized algorithm for $(k, T_\alpha, \tau, \varepsilon)$ functional monitoring that communicates $\widetilde{O}(k\varepsilon^{-3}\tau^{-3}\log^3 m \log n \log(mn))$ bits.* □

3 Lower Bounds for Non-monotone Functions

We now give lower bounds for estimating entropy, and later, F_p. We give deterministic bounds first, and then randomized bounds. We abuse notation and let H denote both the empirical entropy of a stream and the binary entropy function $H : [0,1] \to [0,1]$ given by $H(x) = -x\log x - (1-x)\log(1-x)$.

Theorem 3. *For any $\varepsilon < 1/2$ and $m \geq k/\sqrt{\varepsilon}$, a deterministic algorithm solving $(k, H, \tau, \varepsilon)$ functional monitoring must communicate $\Omega(k\varepsilon^{-1/2}\log(\varepsilon m/k))$ bits.*

Proof. We use an adversarial argument that proceeds in rounds. Each round, the adversary will force the protocol to send at least one bit. The result will follow by showing a lower bound on the number of rounds r that the adversary can create, using no more than m tokens. Let $\tau = 1$, and let z be the unique positive real such that $H(\frac{z}{2z+1}) = 1 - \varepsilon$. Note that this implies $H(\frac{z}{2z+1}) > 1/2 > H(1/10)$, whence $\frac{z}{2z+1} > 1/10$, hence $z > 1/8$. An estimation of H using calculus shows that $z = \Theta(1/\sqrt{\varepsilon})$. Fix a monitoring protocol $\mathcal{P}$. The adversary only uses tokens from $\{0,1\}$, i.e., the stream will induce a two-point probability distribution.

The adversary starts with a "round 0" in which he sends nine 1s followed by a 0 to site S_1. Note that at the end of round 0, the entropy of the stream is $H(1/10) < 1/2$. For $i \in \{0, 1, \ldots, r\}$, let a_i denote the number of 0s and b_i the number of 1s in the stream at the end of round i. Then $a_0 = 1$ and $b_0 = 9$. For all $i > 0$, the adversary maintains the invariant that $b_i = \lceil a_i(z+1)/z \rceil$. This ensures that at the end of round i, the empirical entropy of the stream is

$$H\left(\frac{a_i}{a_i + b_i}\right) \;\leq\; H\left(\frac{a_i}{a_i(1 + (z+1)/z)}\right) \;=\; H\left(\frac{z}{2z+1}\right) \;=\; 1 - \varepsilon\,,$$

which requires the coordinator to output 0.

Consider the situation at the start of round i, where $i \geq 1$. If each player were to receive $\lceil (b_{i-1} - a_{i-1})/k \rceil$ 0-tokens in this round, then at some point the number of 0s in the stream would equal the number of 1s, which would make the empirical entropy equal to 1 and require the coordinator to change his output to 1. Therefore, there must exist a site S_{j_i}, $j_i \in [k]$, who would communicate upon receiving these many 0-tokens in round i. In actuality, the adversary does the following in round i: he sends these many 0s to S_{j_i}, followed by as many 1s as required to restore the invariant, i.e., to cause $b_i = \lceil a_i(z+1)/z \rceil$. Clearly, this strategy forces at least one bit of communication per round. It remains to bound r from below. Note that the adversary's invariant implies $b_i - a_i \leq a_i/z + 1$ and $a_i + b_i \leq a_i(2z+1)/z + 1 = a_i(2 + 1/z) + 1$. Therefore, we have

$$a_i \;=\; a_{i-1} + \left\lceil \frac{b_{i-1} - a_{i-1}}{k} \right\rceil \;\leq\; a_{i-1} + \left\lceil \frac{1 + a_{i-1}/z}{k} \right\rceil \;\leq\; a_{i-1}\left(1 + \frac{1}{zk}\right) + 2\,.$$

Setting $\alpha = (1 + 1/zk)$ and iterating gives $a_r \le a_0\alpha^r + 2(\alpha^r - 1)/(\alpha - 1) = a_0\alpha^r + 2zk(\alpha^r - 1) = \alpha^r(a_0 + 2zk) - 2zk$. Using our upper bound on $a_i + b_i$, the above inequality, and the facts that $a_0 = 1$ and that $z > 1/8$, we obtain

$$\begin{aligned} a_r + b_r &\le \alpha^r\,(1 + 2zk)\,(2 + 1/z) - 2zk(2 + 1/z) + 1 \\ &\le (2 + 1/z)\,(1 + 2zk)\,\alpha^r \le (2 + 1/z)\,(1 + 2zk)\,e^{r/zk} \le 60zke^{r/zk}\,. \end{aligned}$$

Therefore, we can have $a_r + b_r \le m$, provided $r \le zk\ln(m/(60zk))$. Recalling that $z = \Theta(1/\sqrt{\varepsilon})$, we get the claimed lower bound of $\Omega(k\varepsilon^{-1/2}\log(\varepsilon m/k))$. □

Our next lower bounds are for functional monitoring of frequency moments when we allow for deletions. Specifically, we now consider *update streams* that consist of tokens of the form (i, v), where $i \in [n]$ and $v \in \{-1, 1\}$, to be thought of as updates to a vector $(f_1, \ldots, f_n)$ of frequencies. The vector is initially zero and is updated using $f_i \leftarrow f_i + v$ upon receipt of the token (i, v): in English, each update either adds or deletes one copy of item i.

As usual, we let m denote the length of an update stream whose tokens are distributed amongst several sites. Our next results essentially show that no nontrivial savings in communication is possible for the problem of monitoring frequency moments in this setting. These bounds highlight the precise problem caused by the non-monotonicity of the function being monitored. They should be contrasted with the much smaller upper bounds achievable in the monotone case, when there are no deletions (see Table 1).

Our proofs are again adversarial and proceed in rounds. They use appropriate instantiations of the following generic lemma.

Definition 1. *An update stream is said to be* positive *if it consists entirely of tokens from* $[n] \times \{1\}$*, i.e., insertions only. The* inverse *of an update stream* $\sigma = \langle(i_1, v_1), \ldots, (i_m, v_m)\rangle$ *is defined to be* $\sigma^{-1} := \langle(i_m, -v_m), \ldots, (i_1, -v_1)\rangle$*. A function* $G : \mathbb{Z}_+^n \to \mathbb{R}_+$ *on frequency vectors is said to be* monotone *if* G *is nondecreasing in each parameter, separately. We extend such a* G *to a function on streams (or update streams) in the natural way, and write* $G(\sigma)$ *to denote* $G(\boldsymbol{f})$*, where* $\boldsymbol{f}$ *is the frequency vector determined by* σ*.*

Lemma 4. *Let* $G : \mathbb{Z}_+^n \to \mathbb{R}_+$ *be monotone and let* $\mathcal{P}$ *be a protocol for the* $(k, G, \tau, \varepsilon)$ *functional monitoring problem with deletions allowed. Let* $\sigma_0, \sigma_1, \ldots, \sigma_k$ *be a collection of positive update streams such that (1)* $G(\sigma_0) \le \tau(1 - \varepsilon)$*, and (2)* $G(\sigma_0 \circ \sigma_1 \circ \ldots \circ \sigma_k) \ge \tau$*. If* $\mathcal{P}$ *is a deterministic protocol, then the number of bits communicated is at least* $\lfloor(m - |\sigma_0|)\,/\,(2 \cdot \max_{j\in[k]}\{|\sigma_j|\})\rfloor$*. If* $\mathcal{P}$ *is a* δ*-error randomized protocol, then the expected number of bits communicated is at least* $((1 - \delta)/k) \cdot \lfloor(m - |\sigma_0|)\,/\,(2 \cdot \max_{j\in[k]}\{|\sigma_j|\})\rfloor$*.*

Proof. Let $S_1, \ldots, S_k$ be the k sites involved in $\mathcal{P}$. The adversary will send certain tokens to certain sites, maintaining the invariant that the coordinator is always required to output 0. In round 0, the adversary sends σ_0 to S_1; by condition (1), this maintains the invariant.

Let $s = \max_{j\in[k]}\{|\sigma_j|\}$ and $r = \lfloor(m - |\sigma_0|)/2s\rfloor$. The adversary uses r additional rounds maintaining the additional invariant that at the start of each such round the value of G is $G(\sigma_0)$. Consider round i, where $i \in [r]$. By condition (2), if the adversary were to send σ_j to S_j in this round, for each $j \in [k]$, the coordinator's output would have to change to 1.

Suppose $\mathcal{P}$ is a deterministic protocol. Then, since the coordinator's output would have to change to 1, there must exist a site S_{j_i}, with $j_i \in [k]$, that would have to communicate upon receiving σ_{j_i} in this round. In actuality, the adversary sends $\sigma_{j_i} \circ \sigma_{j_i}^{-1}$ to S_{j_i} and nothing to any other site in round i. Clearly, this maintains both invariants and causes at least one bit of communication. Also, this adds at most $2s$ tokens to the overall input stream. Thus, the adversary can cause r bits of communication using $|\sigma_0| + 2sr \leq m$ tokens in all, which proves the claim for deterministic protocols.

The proof when $\mathcal{P}$ is a δ-error randomized protocol proceeds in a similar manner. The difference is that each round i has an associated collection of probabilities $(p_{i1}, \ldots, p_{ik})$, where $p_{ij} = \Pr[S_j$ communicates in round i upon receiving $\sigma_j]$. As before, condition (2) implies that were each S_j to receive σ_j in this round, correctness would require C's output to change to 1. Thus,

$$1 - \delta \ \leq\ \Pr[\mathcal{P} \text{ is correct}] \ \leq\ \Pr[C \text{ receives a bit in round } i] \ \leq\ \textstyle\sum_{j=1}^{k} p_{ij}\ ,$$

where the final inequality uses a union bound. Therefore, there exists a site S_{j_i}, with $j_i \in [k]$, having $p_{ij_i} \geq (1-\delta)/k$. Again, as in the deterministic case, the adversary actually sends $\sigma_{j_i} \circ \sigma_{j_i}^{-1}$ to S_{j_i} and nothing to any other site in round i. By linearity of expectation, the expected total communication with r rounds is at least $r(1-\delta)/k$, which proves the lemma. □

The theorems that follow are for randomized protocols with error $\delta = 1/3$. We prove only the first of these, leaving the (similar) proofs of the other two to the full paper.

Theorem 4. *The expected communication cost of a randomized* $(k, F_0, \tau, \varepsilon)$ *functional monitoring protocol that allows for deletions is* $\Omega(\min\{m/k, m/\varepsilon\tau\})$. □

Proof. Let $a := \max\{1, \lceil \frac{\tau\varepsilon}{k} \rceil\}$, and instantiate σ_0 as a stream of $\tau - ka$ distinct elements and $\sigma_1, \ldots, \sigma_k$ each as a stream of a distinct elements. Note that $ka \geq \tau\varepsilon$, so $F_0(\sigma_0) = \tau - ka \leq \tau(1-\varepsilon)$. Furthermore, note that $F_0(\sigma_0 \circ \sigma_1 \circ \cdots \circ \sigma_k) = \tau$, hence the streams satisfy the conditions of Lemma 4 with $G = F_0$. Applying that lemma, and noting that $|\sigma_j| = a$ gives us a lower bound of $((1-\delta)/k) \cdot \lfloor (m - |\sigma_0|)/(2a) \rfloor = \Omega(\min\{m/k, m/\varepsilon\tau\})$ for m large enough. □

Theorem 5. *The expected communication cost of a randomized* $(k, F_p, \tau, \varepsilon)$ *monitoring protocol (with* $p > 0$*) that allows deletions is* $\Omega(\min\{m/k, m/\tau^{1/p}\varepsilon\})$. □

Theorem 6. *The expected communication cost of a randomized* $(k, H, \tau, \varepsilon)$ *functional monitoring protocol is* $\Omega(\varepsilon^{-1/2} \log(\varepsilon m/k))$ *bits.* □

We note that the deterministic part of Lemma 4 implies corresponding lower bounds that are k times larger. Of course, $\Omega(m)$ bounds were already known for F_p, even without deletions, by the techniques of [1] for deterministic streaming algorithms. We also note that Yi and Zhang [16] study problems similar to ours but in terms of competitive ratio. The bounds in this section rely on the construction of hard instances which might not be possible in their case.

4 Frequency Moments without Deletions: New Bounds

We finish with another set of lower bounds, this time for monitoring F_p (for various p) without deletions. Our bounds either improve or are incomparable with previous lower bounds: see Table 1.

Theorem 7. *A deterministic protocol that solves $(k, F_1, \tau, \varepsilon)$ functional monitoring must communicate at least $\Omega\big(k \log \frac{k+\tau}{k+\varepsilon\tau}\big)$ bits. In particular, when $\tau \geq k/\varepsilon^{\Omega(1)}$, it must communicate $\Omega\left(k \log(1/\varepsilon)\right)$ bits.*

Proof. Again we use an adversary, who proceeds in rounds: each round, he gives just enough tokens to a single site to force that site to communicate.

Let $a_0 = 0$ and, for $i \geq 1$, let a_i be the total number of tokens received by all sites (i.e., the value of F_1 for the input stream) at the end of round i. The adversary maintains the invariant that $a_i \leq \tau(1-\varepsilon)$, so that the coordinator must always output 0. For $j \in [k]$, let b_{ij} be the maximum number of tokens that site j can receive in round i without being required to communicate. The correctness of the protocol requires $a_{i-1} + \sum_{j=1}^{k} b_{ij} \;\; < \;\; \tau$, for otherwise the desired output can change from 0 to 1 without the coordinator having received any communication. Let $j^* = \operatorname{argmin}_{j \in [k]}\{b_{ij}\}$. In round i, the adversary sends $b_{ij^*} + 1$ tokens to site j^*, forcing it to communicate. We have

$$a_i \;=\; a_{i-1} + b_{ij^*} + 1 \;\leq\; a_{i-1} + \frac{\tau - a_{i-1}}{k} + 1 \;=\; 1 + \frac{\tau}{k} + \left(1 - \frac{1}{k}\right) a_{i-1} \, .$$

Letting $\alpha = 1 - 1/k$ and iterating the above recurrence gives $a_i \leq (1 + \tau/k)(1 - \alpha^i)/(1-\alpha) = (k+\tau)(1-\alpha^i)$. Now note that $\alpha \geq e^{-2/k}$, so when $i \leq r := \frac{k}{2} \ln \frac{k+\tau}{k+\varepsilon\tau}$, we have $\alpha^i \geq \frac{k+\varepsilon\tau}{k+\tau}$, so that $a_i \leq (\tau+k)\cdot(k+\tau-k-\varepsilon\tau/(k+\tau) = \tau(1-\varepsilon)$.

This shows that the adversary can maintain the invariant for up to r rounds, forcing $\Omega(r)$ bits of communication, as claimed. □

Our next lower bounds use reductions from a fundamental problem in communication complexity: the "gap Hamming distance" problem, denoted GHD_c, where $c \in \mathbb{R}_+$ is a parameter. In this problem, Alice and Bob are given $x, y \in \{0,1\}^n$ respectively and want to output 1 if $\Delta(x,y) \geq \frac{n}{2} + c\sqrt{n}$ and 0 if $\Delta(x,y) \leq \frac{n}{2} - c\sqrt{n}$; they don't care what happens if the input satisfies neither of these conditions. We shall need the following lower bounds on the randomized communication complexity $\mathrm{R}(\mathrm{GHD}_c)$, as well as the one-way randomized communication complexity (where the only communication is from Alice to Bob) $\mathrm{R}^{\rightarrow}(\mathrm{GHD}_c)$. Proofs of these bounds, as well as further background on the problem, can be found in Woodruff [15].

Theorem 8. *Suppose $c > 0$ is a constant. Then $\mathrm{R}(\mathrm{GHD}_c) = \Omega(\sqrt{n})$ and $\mathrm{R}^{\rightarrow}(\mathrm{GHD}_c) = \Omega(n)$. Here, the Ω notation hides factors dependent upon c.*[5] □

[5] The bounds in [15] restrict the range of c, but this turns out not to be necessary.

It is conjectured that the general randomized bound is in fact as strong as the one-way version. This is not just a tantalizing conjecture about a basic communication problem. Settling it would have important consequences because, for instance, the gap Hamming distance problem is central to a number of results in streaming algorithms. As we shall soon see, it would also have consequences for our work here.

Conjecture 1. For sufficiently small constants c, we have $\mathrm{R}(\mathrm{GHD}_c) = \Omega(n)$.

Theorem 9. *For any $\varepsilon \le 1/2$, a randomized protocol for $(k, F_0, \tau, \varepsilon)$ functional monitoring must communicate $\Omega(1/\varepsilon)$ bits.*

Proof. We give a reduction from GHD_1. Let $\mathcal{P}$ be a randomized protocol for $(k, F_0, \tau, \varepsilon)$ functional monitoring. Set $N := \lfloor 1/\varepsilon^2 \rfloor$ and $\tau = 3N/2 + \sqrt{N}$. We design a two-party public coin randomized communication protocol $\mathcal{Q}$ for GHD_1 on N-bit inputs that simulates a run of $\mathcal{P}$ involving the coordinator, C, and two sites, S_1 and S_2. Let $x \in \{0,1\}^N$ be Alice's input in $\mathcal{Q}$ and let $y \in \{0,1\}^N$ be Bob's input. Alice creates a stream $\sigma_a := \langle a_1, \ldots, a_N \rangle$ of tokens from $[N] \times \{0,1\}$ by letting $a_i := (i, x_i)$ and Bob similarly creates a stream $\sigma_b := \langle b_1, \ldots, b_N \rangle$, where $b_i := (i, y_i)$. They then simulate a run of $\mathcal{P}$ where S_1 first receives all of σ_a after which S_2 receives all of σ_b. They output whatever the coordinator would have output at the end of this run.

The simulation itself occurs as follows: Alice maintains the state of S_1, Bob maintains the state of S_2, and they *both* maintain the state of C. Clearly, this can be done by having Alice send to Bob all of S_1's messages to C plus C's messages to S_2 (and having Bob act similarly). The total communication in $\mathcal{Q}$ is at most that in $\mathcal{P}$.

We now show that $\mathcal{Q}$ is correct. By construction, the combined input stream $\sigma = \sigma_a \circ \sigma_b$ seen by $\mathcal{P}$ has $2\Delta(x,y)$ tokens with frequency 1 each and $N - \Delta(x,y)$ tokens with frequency 2 each. Therefore $F_0(\sigma) = N + \Delta(x,y)$. When $\Delta(x,y) \ge N/2 + \sqrt{N}$, we have $F_0(\sigma) \ge \tau$ and $\mathcal{Q}$, following $\mathcal{P}$, correctly outputs 1. On the other hand, when $\Delta(x,y) \le N/2 - \sqrt{N}$, we have

$$F_0(\sigma) \;\le\; \frac{3N}{2} - \sqrt{N} \;=\; \tau\left(1 - \frac{2\sqrt{N}}{3N/2 + \sqrt{N}}\right) \;\le\; \tau\left(1 - \frac{1}{\sqrt{N}}\right) \;\le\; \tau(1-\varepsilon)\,.$$

Thus $\mathcal{Q}$ correctly outputs 0. Since $\mathcal{Q}$ is correct, by Theorem 8, it must communicate at least $\Omega(\sqrt{N}) = \Omega(1/\varepsilon)$ bits. Therefore, so must $\mathcal{P}$. □

Theorem 10. *For any $\varepsilon < 1/2$ and any constant $p > 1$, a randomized protocol for $(k, F_p, \tau, \varepsilon)$ functional monitoring must communicate $\Omega(1/\varepsilon)$ bits.*

Proof. For simplicity, we assume here that $p \ge 2$. As before, we reduce from GHD_1 on $N := \lfloor 1/\varepsilon^2 \rfloor$-bit inputs. For this reduction, we set $\tau := (N/2 + \sqrt{N})2^p + (N - 2\sqrt{N})$. Let $\mathcal{P}$ be a protocol for $(k, F_p, \tau, \varepsilon)$ functional monitoring. We design a protocol $\mathcal{Q}$ for GHD_1 on input (x,y) that simulates a run of $\mathcal{P}$ involving two sites, creating two streams $\langle (i, x_i) \rangle_{i \in [N]}$ and $\langle (i, y_i) \rangle_{i \in [N]}$, exactly as before; however, in this reduction, the output of $\mathcal{Q}$ is the *opposite* of the coordinator's output at the end of the run of $\mathcal{P}$.

We now show that $\mathcal{Q}$ is correct. The input stream σ seen by $\mathcal{P}$ has the same frequency distribution as before, which means that $F_p(\sigma) = 2\Delta(x,y) + (N - \Delta(x,y)) \cdot 2^p = N \cdot 2^p - \Delta(x,y)(2^p - 2)$. When $\Delta(x,y) \le N/2 - \sqrt{N}$, we have

$$F_p(\sigma) \;\ge\; N \cdot 2^p - (N/2 - \sqrt{N})(2^p - 2) \;=\; (N/2 + \sqrt{N})2^p + (N - 2\sqrt{N}) \;=\; \tau\,.$$

Therefore $\mathcal{P}$ outputs 1, which means $\mathcal{Q}$ correctly outputs 0. On the other hand, when $\Delta(x,y) \ge N/2 + \sqrt{N}$, we have

$$\begin{aligned} F_p(\sigma) \;&\le\; N \cdot 2^p - (N/2 + \sqrt{N})(2^p - 2) \\ =\; \tau &\left(1 - \frac{2\sqrt{N}2^p - 4\sqrt{N}}{(N/2 + \sqrt{N}) \cdot 2^p + (N - 2\sqrt{N})}\right) \;\le\; \tau(1 - 1/\sqrt{N}) \;\le\; \tau(1 - \varepsilon)\,, \end{aligned}$$

where the penultimate inequality uses $p \ge 2$. Therefore $\mathcal{P}$ outputs 0, whence $\mathcal{Q}$ correctly outputs 1. Theorem 8 now implies that $\mathcal{Q}$, and hence $\mathcal{P}$, must communicate $\Omega(\sqrt{N}) = \Omega(1/\varepsilon)$ bits. □

We remark that if Conjecture 1 holds (for a favorable c), then the lower bounds in Theorems 9 and 10 would improve to $\Omega(1/\varepsilon^2)$. This further strengthens the motivation for settling the conjecture.

We also consider a restricted, yet natural, class of protocols that we call *round-based* protocols; the precise definition follows. Note that all nontrivial protocols in [4] are round-based, which illustrates the naturalness of this notion.

Definition 2. *A* round-based *protocol for* $(k, f, \tau, \varepsilon)$ *functional monitoring is one that proceeds in a series of rounds numbered* $1, \ldots, r$. *Each round has the following four stages. (1) Coordinator* C *sends messages to the sites* S_i*, based on the past communication history. (2) Each* S_i *read its tokens and sends messages to* C *from time to time, based on these tokens and the Stage* 1 *message from* C *to* S_i*. (3) At some point, based on the messages it receives,* C *decides to end the current round by sending a special, fixed, end-of-round message to each* S_i*. (4) Each* S_i *sends* C *a final message for the round, based on all its knowledge, and then resets itself, forgetting all previously read tokens and messages.*

It is possible to improve the lower bounds above by restricting to round-based protocols, as in Definition 2. The key is that if the functional monitoring protocol $\mathcal{P}$ in the proofs of Theorems 9 and 10 is round-based, then the corresponding communication protocol $\mathcal{Q}$ only requires messages from Alice to Bob. This is because Alice can now simulate the coordinator C and *both* sites S_1 and S_2, during $\mathcal{P}$'s processing of σ_a: she knows that S_2 receives no tokens at this time, so she has the information needed to compute any messages that S_2 might need to send. Consider the situation when Alice is done processing her tokens. At this time the Stage 4 message (see Definition 2) from S_1 to C in the current round has been determined, so Alice can send this message to Bob. From here on, Bob has all the information needed to continue simulating S_1, because he knows that S_1 receives no further tokens. Thus, Bob can simulate $\mathcal{P}$ to the end of the run.

Theorem 11. *Suppose p is either 0 or a constant greater than 1. For any $\varepsilon \leq 1/2$, a round-based randomized protocol for $(k, F_p, \tau, \varepsilon)$ functional monitoring must communicate $\Omega(1/\varepsilon^2)$ bits.*

Proof. We use the observations in the preceding paragraph, proceed as in the proofs of Theorems 9 and 10 above, and plug in the one-way communication lower bound from Theorem 8. □

References

1. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. J. Comput. Syst. Sci. 58(1), 137–147 (1999); Preliminary version In: Proc. 28th Annu. ACM Symp. Theory Comput., pp. 20–29 (1996)
2. Babcock, B., Olston, C.: Distributed top-k monitoring. In: Proc. Annual ACM SIGMOD Conference, pp. 28–39 (2003)
3. Bhuvanagiri, L., Ganguly, S.: Estimating entropy over data streams. In: Proc. 14th Annual European Symposium on Algorithms, pp. 148–159 (2006)
4. Cormode, G., Muthukrishnan, S., Yi, K.: Algorithms for distributed functional monitoring. In: Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1076–1085 (2008)
5. Cormode, G., Muthukrishnan, S., Zhuang, W.: What's different: Distributed, continuous monitoring of duplicate-resilient aggregates on data streams. In: Proc. 22nd International Conference on Data Engineering, p. 57 (2006)
6. Das, A., Ganguly, S., Garofalakis, M.N., Rastogi, R.: Distributed set expression cardinality estimation. In: Proc. 30th International Conference on Very Large Data Bases, pp. 312–323 (2004)
7. Estrin, D., Govindan, R., Heidemann, J.S., Kumar, S.: Next century challenges: Scalable coordination in sensor networks. In: MOBICOM, pp. 263–270 (1999)
8. Harvey, N.J.A., Nelson, J., Onak, K.: Sketching and streaming entropy via approximation theory. In: Proc. 49th Annual IEEE Symposium on Foundations of Computer Science, pp. 489–498 (2008)
9. Muthukrishnan, S.: Data streams: Algorithms and applications. In: Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithms, p. 413 (2003)
10. Muthukrishnan, S.: Some algorithmic problems and results in compressed sensing. In: Proc. 44th Annual Allerton Conference (2006)
11. Newman, I.: Private vs. common random bits in communication complexity. Information Processing Letters 39(2), 67–71 (1991)
12. Sharfman, I., Schuster, A., Keren, D.: A geometric approach to monitoring threshold functions over distributed data streams. ACM Trans. Database Syst. 32(4) (2007)
13. Slepian, D., Wolf, J.K.: Noiseless coding of correlated information sources. IEEE Trans. Inf. Theory 19(4), 471–480 (1973)
14. Tsallis, C.: Possible generalization of Boltzmann-Gibbs statistics. J. Stat. Phys. 52, 479–487 (1988)
15. Woodruff, D.P.: Efficient and Private Distance Approximation in the Communication and Streaming Models. PhD thesis, MIT (2007)
16. Yi, K., Zhang, Q.: Multi-dimensional online tracking. In: Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1098–1107 (2009)