Streaming Algorithms with One-Sided Estimation

Joshua Brody[1,⋆] and David P. Woodruff[2]

[1] IIIS, Tsinghua University
joshua.e.brody@gmail.com
[2] IBM Research-Almaden
dpwoodru@us.ibm.com

Abstract. We study the space complexity of randomized streaming algorithms that provide one-sided approximation guarantees; e.g., the algorithm always returns an overestimate of the function being computed, and with high probability, the estimate is not too far from the true answer. We also study algorithms which always provide underestimates.

We also give lower bounds for several one-sided estimators that match the deterministic space complexity, thus showing that to get a space-efficient solution, two-sided approximations are sometimes necessary. For some of these problems, including estimating the longest increasing sequence in a stream, and estimating the Earth Mover Distance, these are the first lower bounds for randomized algorithms of any kind.

We show that for several problems, including estimating the radius of the Minimum Enclosing Ball (MEB), one-sided estimation is possible. We provide a natural function for which the space for one-sided estimation is asymptotically less than the space required for deterministic algorithms, but more than what is required for general randomized algorithms.

What if an algorithm has a one-sided approximation from both sides? In this case, we show the problem has what we call a Las Vegas streaming algorithm. We show that even for two-pass algorithms, a quadratic improvement in space is possible and give a natural problem, counting non-isolated vertices in a graph, which achieves this separation.

1 Introduction

Computing on data streams is of growing interest in many areas of computer science, such as databases, networks, and algorithm design. Here it is assumed that the algorithm sees updates to elements of an underlying object one by one in an arbitrary order, and needs to output certain statistics of the input. Therefore

⋆ Supported in part by the National Basic Research Program of China Grant 2007CB807900, 2007CB807901, and the National Natural Science Foundation of China Grant 61033001, 61061130540, 61073174. Also supported in part from the Danish National Research Foundation and the National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part this work was performed.

L.A. Goldberg et al. (Eds.): APPROX/RANDOM 2011, LNCS 6845, pp. 436–447, 2011.

it must maintain a short summary or sketch of what it has seen. We refer the reader to the survey by Muthukrishnan [17] for a list of applications.

In this paper, we consider the space complexity of streaming algorithms which return estimates with *one-sided approximation*—either the streaming algorithm always returns an overestimate, or it always returns an underestimate. As with the case of standard randomized streaming algorithms, we want the algorithm to return an accurate estimate with high probability. While one-sided approximation has been extensively studied in the property testing literature, it has not been considered as an object of study for streaming algorithms.

Definition 1.1. *An ε-overestimator for f is a randomized algorithm that, given a stream σ returns $\hat{f}(\sigma)$ such that*

- $\hat{f}(\sigma) \geq f(\sigma)$.
- *With probability at least* $2/3$, $\hat{f}(\sigma) \leq f(\sigma)(1+\varepsilon)$.

An ε-underestimator for f is a randomized algorithm that returns an underestimate $\hat{f}(\sigma)$ such that with probability at least $2/3$, *we have* $\hat{f}(\sigma) \geq f(\sigma)(1-\varepsilon)$.

An important class of one-sided approximations are problems where the *information lost* by using a small amount of space is one-sided. Perhaps the best known example in this class is the Count-Min sketch [5], which is used to maintain approximate frequency counts and can produce accurate estimations of ϕ-quantiles or ϕ-heavy hitters. The Count-Min sketch essentially works by maintaining a random hash table h of counters and updating the counter in bucket $h(i)$ each time item i is seen on the stream. The counter in bucket $h(i)$ then provides an *overestimate* of the true frequency of item i, since collisions can only increase the count. By maintaining several hash tables $h_1, h_2, \ldots, h_t$ and returning the minimum $h_j(i)$ over all j, the Count-Min sketch gets an overestimate of the frequency of item i that with high probability remains close to the true frequency. Since its inception, the Count-Min sketch has also been used as a subroutine in several other applications.

Surprisingly, the Count-Min sketch is also used to generate ε-underestimators. In the k-median problem, the input is a set of points P on a discrete grid $[\Delta]^d$, and the goal is to output a set of k points Q that minimizes $C(Q,P) := \sum_{p\in P} \min_{q\in Q} ||p-q||$. Such a set is called a k-median. Indyk [12] uses a Count-Min sketch to *underestimate* $C(P,Q)$.

We are interested in the space complexity of one-sided approximations and how this space complexity relates to the complexity of randomized and deterministic algorithms that give two-sided approximations. We also study what happens when both underestimates and overestimates are possible. By properly scaling the one-sided estimates, we can get an algorithm that provides a $(1\pm\varepsilon)$-approximation with high probability, and **knows** when its estimate is a poor approximation. We call such algorithms Las Vegas algorithms.

Definition 1.2. *A Las Vegas algorithm for f is a randomized streaming algorithm that, given a stream σ either returns $\hat{f}(\sigma)$ such that $|\hat{f}(\sigma) - f(\sigma)| \leq \varepsilon f(\sigma)$ or outputs* FAIL. *The algorithm* FAIL*s with probability at most* $1/3$.

Remark 1.1. Las Vegas algorithms can alternatively be thought of as multipass algorithms that never fail; instead, they repeat until accepting an estimate. That notion corresponds more with the concept of Las Vegas algorithms used in communication complexity. Our definition has meaning even for one-pass algorithms.

1.1 Our Problems

We consider the space complexity of streaming algorithms under several models: two-sided $(1 \pm \varepsilon)$-approximations, ε-overestimates, ε-underestimates, Las Vegas algorithms, and deterministic algorithms. Let $\mathrm{S}_{1\pm\varepsilon}(f), \mathrm{S}_{\varepsilon\text{-under}}(f)$, and $\mathrm{S}_{\varepsilon\text{-over}}(f)$ denote the space complexity of two-sided estimators, ε-underestimators, and ε-overestimators. $\mathrm{S}_{LV}(f)$ and $\mathrm{S}_{det}(f)$ denote the space complexity of Las Vegas and deterministic algorithms that compute f exactly; $\mathrm{S}_{\varepsilon,LV}(f)$ and $\mathrm{S}_{\varepsilon,det}(f)$ are the complexity of Las Vegas and deterministic algorithms that return $(1 \pm \varepsilon)$-approximations. The relationship between these measures is captured in the following lemma, which we prove in Section 3.

Lemma 1.1. *For any f, the space complexities are characterized (up to small changes in ε) by the following:*

$$\begin{aligned}\mathrm{S}_{1\pm\varepsilon}(f) &\leq \min\{\mathrm{S}_{\varepsilon\text{-}under}(f), \mathrm{S}_{\varepsilon\text{-}over}(f)\} \\ &\leq \max\{\mathrm{S}_{\varepsilon\text{-}under}(f), \mathrm{S}_{\varepsilon\text{-}over}(f)\} = \Theta(\mathrm{S}_{\varepsilon,LV}(f)) \leq \mathrm{S}_{\varepsilon,det}(f)\ .\end{aligned}$$

Our next collection of results provides strict separations for these inequalities.

Cascaded Norms. In Section 3, we consider the problem of estimating the cascaded norm $\ell_0(Q)(A)$ in a stream of updates to an $n \times n$ matrix A. Here, $\ell_0(Q)(A)$ is the number of non-zero rows of A. We show that two-sided approximations are possible in $\mathrm{poly}(\log(n)/\varepsilon)$ space; an ε-overestimate is possible in $\tilde{O}(n)$ space, and $\Omega(n^2)$ space is required for deterministic algorithms.

Theorem 1.1. *For the problem of estimating $\ell_0(Q)(A)$ in the streaming model, the following bounds hold: (i) $\mathrm{S}_{1\pm\varepsilon}(\ell_0(Q)) = \tilde{O}(1)$, (ii) $\mathrm{S}_{\varepsilon\text{-}over}(\ell_0(Q)) = \tilde{\Theta}(n)$, and $\mathrm{S}_{\varepsilon,det}(\ell_0(Q)) = \Omega(n^2)$.*[1]

This problem also corresponds to estimating the number of non-isolated vertices in a graph[8] and can be useful for counting outliers in social networks.

Earth Mover Distance. In this problem, the elements on the stream define two point sets $A, B \subseteq [\Delta^2]$, and the algorithm should estimate the cost of the best matching between A and B. In Section 3, we show

Theorem 1.2. *For all constant ε, $\mathrm{S}_{\varepsilon\text{-}under}(EMD) = \mathrm{S}_{\varepsilon\text{-}over}(EMD) = \Omega(\Delta^2)$. Moreover, these bounds hold even for underestimators that return a value that is at least $1/c \cdot EMD$ or overestimators that return a value that is at most $c \cdot EMD$ with constant probability, for any constant $c > 1$.*

This is the first lower bound for EMD for any class of randomized algorithms. A result of Andoni et al. [1] gives a c-approximation in $\Delta^{O(1/c)}$ space for any $c > 1$, and so this separates the complexity of one-sided and two-sided estimations.

[1] The $\tilde{O}(\cdot)$ notation hides terms polynomial in $\log n$ and ε.

List Equality. To separate the deterministic and Las Vegas space complexities, we adapt a problem of Mehlhorn and Schmidt [15] to the streaming setting. The problem is called List-Equality. The inputs are two lists of n-bit numbers $X, Y \in (\{0,1\}^n)^n$, and the goal is to compute $\text{ListEQ}(X,Y) := \vee_{i=1}^n \text{EQ}(X_i, Y_i)$. Mehlhorn and Schmidt [15] introduced this problem and use it to show a quadratic separation between the deterministic and Las Vegas versions of communication complexity. In the streaming version of this problem, X, Y appear sequentially on a stream of n^2 bits. We give a $\tilde{O}(n)$ space Las Vegas algorithm; an $\Omega(n^2)$ bound follows from [15].

Theorem 1.3. *For the* List-Equality *problem in the streaming model, we have* $\text{S}_{LV}(\text{ListEQ}) = \tilde{O}(n)$*, while* $\text{S}_{det}(\text{ListEQ}) = \Omega(n^2)$.

In addition to the space complexity separations, in Section 4 we give new one-sided estimators for two problems motivated by machine learning: the Minimum Enclosing Ball and Classification problems. These problems were studied in the streaming setting by Clarkson et al. [4] who gave efficient two-sided estimates for both problems. We extend their work to give one-sided estimates.

In Section 5, we give lower bounds for one-sided estimates for a large range of problems, including estimating the length of the longest increasing subsequence (LIS), the ℓ_p-norms and ℓ_p-heavy hitters, and the empirical entropy of a stream.

We also discuss open questions in Section 5.

2 Preliminaries

For many of the problems we consider, the stream is a sequence of m tokens $(i_1, v_1), \ldots, (i_m, v_m) \in [n] \times \{-M, \ldots, M\}$ interpreted as updates to a frequency vector $z \in \mathbb{N}^n$, where a token (i, v) causes $z_i \leftarrow z_i + v$. In these problems we implicitly associate the frequency vector z with the corresponding stream σ. In an *insertion-only* stream, v_i is always positive. In the *strict turnstile* model, the current value of z_i is always positive, though some of the v_i may be negative. The *general turnstile* model allows arbitrary z_i.

Given $z \in \mathbb{R}^m$, the ℓ_p-norm of z is defined as $||z||_p := \left(\sum_{i=1}^m |z_i|^p\right)^{1/p}$. The pth frequency moment is $F_p(z) := ||z||_p^p = \sum_{i=1}^m |z_i|^p$. We use $\delta(x, y)$ to denote the Hamming distance between strings x and y, that is, the number of positions that differ in x and y.

In rest of this section, we briefly describe the basic terminology and notation we need for communication complexity, as well as the problems we use to prove our streaming lower bounds. For a more complete treatment, we refer the reader to the standard text by Kushilevitz and Nisan [14].

Given a boolean function $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, let $R_\varepsilon(f)$ denote the minimum communication cost of a public-coin randomized protocol P such that on all inputs, $P(x, y) = f(x, y)$ with probability at least $1 - \varepsilon$. We are particularly interested in the communication complexity of protocols with one-sided error. For $b \in \{0,1\}$, let $R_\varepsilon^b(f)$ be the cost of the best randomized protocol P for f such that (i) when $f(x, y) \neq b$, P correctly computes $f(x, y)$, and (ii) when $f(x, y) = b$,

P computes $f(x, y)$ with probability $\geq 1 - \varepsilon$. We usually take $\varepsilon := 1/3$; in this case, we drop the subscript.

Next we describe two problems we use extensively to show our bounds. In the EQUALITY problem, Alice and Bob receive n-bit strings x, y and wish to compute $\mathrm{EQ}(x, y) = 1$ iff $x = y$. The standard EQ test gives $R^0_\varepsilon(\mathrm{EQ}) = O(\log(1/\varepsilon))$; in contrast, we have $R^1(\mathrm{EQ}) = \Omega(n)$. In essence, protocols which must be correct when $x \neq y$ are as hard as the deterministic case. When making reductions in this case, we'll often describe the problem as NEQ to emphasize that the protocol must be correct on $x \neq y$ instances.

Our second problem is the promise problem GAP-$\mathrm{EQ}_{n,t}$. Here, Alice and Bob receive n-bit strings under the promise that either $x = y$ or $\delta(x, y) = t$ and output 1 iff $x = y$. Using a combinatorial result of Frankl and Rödl [9], Buhrman et al. [3] proved that $R^1(\text{GAP-EQ}_{n,t}) = \Omega(n)$ for all $t = \Theta(n)$ and used it to get seperations between classical and quantum communication complexity. We supress the subscripts when n is clear from context and $t = n/2$.

3 Space Complexity Separations

In this section, we develop separations between the space complexities for different classes of streaming algorithms.

Lemma 3.1 (Restatement of Lemma 1.1). *For any f, the space complexities are characterized (up to small changes in ε) by the following inequality*

$$\begin{aligned} \mathrm{S}_{1\pm\varepsilon}(f) &\leq \min\{\mathrm{S}_{\varepsilon\text{-}under}(f), \mathrm{S}_{\varepsilon\text{-}over}(f)\} \\ &\leq \max\{\mathrm{S}_{\varepsilon\text{-}under}(f), \mathrm{S}_{\varepsilon\text{-}over}(f)\} = \Theta(\mathrm{S}_{\varepsilon,LV}(f)) \leq \mathrm{S}_{\varepsilon,det}(f) \ . \end{aligned}$$

Proof. The inequalities are trivial inclusions. To prove the equality, fix an ε-underestimator $\mathcal{A}_U$ and an ε-overestimator $\mathcal{A}_O$, and create a Las Vegas algorithm in the following way: Run $\mathcal{A}_U$ and $\mathcal{A}_O$ in parallel, scale the underestimator by $(1 + \varepsilon)$ and the overestimator by $(1 - \varepsilon)$, and FAIL if the scaled underestimate remains less than the scaled overestimate. If the algorithm accepts, return the geometric mean of the estimates. This algorithm accepts with high probability, since it accepts whenever both estimators return good ranges. Furthermore, it's easy to show that when it accepts, the algorithm returns a $(1\pm\varepsilon)$-approximation.

We provide strict separations for each of the inequalities in Lemma 3.1. Our first separation result is for the problem of estimating Cascaded Norms.

Many streaming papers have focused on *single-attribute* aggregation, such as norm estimation. Most applications however deal with multi-dimensional data where the real insights are obtained by slicing the data several times and applying several aggregations in a cascaded fashion. A *cascaded aggregate* $P \circ Q$ of a matrix is defined by evaluating aggregate Q repeatedly over each row of the matrix, and then evaluating aggregate P over results obtained from each row. A well-studied aggregate is the so-called cascaded norm problem on numerical data, for which we first compute the Q norm of each row, then the P norm of the vector of values

obtained, for arbitrary norms P and Q. These were introduced by Cormode and Muthukrishnan [6], and studied in several followup works [16,2,13,1], with particular attention to the case when $P = \ell_p$ and $Q = \ell_q$. In the streaming model, the underlying matrix is initialized to 0, and receives multiple updates in the form of increments and decrements to its entries in an arbitrary order.

One special case of this problem is $\ell_0(Q)$, which corresponds to the number of non-zero rows in an $n \times d$ matrix A. This problem was studied in [13], where the authors obtained a $\text{poly}(\log(nd)/\varepsilon)$ space randomized algorithm for $(1 \pm \varepsilon)$-approximation. This measure is important since it corresponds to estimating the number of non-isolated vertices in a graph. This follows by taking $d = n$ and viewing the matrix A as the adjacency matrix of a graph. Its complement, $n - \ell_0(Q)$, is the number of isolated vertices and may be useful for counting outliers in social networks. This was studied in a sampling (a special case of streaming) context in, e.g., [8].

The following theorem characterizes the space complexity of the different estimators for $\ell_0(Q)$.

Theorem 3.1 (Restatement of Theorem 1.1). *The problem of estimating the cascaded norm $\ell_0(Q)$ in the general turnstile model has the following space complexities:*

1. *There exists a $(1 \pm \varepsilon)$-approximation that uses $O(\text{poly}(\log(nd)/\varepsilon))$ space.*
2. *There is an ε-underestimator for $\ell_0(Q)$ that uses $O(n\,\text{poly}(\log(nd)/\varepsilon))$ space.*
3. *Any ε-underestimator for $\ell_0(Q)$ requires $\Omega(n)$ space.*
4. *Any ε-overestimator for $\ell_0(Q)$ requires $\Omega(nd)$ space.*
5. *Any deterministic approximation for $\ell_0(Q)$ requires $\Omega(nd)$ space.*

Proof. The upper bound for $(1 \pm \varepsilon)$-approximation comes from Jayram and Woodruff [13]. To get an upper bound for ε-underestimators, let u be the maximal possible element in the matrix. We assume that u is polynomially related to n, d and the length of the stream. Next, choose a prime $q = \Theta(und)$, and let V be the $q \times d$ Vandermonde matrix, where the ith row $V_i = (1, i, i^2, \ldots, i^{d-1}) \in GF(q)^d$. It's well known that any d rows of V are linearly independent. It follows that for any nonzero A_i, at most $d - 1$ rows v of V have $\langle A_i, v\rangle = 0 (\text{mod } q)$.

We estimate $\ell_0(Q)(A)$ by picking a random row of V, computing the inner product $\langle A_i, v\rangle$ in $GF(q)$ for each row i of A, and returning the number of rows that give $\langle A_i, v\rangle = 0$. It's easy to see that each inner product can be maintained in $O(\log(u))$ space. Furthermore, we always underestimate the number of nonzero rows, and for a random v, $\langle A_i, v\rangle = 0$ with probability at most $O(q/d) = O(1/n)$. By the union bound, our choice of v identifies *all* nonzero rows with probability 9/10. Therefore, we always underestimate the number of rows with nonzero entries, and with high probabilty we compute $\ell_0(Q)(A)$ exactly. The space required is $O(\log(nd))$ to store a pointer to v, and $n \log u$ to maintain $\langle A_i, v\rangle$ for each row i.

On the other hand, a reduction from Gap-EQ gives an $\Omega(n)$ lower bound for ε-underestimators. Specifically, fix $d := 1$, and given n-bit strings x, y, Alice converts each bit x_i of her input into a token $(i, 1, 1 - x_i)$. Bob converts each

bit of y_i of his input into a token $(i, 1, -y_i)$. They then simulate the algorithm for underestimating $\ell_0(Q)$ on the resulting matrix A and output NO when the estimate is at most $n/2$. Note that $\ell_0(Q)(A) = n$ when $x = y$ and $\ell_0(Q)(A) = n/2$ when $\delta(x, y) = n/2$, hence an ε-underestimator for $\ell_0(Q)$ always produces a correct answer for $\delta(x, y) = n/2$, and with high probability produces a correct answer for the $x = y$ case.

For ε-overestimators, a stronger lower bound is possible, via reduction from NEQ on strings of length nd. Each coordinate in the string maps to an entry in the matrix. Alice maps each $x_{i,j} \to (i, j, x_{i,j})$, and Bob maps each $y_{i,j} \to (i, j, -y_{i,j})$. Thus, $\ell_0(Q)(A) > 0$ iff $x \neq y$. Alice and Bob then compute $\text{NEQ}(x, y)$ by simulating an ε-overestimator for $\ell_0(Q)(A)$ and outputting $x \neq y$ whenever it returns a positive value. This also implies the $\Omega(nd)$ deterministic bound.

Next, we prove lower bounds for estimating Earth Mover Distance. The Earth Mover Distance between multisets $A, B \subseteq [\Delta]^2$ is the cost of the best matching between A and B. Formally, we define

$$\text{EMD}(A, B) = \min_{\pi: A \to B} \sum_{a \in A} ||a - \pi(a)|| \ .$$

Andoni et al. [1] gave a 1-pass, $\Delta^{O(1/c)}$-space algorithm that returns $\widehat{EMD}$ such that $\text{EMD}(A, B)/c \leq \widehat{EMD}(A, B) \leq c\text{EMD}(A, B)$. In general, this approximation factor c can be much greater than 1; for this reason, we refer to results in this section as c-approximations instead of ε-approximations.

Proof (of Theorem 1.2). Partition $[\Delta]^2$ into $n := \Delta^2/2$ pairs of adjacent points $\{(p_{i,0}, p_{i,1}) : 1 \leq i \leq n\}$. The nature of this construction is immaterial; we only require that the pairs of points are adjacent.

To get the lower bound for c-overestimators, we reduce from NEQ. Given $x, y \in \{0, 1\}^n$, Alice creates a set of points $A := \{a_1, \ldots, a_n\}$ by mapping each coordinate $x_i \to p_{i,x_i} =: a_i$. Bob similarly creates $B := \{b_1, \ldots, b_n\}$ by mapping $y_i \to p_{i,y_i} =: b_i$. Then, Alice and Bob simulate a c-overstimating algorithm for EMD and output $x \neq y$ if $\widehat{EMD}(A, B) > 0$.

Note that if $x \neq y$ then clearly $EMD(A, B) > 0$, and since the streaming algorithm returns an overestimate, Alice and Bob will always correctly compute $x \neq y$. Furthermore, when $x = y$, then $EMD(A, B) = 0$; hence, the overestimator will output $\widehat{EMD}(A, B) \leq c\text{EMD}(A, B) = 0$ with high probability. In this way, a c-overestimator for EMD gives a protocol for NEQ. Since $R^1(\text{NEQ}) = \Omega(n) = \Omega(\Delta^2)$, the lower bound for c-overstimators follows.

To get a lower bound for c-underestimators, set $\gamma := 1 - 1/2c$, and reduce from $\text{GAP-EQ}_{n,\gamma n}$. As in the lower bound for overestimators, Alice and Bob map their inputs x, y to pointsets A, B. This time, Alice again sets $a_i := p_{i,x_i}$, but Bob creates $b_i := p_{i,1-y_i}$ Then they simulate a c-underestimator for EMD and output $\delta(x, y) = \gamma n$ if $\widehat{EMD}(A, B) \leq n(1 - \gamma)$.

Essentially, Alice and Bob solve Gap-EQ by using the EMD algorithm to estimate $\delta(x, -y)$. Note that

$$\mathrm{EMD}(A, B) = \begin{cases} n & \text{if } x = y \ , \\ n(1-\gamma) & \text{if } \delta(x, y) = \gamma n \ . \end{cases}$$

Since $\mathrm{EMD}(A, B) = n(1-\gamma)$ when $\delta(x, y) = \gamma n$, a c-underestimator always returns a correct value for $\delta(x, y) = \gamma n$. When $x = y$, the c-understimator returns $\widehat{EMD}(A, B) \geq n/c > n(1-\gamma)$ with high probability. Hence, the $\Omega(\Delta^2)$ lower bound follows from the $\Omega(n)$ lower bound on $\text{Gap-EQ}_{n,\gamma n}$.

We end this section with a two-pass Las Vegas algorithm for List-Equality.

Proof (of Theorem 1.3). We convert the two player communication protocol of Mehlhorn and Schmidt [15] to work in a Las Vegas environment. In the first pass, the algorithm uses an r-bit Equality test to compare X_i and Y_i for each i. Let I be the set of indices i that pass this test. If I is empty, then the algorithm outputs $\mathrm{ListEQ}(X, Y) = 0$. Otherwise, if $|I| > m$, let I' be a random m-subset of I. In the second pass, the algorithm saves X_i for each $i \in I'$ and compares X_i, Y_i directly. If it finds any i such that $X_i = Y_i$, then the algorithm outputs $\mathrm{ListEQ}(X, Y) = 1$. Otherwise, the algorithm outputs fail.

This algorithm uses nr space to maintain the n equality tests, nm space to store X_i for up to m indices $i \in I'$, and $O(n)$ other space for bookkeeping. Therefore, it uses $O(n(r+m))$ bits total. As for correctness, the algorithm never outputs incorrectly, since the Equality test is one-sided in the first pass, and the test in the second pass has zero error. By a union bound, the chance that the algorithm does *not* terminate after the first pass when $\mathrm{ListEQ}(X, Y) = 0$ is at most $n2^{-r}$. When $\mathrm{ListEQ}(X, Y) = 1$, the algorithm fails to terminate only when $X_i \neq Y_i$ for all $i \in I'$. This only happens when at least m Equality tests fail in the first pass, which happens with probability (much less than) $n^m 2^{-rm} = 2^{m \log n - rm}$. Taking $m = r = 2 \log n$ gives a two-pass, $O(n \log n)$ space Las Vegas algorithm for ListEQ.

4 Upper Bounds

In this section, we present new one-sided estimators for two problems motivated by machine learning from the recent work of Clarkson et al. [4]

Minimum Enclosing Ball. In the Minimum Enclosing Ball (MEB) problem, the input is a matrix $A \in \{-M, -M+1, \ldots, M\}^{nd}$, for some $M = \mathrm{poly}(nd)$, whose rows are treated as points in d-dimensional space. The goal is to estimate the radius of the minimum enclosing ball of these points; i.e., to estimate $\min_{y \in \mathbb{R}^d} \max_{1 \leq i \leq n} ||A_i - y|| \ .$

In the streaming version of this problem, we assume that we see the rows of A one at a time (and exactly once), but in an arbitrary order. An algorithm from [4] runs in $\tilde{O}(1/\varepsilon^2)$ space and uses $\tilde{O}(1/\varepsilon)$ passes and returns $1/\varepsilon$ indices

$i_1, \ldots, i_{i/\varepsilon}$ such that with probability at least $1/2$, the ball centered around these indices that encloses all points has radius close to the smallest possible. In other words, the point $y := \sum_{j=1}^{1/\varepsilon} \varepsilon A_{i_j}$ is the center of a ball whose radius $r := \max_{1 \le i \le n} ||A_i - y||$ is an ε-overestimate of the radius of the MEB. It is easy to see that y can be computed with one more pass and $O(d \log M)$ more bits of space. Given y, the radius of the ball centered at y can be computed in an extra pass using $O(\log M)$ additional space by maintaining the maximum distance of a point from y. This radius is thus an ε-overesetimator for MEB. One can reduce the failure probability from $1/2$ by repeating this process independently and in parallel several times and taking the minimum radius found.

Theorem 4.1. *There is an ε-overestimator for Minimum Enclosing Ball that uses $O(d \log(nd) + \text{polylog}(nd/\varepsilon)/\varepsilon^2)$ space and $O(\text{polylog}(nd/\varepsilon)/\varepsilon)$ passes.*

Classification. As with the previous problem, the input is a set of n points $A_1, \ldots, A_n \in \{-M, -M+1, \ldots, M\}^d$ (Points A_i are assumed to have $||A_i|| \le 1$.) Given $x \in \mathbb{R}^d$, define $\sigma_x := \min_i \langle A_i, x \rangle$. In the classification problem, the goal is to output the margin $\sigma := \min_{x:||x|| \le 1} \sigma_x$. Another algorithm from [4] runs in $\tilde{O}(1/\varepsilon^2)$ space and $\tilde{O}(1/\varepsilon^2)$ passes and returns a set of $t = O(1/\varepsilon^2)$ indices $i_1, \ldots, i_t$ such that with constant probability, a certain linear combination y of $\{A_{i_j}\}_{j=1}^t$ gives an additive ε-approximation to the margin. As in the case of MEB, y can be computed in $O(d \log M)$ additional bits of space, from which σ_y can be computed exactly, which is an ε-underestimator for the margin.

Theorem 4.2. *There is an $O(d \log(nd) + \text{polylog}(nd/\varepsilon)/\varepsilon^2)$ space, $O(\text{polylog}(nd/\varepsilon)/\varepsilon^2)$-pass algorithm that computes y such that $\sigma \ge \sigma_y \ge \sigma - \varepsilon$.*

5 Lower Bounds

In the Longest-Increasing-Subsequence problem, the input is a stream of n tokens $\sigma \in [m]^n$, and the goal is to estimate the length of the longest increasing sequence of σ, which we denote $\text{lis}(\sigma)$. Gopalan et al. [11] gave an $O(\sqrt{n/\varepsilon})$ deterministic algorithm for estimating $\text{lis}(\sigma)$; this space complexity was later proven tight by Gál and Gopalan [10] and Ergun and Jowhari [7].

The proof of Gál and Gopalan uses a reduction from the Hidden-Increasing-Subsequence ($\text{HIS}_{\ell,t,k}$) problem. $\text{HIS}_{\ell,t,k}$ is a t-player communication problem where PLR_i is given the ith row of a matrix $M \in [m]^{t\ell}$, with the promise that either (i) all columns are decreasing, or (ii) there exists a column with an increasing subsequence of length k. The players wish to distinguish these cases.

Gál and Gopalan proved a lower bound on the maximum communication complexity of deterministic, one-way protocols for $\text{HIS}_{\ell,t,k}$. We need similar lower bounds for randomized protocols that make no mistakes when there exists a hidden increasing sequence. Let $\text{R}^{\text{MAX},0}(\text{HIS}_{\ell,t,k})$ denote the maximum communication complexity (i.e., the size of the largest message) of the best randomized, one-way protocol for $\text{HIS}_{\ell,t,k}$ that errs only when all columns of M are decreasing. We observe that the deterministic lower bound technique of Gál and Gopalan generalizes to $\text{R}^{\text{MAX},0}(\text{HIS}_{\ell,t,k})$.

Theorem 5.1. $\mathrm{R}^{\mathrm{MAX},0}(\mathrm{HIS}_{\ell,t,k}) \geq \ell((1-k/t)\log(m/(k-1)) - H(k/t)) - \log t$. *In particular, taking* $n := t\ell, k := t/2+1$, *and* $\varepsilon := (k-1)/\ell$, *we have*

$$\mathrm{R}^{\mathrm{MAX},0}(\mathrm{HIS}_{\ell,t,k}) = \Omega(\sqrt{n/\varepsilon}\log(m/\varepsilon n)) = \tilde{\Omega}(\sqrt{n/\varepsilon})\ .$$

Using the reduction from Gopalan et al [11], we get the following corollary.

Corollary 5.1. *An* ε*-overestimator for* Longest-Increasing-Subsequence *requires* $\Omega(\sqrt{n/\varepsilon})$ *space.*

In the rest of this section, we provide a suite of lower bounds for streaming statistics. Unless otherwise specified, the underlying vector $z \in [m]^n$ is initialized to zero, and tokens (i, v) represent updates $z \leftarrow z_i + v$. Our lower bounds cover the following problems.

- ℓ_p**-norm:** estimate $||z||_p := (\sum_{i=1}^n |z_i|^p)^{1/p}$.
- ℓ_p **heavy hitters:** For "heavy hitter thresholds" $\hat{\phi} < \phi$, return all i such that $|z_i|^p \geq \phi F_p(z)$ and no i such that $|z_i|^p \leq \hat{\phi} F_p(z)$.
- **empirical entropy:** estimate $H(z) = \sum_i (|z_i|/F_1(z))\log(F_1(z)/|z_i|)$. (Recall that $F_1(z) := \sum_i |z_i|$ is the ℓ_1-norm of the stream.)

All of these lower bounds come from reductions from NEQ or Gap-EQ. Alice and Bob convert strings x, y into streams σ_A, σ_B. The communication protocol works by simulating a streaming algorithm on $\sigma := \sigma_A \circ \sigma_B$ and estimating the resulting statistic. Because these lower bounds are similar and space is limited, we include only a few proofs and defer others to the full version of the paper.

Theorem 5.2. *For all* p, $\mathrm{S}_{\varepsilon\text{-}over}(\ell_p\text{-}norm) = \Omega(n)$ *in the general turnstile model.*

Proof. This is a simple reduction from NEQ. We omit the details.

Theorem 5.3. *For all* $p \neq 1$, *there exists* $\varepsilon > 0$ *such that* $\mathrm{S}_{\varepsilon\text{-}under}(\ell_p\text{-}norm) = \mathrm{S}_{\varepsilon\text{-}over}(\ell_p\text{-}norm) = \Omega(n)$ *in the insertion-only model.*

Proof. We require different reductions for ε-overestimators and ε-underestimators and for when $p < 1$ and $p > 1$; however, in all cases, we reduce from Gap-EQ by embedding either (x, y) or $(x, -y)$ into the streaming problem. In all cases, choosing ε appropriately ensures that the relevant one-sided estimator gives a protocol for Gap-EQ with one-sided error. All four reductions are similar; we include a proof for the case where $p < 1$ and we want a lower bound for ε-overestimators and defer the other proofs to the full version.

Suppose that $p < 1$, and let $\mathcal{A}_O$ be an ε-overestimator for the ℓ_p-norm, where $\varepsilon := \min\{1/3, (1-2^{p-1})/(2^{p+1}p)$. Given x, Alice creates a stream $\sigma_A = (a_1, \ldots, a_n)$, where $a_i := (2i - x_i, 1)$. Bob converts y into a stream $\sigma_B := (b_1, \ldots, b_n)$, where $b_i := (2i - y_i, 1)$. Note that

$$||z||_p = \begin{cases} 2n^{1/p} & \text{if } x = y\ , \\ 2n^{1/p}\left(\frac{1}{2} + 2^{-p}\right)^{1/p} & \text{if } \delta(x,y) = n/2\ . \end{cases}$$

When $p < 1$, the ℓ_p-norm given by the $x = y$ case is less than the $\delta(x, y) = n/2$ case. Therefore, Alice and Bob can solve GAP-EQ by simulating $\mathcal{A}_O$ and returning "$\delta(x, y) = n/2$" if $\mathcal{A}_O$ returns an estimate at least $2n^{1/p}(1/2+2^{-p})^{1/p}$. Since $\mathcal{A}_O$ always provides an overestimate, the protocol always computes the $\delta(x, y) = n/2$ cases. Further, note that

$$(1+\varepsilon)^p < (1+2\varepsilon p) \leq 1 + 2p\left((1-2^{p-1})/p2^{p+1}\right) \leq (1/2+2^{-p}) \ ,$$

where the first inequality uses $(1+x)^r < 1+2xr$, which holds for $r > 0$ and $0 \leq x \leq 1/2$. Therefore, when $x = y$, $\mathcal{A}_O$ (with high probability) returns an estimate at most $2n^{1/p}(1+\varepsilon) < 2n^{1/p}(1/2+2^{-p})^{1/p}$, hence the protocol computes "$x = y$" correctly with high probability.

Note that this reduction fails for the case $p = 1$ because the gap in ℓ_p-norm in the YES and NO instances disappears. The ℓ_p-norm in this case corresponds to counting the net number of items in the stream. This can easily be exactly computed in $O(\log n)$ space.

Finally, we consider one-sided estimates for ℓ_p-heavy hitters. The notion of one-sidedness is slightly different here, since the algorithm is to output a set of items instead of an estimation. Here, we define the over- and under-estimation to refer to the set of items that are reported.

Definition 5.1. *A two-sided estimator for the $(\hat{\phi}, \phi, \ell_p)$ heavy hitters problem is a randomized algorithm that with probability* $2/3$

1. *returns all i such that $|z_i|^p \geq \phi F_p(z)$.*
2. *returns no i such that $|z_i|^p \leq \hat{\phi} F_p(z)$.*

An overestimator is an algorithm that achieves condition (1) with probability 1*. An underestimator fufills condition (2) with probability* 1*.*

Theorem 5.4. *The following bounds hold for $(\hat{\phi}, \phi, \ell_p)$-heavy hitters:*

- *For all $0 < \phi < 1, \hat{\phi} = \phi/2$, and $p \geq 0$, $\Omega(n)$-space is required in the general turnstile model for both over- and underestimators.*
- *For all $0 < \phi < 1$ and $p \neq 1$, there exists $\hat{\phi}$ such that $\Omega(n)$ space is required in the insertion-only model for both over- and underestimators.*
- *$\Theta(\log n/\phi)$ space is required in the insertion-only model for all $(\phi/2, \phi, 1)$ heavy-hitters.*

Theorem 5.5. *For $\varepsilon = O(1/\log n)$, $\Omega(n)$ space is necessary to ε-overestimate or ε-underestimate the empirical entropy $H(z)$ in the insertion-only model.*

Open Questions: Our work leaves open several natural questions.

1. Can one characterize the functions f for which $\mathrm{S}_{\varepsilon\text{-under}}(f) = \mathrm{S}_{1\pm\varepsilon}(f)$ or $\mathrm{S}_{\varepsilon\text{-over}}(f) = \mathrm{S}_{1\pm\varepsilon}(f)$? A complete characterization may be hard, as it could be used to obtain bounds on $\mathrm{S}_{1\pm\varepsilon}$(LONGEST-INCREASING-SUBSEQUENCE) and $\mathrm{S}_{1\pm\varepsilon}(EMD)$, two challenging questions in the data stream literature. Even a partial characterization would be interesting.
2. What results hold for estimators $\hat{f}(\sigma)$ for which $\hat{f}(\sigma) \geq f(\sigma)$ always, and with probability at least $2/3$, $\hat{f}(\sigma) \leq f(\sigma)(1+\varepsilon) + \beta$?

Acknowledgements. We thank Kevin Matulef for several helpful discussions.

References

1. Andoni, A., Ba, K.D., Indyk, P., Woodruff, D.: Efficient sketches for earth-mover distance, with applications. In: Proc. 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 324–330 (2009)
2. Andoni, A., Indyk, P., Krauthgamer, R.: Overcoming the ℓ_1 non-embeddability barrier: Algorithms for product metrics. In: SODA (2009)
3. Buhrman, H., Cleve, R., Wigderson, A.: Quantum vs. classical communication and computation. In: Proc. 30th Annual ACM Symposium on the Theory of Computing, pp. 63–68 (1998)
4. Clarkson, K.L., Hazan, E., Woodruff, D.P.: Sublinear optimization for machine learning. In: Proc. 51st Annual IEEE Symposium on Foundations of Computer Science, pp. 449–457 (2010)
5. Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. J. Alg. 55(1), 58–75 (2005); preliminary version in Proc. 6th Latin American Theoretical Informatics Symposium, pp. 29–38 (2004)
6. Cormode, G., Muthukrishnan, S.: Space efficient mining of multigraph streams. In: Proc. 24th ACM Symposium on Principles of Database Systems, pp. 271–282 (2005)
7. Ergün, F., Jowhari, H.: On distance to monotonicity and longest increasing subsequence of a data stream. In: Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 730–736 (2008)
8. Frank, O.: Estimation of the number of vertices of different degrees in a graph. Journal of Statistical Planning and Inference 4(1), 45–50 (1980)
9. Frankl, P., Rödl, V.: Forbidden intersections. Trans. Amer. Math. Soc. 300(1), 259–286 (1987)
10. Gál, A., Gopalan, P.: Lower bounds on streaming algorithms for approximating the length of the longest increasing subsequence. In: Proc. 48th Annual IEEE Symposium on Foundations of Computer Science, pp. 294–304 (2007)
11. Gopalan, P., Jayram, T.S., Krauthgamer, R., Kumar, R.: Estimating the sortedness of a data stream. In: Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 318–327 (2007)
12. Indyk, P.: Algorithms for dynamic geometric problems over data streams. In: Proc. 36th Annual ACM Symposium on the Theory of Computing, pp. 373–380 (2004)
13. Jayram, T., Woodruff, D.P.: The data stream space complexity of cascaded norms. In: FOCS, pp. 765–774 (2009)
14. Kushilevitz, E., Nisan, N.: Communication Complexity. Cambridge University Press, Cambridge (1997)
15. Mehlhorn, K., Schmidt, E.M.: Las vegas is better than determinism in vlsi and distributed computing (extended abstract). In: Proc. 14th Annual ACM Symposium on the Theory of Computing, pp. 330–337 (1982)
16. Monemizadeh, M., Woodruff, D.P.: 1-pass relative-error l_p sampling with applications. In: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1143–1160 (2010)
17. Muthukrishnan, S.: Data Streams: Algorithms and Applications. Foundations and Trends in Theoretical Computer Science 1(2), 117–236 (2005)