Stochastic Streams: Sample Complexity vs. Space Complexity

Michael Crouch[1], Andrew McGregor[2], Gregory Valiant[3], and David P. Woodruff[4]

1 **Bell Labs Ireland**
2 **University of Massachusetts Amherst**
3 **Stanford University**
4 **IBM Research Almaden**

Abstract

We address the trade-off between the computational resources needed to process a large data set and the number of samples available from the data set. Specifically, we consider the following abstraction: we receive a potentially infinite stream of IID samples from some unknown distribution D, and are tasked with computing some function $f(D)$. If the stream is observed for time t, how much memory, s, is required to estimate $f(D)$? We refer to t as the sample complexity and s as the space complexity. The main focus of this paper is investigating the trade-offs between the space and sample complexity. We study these trade-offs for several canonical problems studied in the data stream model: estimating the collision probability, i.e., the second moment of a distribution, deciding if a graph is connected, and approximating the dimension of an unknown subspace. Our results are based on techniques for simulating different classical sampling procedures in this model, emulating random walks given a sequence of IID samples, as well as leveraging a characterization between communication bounded protocols and statistical query algorithms.

1998 ACM Subject Classification F.2.3 Tradeoffs between Complexity Measures

Keywords and phrases data streams, sample complexity, frequency moments, graph connectivity, subspace approximation

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.258

1 Introduction

Big data systems must process data at multiple levels of a hierarchy, starting at local per-process or per-container monitoring and extending up to aggregate statistics for entire databases or data centers. When designing monitoring or analytics for these systems, some of the first decisions which must be made are which levels of the hierarchy should hold the analytics, and consequently what computational resources are available for processing data without introducing significant overhead. In this paper, we initiate the theoretical investigation of one of the fundamental trade-offs involved in architecting these systems: the amount of memory needed to process incoming data and the total amount of data the system must collect.

In algorithmic terms, we consider the following abstraction: we receive a stream of IID samples from some unknown distribution D, and are tasked with estimating some function $f(D)$ of the distribution. Two natural questions about this task have been studied extensively:

1. The statistics question is how to bound the *sample complexity*: how many samples are required to estimate $f(D)$ to some prescribed accuracy with high probability?

2. The data stream question is how to bound the *space complexity*: how much memory is required to compute or approximate the estimator for $f(D)$?

In real systems, of course, both questions are important. Requiring more samples adds processing overhead to the system, and increases the time necessary for the system to detect and react to changes in the underlying distribution. Requiring additional memory increases the overhead on the system, and may make some analytics impractical, or require them to be relocated to separate machines or systems.

Despite the clear importance of this trade-off, our work is the first to our knowledge which explicitly examines the trade-off between the number of samples which must be taken from a stream and the amount of memory necessary to process these samples. We begin this investigation with a study of three canonical problems from the data streaming literature: estimating the collision probability, also known as the second frequency moment [2], undirected connectivity [13, 14, 35], and rank estimation [10, 25, 6]. For all three problems, we find trade-offs between the sample and space requirements.

Sufficient Statistics and Data Streams. The goal of space-efficiency in statistical estimation is not new. In the study of *sufficient statistics* [15] the goal is to prove that it suffices to maintain a small number of statistics about the input when estimating certain parameters of the source distribution D. For example, to estimate μ if $D \sim N(\mu, 1)$, it is sufficient to maintain the sum and count of the samples; other information can be discarded. However, for non-parametric problems sufficient statistics typically do not exist. Our work could be seen as "approximate sufficient statistics" – statistics about the stream of samples that can be computed online that will suffice to estimate the relevant properties of the input with high probability.

In contrast to the majority of data stream research, in our setting we do not need to consider adversarially-ordered streams since the assumption is that the input stream is generated by a stochastic process. There is a growing body of work on randomly-ordered streams [20, 33, 29, 8, 27, 17, 7]. Some work has also explicitly considered streams of IID samples [9, 41, 19, 30]. There has also been work on hypothesis testing given limited space [23, 21].

Subsampling vs. Supersampling. Other related work includes a paper by an overlapping set of authors [26] (see also [37]) that considered the problem of processing data streams whose arrival rate was so high that it was not possible to observe every element of the stream. Consequently, it was presumed that the stream was first *subsampled* and then properties of the original stream had to be deduced from the samples. In contrast, in this work we essentially consider oversampling, or *supersampling* the data set. The motivation is two-fold. First, in many applications there is an abundance of redundant data, e.g., from sensors that continually monitor a static environment, and it makes sense to find a way to capitalize on this data. Second, it may be preferable from a computational point of view, to run a fast light-weight algorithm on a lot of data rather than a computationally expensive algorithm on a small amount of data.

A number of early works consider learning in online settings, and we only mention a few here. There is a line of work on hypothesis testing and statistical estimation with finite memory, see, e.g., [11, 24], but the lower bounds in such models come from finite precision constraints, and do not seem directly related to the model or problems that we study. Other work [31] considers lower bounds for algorithms given data in a specific form, i.e., in so-called oracle models, and the limitations of such assumptions on the model have been exploited to design faster algorithms in some cases [32].

More recently, [38, 39] consider the question of learning with memory or communication

constraints in the setting in which the algorithm has access to a stream of randomly drawn labeled examples. Their algorithmic "memory bounded" model corresponds to our model. We note that these works study a different set of problems than the ones from the data stream literature that we focus on, reinforcing the naturalness of the model. The lower bounds in these works use a more general communication model than our streaming machines: given a communication bound of b bits per example, an algorithm must compress the ith example according to an arbitrary function that may depend on the compressed versions of the first $i-1$ examples; the algorithm may then compute an arbitrary function of the compressed examples. The authors show tight connections between communication bounded protocols, and statistical query algorithms, and we leverage their characterization to show a communication lower bound, and ultimately space lower bound in Section 4.

Our Results. We study the trade-off between sample and space complexity for canonical problems in the data stream literature such as estimating the collision probability or second frequency moment [2], undirected connectivity [13, 14, 35], and rank estimation [10, 25, 6]; see also the references therein. We obtain the following tradeoffs:

1. *Collision Probability.* Suppose $D = D_p$ is a distribution $(p_1, \ldots, p_n)$ over $[n]$. Then, to estimate $F_2(D) = \sum_i p_i^2$ up to a factor $(1+\epsilon)$ with probability $1-\delta$, for any[1] $t \geq t^* = \tilde{\Omega}_{\epsilon,\delta}(n^{1/2})$ it is sufficient for

$$s \cdot t = \tilde{O}_{\epsilon,\delta}(n)$$

   Moreover, we show a lower bound that $s \cdot t^{1.5} = \tilde{\Omega}(n^{5/4})$ space is necessary to return a constant factor approximation with constant probability, which is tight when s and t are $\tilde{\Theta}_{\epsilon,\delta}(\sqrt{n})$. We also point out an error in existing work, which if fixable, would result in the tight $s \cdot t = \tilde{\Omega}(n)$ tradeoff. We present these results in Section 2.
2. *Graph Connectivity.* Suppose $D = D_G$ is the uniform distribution of the set of edges of an undirected, unweighted graph G. We show in Section 3 that for $t \geq t^* = \Omega(|E| \log |E|)$, it suffices for

$$s^2 \cdot t = \tilde{O}(|E| \cdot |V|^2)$$

3. *Subspace Approximation.* In Section 4 we consider the problem of determining whether a set of samples from $\{0,1\}^n$ is being drawn from the entire space or from some rank $n/2$ subspace. We first note an $s = O(\log n)$-bit space streaming algorithm which achieves sample complexity $t = O(2^n)$. More interestingly, we then show that this is near-optimal by showing that any streaming algorithm using $s \leq n/8$ space requires

$$2^s \cdot t = \Omega(2^{n/8}) \ .$$

   So even if one allows say, $s = n/16$ bits of space, one still needs $2^{\Omega(n)}$ examples.

We believe one of the main contributions of our work is the new framework for analyzing tradeoffs between the sample and space complexity of streaming algorithms. We have chosen this set of problems because it is representative of well-studied sub-areas in the data stream literature such as estimating statistical quantities such as frequency moments, graph problems in the semi-streaming model, and numerical linear algebra problems.

[1] $\tilde{O}_{\epsilon,\delta}(f(n))$ and $\tilde{\Omega}_{\epsilon,\delta}$ omit $\mathrm{poly}(\log(n/\delta), 1/\epsilon)$ terms.

2 Collision Probability

The collision probability is a fundamental quantity quantifying how far a distribution is from the uniform distribution, the latter having the smallest possible collision probability. Estimating the empirical collision probability is well-studied for worst case data streams of a given length, dating back to the seminal work of Alon, Matias, and Szegedy [2]. It has applications to estimating self-join sizes in databases, and is also known as the *repeat rate* or *Gini's Index of homogeneity*, which is used to compute the *surprise index* [16].

Here we consider a stream of independent samples $\langle a_1, a_2, a_3, \ldots\rangle$ from an unknown discrete distribution p over $[n]$. Let $p_j = \mathbb{P}[a_i = j]$ and let $F_2 = \sum_j p_j^2$.

2.1 The Upper Bound

▶ **Theorem 1.** *For $t = \tilde{\Omega}_{\epsilon,\delta}(n^{1/2})$, estimating F_2 up to a factor $(1 \pm \epsilon)$ given t samples with probability at least $1 - \delta$ is possible in $\tilde{O}_{\epsilon,\delta}(1 + n/t)$ bits of space.*

Proof. We can assume δ is a constant, since the algorithm generalizes to smaller δ simply by increasing the sample complexity by a factor of $\log(1/\delta)$, running our algorithm $\log(1/\delta)$ times in sequence on each of $\log(1/\delta)$ independent groups of samples, and taking the median. Note $t = \Omega(n^{1/2})$ was shown necessary by Bar-Yossef [4]; with fewer samples estimating F_2 is information-theoretically impossible.

Our algorithm partitions the input stream into t/w groups of w consecutive samples. We now analyze one specific group of w samples. Suppose the samples are $a_1, \ldots, a_w$. For each pair $i \neq j \in \{1, 2, \ldots, w\}$, let $X_{i,j}$ be an indicator random variable which is 1 iff $a_i = a_j$. Let $X = \frac{1}{\binom{w}{2}} \sum_{i \neq j} X_{i,j}$. Then $\mathbf{E}[X]$ is equal to $\mathbf{E}[X_{i,j}]$ for an arbitrary $i \neq j$, and the latter is precisely $F_2 = \sum_i p_i^2$. We also have, assuming $i \neq j$ and $i' \neq j'$ in the following:

$$\begin{aligned}
\binom{w}{2}^2 \mathbf{Var}[X] &= \sum_{(i,j)=(i',j')} \mathbf{E}[X_{i,j}X_{i',j'}] - \mathbf{E}[X_{i,j}]\mathbf{E}[X_{i',j'}] \\
&\quad + \sum_{|\{i,j,i',j'\}|=3} \mathbf{E}[X_{i,j}X_{i',j'}] - \mathbf{E}[X_{i,j}]\mathbf{E}[X_{i',j'}] \\
&\quad + \sum_{|\{i,j,i',j'\}|=4} \mathbf{E}[X_{i,j}X_{i',j'}] - \mathbf{E}[X_{i,j}]\mathbf{E}[X_{i',j'}] \\
&= \sum_{(i,j)} \mathbf{E}[X_{i,j}^2] - \mathbf{E}^2[X_{i,j}] + \sum_{|\{i,j,i',j'\}|=3} \mathbf{E}[X_{i,j}X_{i',j'}] - \mathbf{E}[X_{i,j}]\mathbf{E}[X_{i',j'}] \\
&\quad + \sum_{|\{i,j,i',j'\}|=4} \mathbf{E}[X_{i,j}]\mathbf{E}[X_{i',j'}] - \mathbf{E}[X_{i,j}]\mathbf{E}[X_{i',j'}] \\
&\leq \sum_{(i,j)} \mathbf{E}[X_{i,j}^2] + \sum_{|\{i,j,i',j'\}|=3} \mathbf{E}[X_{i,j}X_{i',j'}] - \mathbf{E}[X_{i,j}]\mathbf{E}[X_{i',j'}],
\end{aligned}$$

where the first equality follows by expanding the variance into covariances, the second equality uses independence of disjoint indices, and the inequality cancels the terms for which

$|\{i,j,i',j'\}| = 4$ and drops non-positive terms. Therefore,

$$\begin{aligned}
\mathbf{Var}[X] &\leq \frac{1}{\binom{w}{2}^2}\left(\sum_{(i,j)}\mathbf{E}[X_{i,j}^2] + \sum_{|\{i,j,i',j'\}|=3}\mathbf{E}[X_{i,j}X_{i',j'}]\right)\\
&= \frac{\mathbf{E}[X]}{\binom{w}{2}} + \Theta\left(\frac{1}{w^4}\right)\sum_{\text{distinct } i,j,k}\mathbf{E}[X_{i,j}X_{j,k}]\\
&= \frac{F_2}{\binom{w}{2}} + \Theta\left(\frac{1}{w^4}\right)\sum_{\text{distinct } i,j,k}\Pr[X_{i,j}=1 \mid X_{j,k}=1]\cdot\Pr[X_{j,k}=1]\\
&= \frac{F_2}{\binom{w}{2}} + \Theta\left(\frac{1}{w^4}\right)\sum_{\text{distinct } i,j,k}\sum_{\ell}\Pr[X_{i,j}=1 \mid a_j=\ell]\cdot\Pr[a_j=\ell \mid X_{j,k}=1]\cdot F_2\\
&= \frac{F_2}{\binom{w}{2}} + \Theta\left(\frac{1}{w^4}\right)\sum_{\text{distinct } i,j,k}\sum_{\ell} p_\ell\cdot\frac{p_\ell^2}{F_2}\cdot F_2\\
&= \frac{F_2}{\binom{w}{2}} + \Theta\left(\frac{F_3}{w}\right),
\end{aligned}$$

where we have used the law of total probability and the definitions, and here $F_3 = \sum_j p_j^3$.

In the stream, we will use $O(w\log n)$ bits of space to compute X as above. Then, since we have partitioned the samples into $q = t/w$ groups (which we can assume is an integer w.l.o.g.), we will compute an independent estimate X for each group, and take their average, obtaining a random variable Y. Thus Y can be computed in $O(w\log n)$ bits of space. Then $\mathbf{E}[Y] = \mathbf{E}[X] = F_2$ and $\mathbf{Var}[Y] = \mathbf{Var}[X]/q$. By Chebyshev's inequality,

$$\Pr[|Y - F_2| \geq \epsilon F_2] \leq \frac{\mathbf{Var}[X]}{q\epsilon^2 F_2^2} \leq \frac{1}{\binom{w}{2}F_2 q\epsilon^2} + O\left(\frac{F_3}{wq\epsilon^2 F_2}\right) = O\left(\frac{n}{w^2 q\epsilon^2}\right) + O\left(\frac{\sqrt{n}}{wq\epsilon^2}\right),$$

where the final inequality uses that $F_2 \geq 1/n$ in the first expression, while the second expression uses Hölder's inequality to show that $F_3 \leq \sqrt{n}F_2$.

Plugging in $q = t/w$, this probability is $O(n/(tw\epsilon^2) + \sqrt{n}/(t\epsilon^2))$. Thus, for $t = \Omega(n^{1/2}\epsilon^{-2})$ samples, we can set $w = O(n/t)$ and have this probability be smaller than an arbirarily small constant. Note that as a sanity check, we need $t = \Omega(n^{1/2})$ to ensure $w \leq t$, as required by the definition of these variables. This completes the proof. ◀

2.2 The Lower Bound

Our lower bound relies on a result by Andoni et al. [3] for the random order data stream model, which is different than the model we consider in this paper. We note that an improvement to the work of Andoni et al. [3] was claimed in [18]; however, after communication with the authors, the proof given in [18] seems to have a bug and is currently not known to be fixable. If the result in [18] is fixable, then we obtain an optimal tradeoff of $s \cdot t = \tilde{\Omega}(n)$; otherwise we obtain the slightly weaker tradeoff presented here.

The work of [3] concerns distinguishing between the following two cases with constant probability, and shows that $\Omega(t/r^{2.5})$ space is required:

1. (Case 1:) A sequence of t samples from a distribution p^{no} that is uniform on some subset $S \subseteq [t]$ of size $\Theta(t)$.
2. (Case 2:) A sequence of t samples from a distribution p^{yes} such that $p_i^{\text{yes}} = r/t$ for some $i \in [t]$ and uniform on some subset $T \subseteq [t] \setminus \{i\}$.

By combining this result with a hashing technique we establish the following result.

▶ **Theorem 2.** *Any constant factor approximation of $F_2(D)$ with constant probability given a sequence of t IID samples on $[n]$ requires $\Omega(n^{5/4}/((\log^{2.5} n) \cdot t^{1.5}))$ bits of space.*

Proof. Let $h : [t] \to [n]$ be a fully-random hash function and consider the problem of distinguishing p^{no} and p^{yes} where we set $r = c \log n \cdot t \cdot n^{-1/2}$ for some constant $c > 0$. By applying h on each distribution (i.e., applying h to each observed sample) we generate two new distributions q^{no} and q^{yes} over $[n]$ where:

$$q_i^{\text{no}} = \sum_{j:h(j)=i} p_j^{\text{no}} \quad \text{and} \quad q_i^{\text{yes}} = \sum_{j:h(j)=i} p_j^{\text{yes}}$$

Note that with high probability $\max_i q_i^{\text{no}} = O(\log n \cdot 1/n)$ and hence $F_2(q^{\text{no}}) \leq n \cdot O((\log n \cdot 1/n)^2) = O(\log^2 n \cdot n^{-1/2})$. However, $\max_i q_i^{\text{yes}} \geq r/t$ and so $F_2(q^{\text{yes}}) \geq r^2/t^2 = c^2 \cdot \log^2 n \cdot n^{1/2}$. Hence, for a sufficiently large value of the constant $c > 0$ we can ensure that any constant approximation of F_2 distinguishes between q^{yes} and q^{no} and hence, also distinguishes between p^{yes} and p^{no}. However, by the result of Andoni et al. [3] we know that this requires $\Omega(t/r^{2.5}) = \Omega(n^{2.5(1/2)}/((\log^{2.5} n) \cdot t^{1.5}))$ bits of space. ◀

3 Connectivity

In this section, we consider a graph model where our input stream is a sequence of t random samples drawn (with replacement) from the graph's entire edge set E. This model is appropriate for random processes where the "graph" involved is defined only implicitly, and is not available for querying. For example, on a large social network, we can imagine a graph where nodes represent users and weighted edges represent user frequencies of interaction. We may not have enough space or processing power to analyze the history of interactions between users directly, or to provide random access to this history; according to [40] an estimated 50 billion text messages per day were sent in 2014. However, the stream of ongoing interactions approximates random samples from the weighted edge distribution.

We begin the study of this model by first examining unweighted graphs, and by examining the canonical problem of determining connectivity for such graphs. Our algorithm uses the sampled edges to simulate classical random walks on the graph. In Section 3.1, we discuss how to simulate and tightly analyze random walks in our model. The main technical difficulty is ensuring independence when simulating multiple random walks in parallel. Then, in Section 3.2, we adapt a connectivity algorithm of Feige [14] to achieve the required space/sample trade-off. Since our algorithm provides a general technique for space/sample trade-offs in simulating random walks, we believe it will be a useful starting point for examining additional problems in this model.

For notational convenience, denote the graph by $G = (V, E)$, the number of nodes by $n = |V|$, and the number of edges by $m = |E|$.

3.1 Technique: Emulating Classical Random Walks

Consider the following basic algorithm: given a node v, we sample edges until we receive an edge $\{v, u\}$ for some u. At this point, we move to node u, and repeat. We refer to this method as a *sampling walk*. Note that the expected time to leave v is $m/d(v)$ samples[2] where $d(v)$ is the degree of a node v, and so a single step of a classical random walk may require

[2] This is because the number of samples is geometrically distributed with parameter $d(v)/m$.

$\Omega(m)$ samples if v has low degree.

An Inefficient Connectivity Algorithm. This basic algorithm already leads to a $O(\log n)$ space algorithm which uses $O(m^2n^2)$ samples in expectation. This follows by starting a sampling walk at node 1 and emulating a classical walk until it traverses nodes $2, 3, \ldots, n$ in order. The expected length of this walk is $O(mn^2)$ because the cover time of G, i.e., the expected length of walk until it visits all nodes (see e.g., [28]), is $O(mn)$ and there are $n-1$ segments in the traversal. Hence, emulating the random walk takes $O(m^2n^2)$ samples in expectation. The space use is $O(\log n)$ bits because the algorithm just needs to remember the current node and the furthest node that has been reached in the sequence. In what follows, we will improve upon the number of samples required and generalize to algorithms that use more space.

The Loopy Graph and an Improved Analysis. The first improvement comes via a better analysis. At a node v with $d(v)$ neighbors, there are $d(v)$ possible samples which would result in a move, and $m - d(v)$ samples which would not. We can thus view our sampling-based walk on G as a classical random walk on a new graph H formed by adding $m - d(v)$ self-loops to each vertex v in G. We call H the "loopy graph".

This view of the sampling walk illuminates its properties. Specifically, H's cover time is $O(mn^2)$ since there are mn edges and n nodes. Hence, the above "inefficient" algorithm actually only requires $(n-1) \times \text{cover-time}(H) = O(mn^3)$ samples. We will also subsequently use the fact that since H is m-regular, its stable distribution is uniform across all nodes.

3.1.1 Multiple Independent Random Walks

Random walks experience dramatic speedups in cover time, hitting time, etc., when they are split into multiple shorter walks; [12] provides a recent survey and results. These speedups naturally require the walks to be independent. In this section, we consider performing $p \leq n$ random walks in parallel, with the starting point of each walk chosen independently and uniformly from the nodes (and thus according to the stationary distribution on H). Running these p walks will require $O(p \log n)$ space. The main theorem of this section establishes that it is possible to efficiently perform p independent, parallel walks in H.

▶ **Theorem 3.** *Given $p \leq n$ parallel random walks in H, each starting at an independently-chosen uniformly random node, we can simulate one independent step of each walk using $O(\log n / \log\log n)$ total samples.*

Issue 1: Multiple Walks can use a Sampled Edge. The first issue we encounter is that a single sample may be a valid move for multiple walks. If we allow multiple walks to use the same sample, we introduce obvious dependence; if we only allow one of our walks to use the sample, we are "slowing down" walks that have collisions, and again introducing dependence.

When multiple walks are at the same node, we will handle them independently in the following way. We partition the p walks into $B_1 \cup B_2 \cup \ldots \cup B_r$ where each B_i contains at most one walk at each node. We process each batch in turn and hence the total number of samples required equals the number of samples required for a batch multiplied by the number of batches. The next lemma establishes that it suffices to consider $r = O(\log n / \log\log n)$ batches.

▶ **Lemma 4.** *With high probability, no node ever contains more than $O(\log n / \log\log n)$ walks.*

Proof. Consider a fixed node at a fixed time. Let Z be the number of walks in this node. Note that $Z \sim Bin(p, 1/n)$ since each walk is independent and is equally likely to be at any node. Hence $\mathbb{E}[Z] = p/n \leq 1$. By an application of the Chernoff bound, for some large constant c we have $\mathbb{P}[Z \geq c \log n / \log\log n] \leq n^{-10}$. The lemma follows by taking the union bound over the n nodes and poly n time-steps. ◀

Henceforth, we assume that at most one walk is at each node, i.e., we analyze how many samples are required to process a single batch. The remaining case where a sampled edge may be valid for multiple walks is if there are walks at both endpoints. To solve this problem, we randomly orient each sampled edge so that it is valid for only one walk. This increases the expected number of samples required by a factor of 2.

Issue 2: Negative Correlation. We have reduced the problem to the following situation: we have p distinct nodes $u_1, \ldots, u_p$ and can sample arcs uv uniformly from the set $E^+ = \{uv : \{u, v\} \in E_G\}$, i.e., the set of arcs formed by bidirecting each edge in E_G. Note that $|E^+| = 2|E_G|$. The goal is to generate a set of arcs $\{u_1v_1, \ldots, u_pv_p\}$ such that each arc is chosen independently and for each i,

$$v_i = \begin{cases} v \in_R \Gamma(u_i) & \text{with probability } d_G(v_i)/|E^+| \\ u_i & \text{with probability } 1 - d_G(v_i)/|E^+| \end{cases} \tag{1}$$

where $\Gamma(u_i) = \{v : \{u, v\} \in E\}$ is the neighborhood of u_i in G, and where $v \in_R \Gamma(u_i)$ denotes an edge drawn randomly from the uniform distribution over the set $\Gamma(u_i)$.

Consider the following procedure: draw a single sample $uv \in_R E^+$ and, for each i, set $v_i = v$ if $u = u_i$, or $v_i = u_i$ otherwise. This procedure picks each v_i according to the desired distribution:

$$\mathbb{P}[v_i = u_i] = 1 - d_G(v_i)/|E^+|$$

and conditioned on $\{v_i \neq u_i\}$, v_i is uniformly chosen from $\Gamma(u_i)$. Unfortunately, the procedure obviously does not satisfy the independence requirement because the events $\{u_i = v_i\}$ and $\{u_j = v_j\}$ are negatively correlated. However, the following theorem establishes that, with only $O(1)$ samples from E^+ in expectation, it is possible to sample independently according to the desired distribution.

▶ **Theorem 5** (Efficient Parallel Sampling). *There exists an algorithm that returns samples* $(v_1, \ldots, v_p)$ *drawn from the desired distribution* (1) *while using at most* $2e - 1$ *samples from* E^+ *in expectation.*

Proof. Our algorithm operates in rounds; each round uses at most 2 samples from E^+. At the beginning of a round, suppose we have already assigned values to $v_1, \ldots, v_i$ for $i \geq 0$. Then the round proceeds as follows:

1. Sample $uv \in E^+$:
   - **a.** If $u \notin \{u_{i+1}, \ldots, u_p\}$ then set $v_{i+1} = u_{i+1}, \ldots, v_p = u_p$
   - **b.** If $u = u_j$ for some $j \in \{i + 1, \ldots, p\}$ then sample an additional arc $wx \in E^+$
     - **i.** If $w \in \{u_{i+1}, \ldots, u_{j-1}\}$ then set $v_{i+1} = u_{i+1}, \ldots, v_{j-1} = u_{j-1}, v_j = u_j$
     - **ii.** If $w \notin \{u_{i+1}, \ldots, u_{j-1}\}$ then set $v_{i+1} = u_{i+1}, \ldots, v_{j-1} = u_{j-1}, v_j = v$

and we repeat the process until all $v_1, \ldots, v_p$ have been assigned.

To analyze the algorithm we define $T_j = \{u_jv : \{u_j, v\} \in E_G\}$ to be the set of $d_G(u_j)$ arcs leaving u_j and note that because $u_1, \ldots, u_p$ are distinct, $T_1, \ldots, T_p$ are disjoint. Also

define A_j to be the event that $\{v_j \neq u_j\}$. Then, in any round in which v_j hasn't yet been assigned:

$$\mathbb{P}\left[v_j \text{ is assigned and } A_j | v_j \text{ is assigned}\right] = \frac{|T_j|}{|E^+| - \sum_{k=i+1}^{j-1} |T_k|} \cdot \frac{|E^+| - \sum_{k=i+1}^{j-1} |T_k|}{|E^+|} = \frac{|T_j|}{|E^+|} \quad (2)$$

and hence v_j is chosen according the desired distribution.

We next show that each v_j is chosen independently. First observe that, conditioned on A_j, v_j is independent of $(v_1, \ldots, v_{j-1}, v_{j+1}, \ldots, v_p)$. Hence, it suffices to show that all A_j are independent. Note that the RHS of (2) does not depend on decisions made in previous rounds. Hence, we may deduce that A_j is independent of the outcome of rounds before v_j is assigned. Hence, for any $1 \leq i_1 < i_2 < \ldots < i_r \leq p$,

$$\mathbb{P}\left[A_{i_1} \cap \ldots \cap A_{i_r}\right] = \mathbb{P}\left[A_{i_1}\right] \mathbb{P}\left[A_{i_2} | A_{i_1}\right] \ldots \mathbb{P}\left[A_{i_r} | A_{i_1} \cap \ldots \cap A_{i_{r-1}}\right] = \mathbb{P}\left[A_{i_1}\right] \mathbb{P}\left[A_{i_2}\right] \ldots \mathbb{P}\left[A_{i_r}\right] .$$

The worst case for the expected number of samples is achieved when $p = |E^+|$ and each set is of size 1. For the algorithm not to terminate in a given round, we need $u \in \{u_{i+1}, \ldots, u_p\}$ and hence the index of the sampled u needs to strictly increase over previous rounds. The probability of this happening for r rounds is $\binom{m}{r}/m^r$ and the expected number of rounds which do not terminate is $\sum_{r=1}^{m} \binom{m}{r}/m^r \leq e - 1$. Because each non-terminating round involves two samples, the expected total number of samples is thus at most $2e - 1$. ◀

3.2 Connectivity Algorithm and Analysis

Our algorithm adapts a technique of Feige [13] for determining graph connectivity via a two step process. We first test whether G contains any connected components containing k or fewer nodes (for some $k < n$ to be chosen). If all of the connected components of G contain at least k nodes, we choose $O((n \log n)/k)$ nodes at random, and verify that they are all connected to each other. Note that we can expect to have chosen a vertex from each connected component. If we find that all of our chosen vertices are connected, we conclude that G is connected; otherwise, we conclude that G is disconnected.

Our connectivity algorithm thus relies on algorithms for two problems: 1) determining whether the graph has any connected components below a certain size, and 2) determining whether a set of nodes is mutually connected in the graph. In the next two sections, we develop algorithms with sample/space tradeoffs for each of these two problems. We then use them in an algorithm for determining whether the graph G is connected.

3.2.1 Finding Small Components

Our first subproblem is to determine whether the graph has any connected components below a certain size. Given a node v, let the set of nodes in the connected component containing v be denoted $\text{cc}(v)$.

▶ **Lemma 6.** *Given a node v of G and a parameter r, we can distinguish between the case where $|\text{cc}(v)| < r$ and the case where $|\text{cc}(v)| > 2r$ with constant probability using $O(mr^2)$ samples and $\tilde{O}(1)$ space.*

Proof. We perform a sampling walk of length $O(mr^2)$ samples. During this walk, we maintain a 1.1-approximation of the number of distinct vertices visited using an F_0 estimator [22]. If the estimated number of vertices visited is at least $3r/2$, we conclude that $|\text{cc}(v)| \geq r$; otherwise, we conclude that $|\text{cc}(v)| \leq 2r$.

If $|\text{cc}(v)| \leq r$, we will clearly visit at most r nodes. Our algorithm correctly concludes this so long as the F_0 estimator returns the promised approximation. If $|\text{cc}(v)| \geq 2r$, we need to argue that in $O(mr^2)$ samples we will hit at least $2r$ distinct nodes (except with constant probability). This follows from a result by Barnes [5, Thm 1.3] that states that for any connected (multi-)graph, it takes $O(\mathcal{MN})$ time in expectation to hit either $\mathcal{N}$ distinct nodes or $\mathcal{M}$ distinct edges. Using $\mathcal{M} = 2mr$ and $\mathcal{N} = 2r$ establishes the result. ◀

▶ **Theorem 7.** *We can determine whether G has a connected component of size less than 2^k using $O(p)$ space and $\tilde{O}(2^k \cdot mn/p)$ samples for any $p \leq n$.*

Proof. Our algorithm has k rounds, each corresponding to a value $r = 1, 2, 4, \ldots, 2^{k-1}$. In each round we reach one of two conclusions: 1) G has no connected components with size in the range $[r, 2r]$ or 2) there exists a connected component of G of size $< 3r$. All graphs satisfy at least one of these conclusions. We then determine G has no connected component of size less than 2^k if we never reach the first conclusion.

At a given value of r, we choose $O(n \log n/r)$ nodes, so that we hit any connected component of at least r nodes with high probability. From each node, we perform $\tilde{O}(1)$ random walks of length $\tilde{O}(mr^2)$ samples; from Lemma 6 this will suffice to determine with high probability whether any of these nodes is in a connected component of size $\leq 2r$.

We choose p nodes at a time, and perform p walks in parallel. From Theorem 3 we can perform each set of p walks using $\tilde{O}(mr^2)$ samples. The number of samples required for each r value is then $O(\frac{n}{rp}mr^2) = O(mnr/p)$, and we thus require a total number of samples $O(mn2^k/p)$. ◀

3.2.2 Checking Mutual Connectivity

The remaining subproblem is to determine whether a set of randomly-chosen nodes is mutually connected.

▶ **Lemma 8.** *We can determine whether a set of $O(p)$ randomly-chosen nodes is mutually connected in G using $\tilde{O}(p)$ space and $\tilde{O}(mn^2/p^2)$ samples for any $p \leq n$.*

Proof. In the context of traditional random walks, Feige [14] provides a method for testing whether two nodes s and t are connected using space $\tilde{O}(p)$ and a total of $\tilde{O}(mn/p)$ random-walk steps. They proceed by choosing p "landmark" nodes; we then run $O(\log n)$ random walks from each landmark and from s and t. Each walk is of length $\tilde{O}(mn/p^2)$. During these walks we build a union-find data structure indicating which sets of landmark nodes are connected. At the end of the algorithm, we conclude that s and t are connected if they are in the same union-find component.

Since H is regular, the landmark selection process chooses each node with equal probability. Using Feige's algorithm on the p randomly-chosen landmarks determines whether this set of p nodes is mutually connected. The graph H has n nodes and mn edges, so from [14] each walk should be of length $\tilde{O}(mn^2/p^2)$. Using Theorem 3 we can simulate the p walks with a total of $\tilde{O}(mn^2/p^2)$ samples. ◀

We are now ready to prove our main connectivity result.

▶ **Theorem 9.** *Given sampling access to a graph G, we can determine with high probability whether G is connected using $O(p \log n)$ space and $\tilde{O}(mn^2/p^2)$ samples, for any $p \leq n$.*

Proof. We use Theorem 7 with $2^k = n/p$ to verify that G has no connected components of size less than n/p. If it has such a component, then G is disconnected. If not, we choose

$O(p \log n)$ random vertices, hitting each remaining component with high probability. Using Lemma 8, we test that these vertices are mutually connected. Since we have chosen enough vertices to hit every connected component, this suffices to show that the graph is connected. Each of the two subproblems requires $O(mn^2/p^2)$ samples and $\tilde{O}(p)$ space, so these are the sample and space requirements of our algorithm. ◀

4 Rank Estimation

In this section we study the rank estimation problem, namely, that of distinguishing if a stream of vectors is coming from a full dimensional subspace or a subspace of low rank. This is quite useful in data streams and machine learning tasks, for which the vectors correspond to examples. If the subspace is rank-deficient, then one might be interested in a low rank approximation to it. This problem is also relevant in testing codewords, which has been studied in the streaming model in [36].

We will start with a lower bound and then mention a simple matching upper bound for a natural setting of parameters. The lower bound is described in terms of communication complexity, though it yields a lower bound on the memory for data stream algorithms via a standard simulation whereby the state of the streaming algorithm is the message between players in a communication protocol. We mention this after presenting the lower bound for the communication game.

The authors of [39] show a tight connection between learning tasks that can be accomplished via bounded communication algorithms and learning tasks that can be accomplished via "statistical query" algorithms. They operate in the following quite general communication model, and we leverage their result to show a space-sample lower bound trade-off for a streaming version of the rank estimation problem: the task is to distinguish

1. (Case 1): A sequence of t samples chosen uniformly from some rank $n/2$ subspace $S \subseteq \{0,1\}^n$.
2. (Case 2): A sequence of t samples chosen uniformly from $\{0,1\}^n$.

A "statistical query" algorithm for this task, in the language of [39], is an algorithm that adaptively proposes a sequence of functions $f_1, f_2, \ldots$ with $f_i : \{0,1\}^n \to [-1,1]$, and receives estimates of $E_x[f_i(x)]$ that have been corrupted via some adversarial noise. We say that there exists an n-query statistical query algorithm with tolerance τ for this testing task if, for every rank $n/2$ subspace S, after asking n statistical queries $f_1, \ldots, f_n$, with responses $r_1, \ldots, r_n$ that satisfy $|r_i - \mathrm{E}_{x \leftarrow unif[S]}[f_i(x)]| \leq \tau$, the algorithm will output *rank n/2* with probability at least 3/4, and if the responses satisfy $|r_i - \mathrm{E}_{x \leftarrow unif[\{0,1\}^n]}[f_i(x)]| \leq \tau$, then the algorithm will output *full rank* with probability at least 3/4, where the probability is over the randomness of the algorithm that decides on the next query f_i given $f_1, \ldots, f_{i-1}$ and $r_1, \ldots, r_{i-1}$.

▶ Proposition 10. Any statistical query algorithm for distinguishing the uniform distribution over rank $n/2$ subspaces of $\{0,1\}^n$ from a uniform distribution over $\{0,1\}^n$ using statistical queries of tolerance $\tau > \frac{1}{2^{n/8}}$ requires at least $\Theta(2^{n/4})$ statistical queries.

The following lemma is the core to the proof.

▶ **Lemma 11.** *Let $f : \{0,1\}^n \to [-1,1]$ be a function with $\sum_{x \in \{0,1\}^n} \frac{f(x)}{2^n} = \mu$, and let S be a rank $n/2$ subspace of $\{0,1\}^n$. Define the random variable X_f by choosing S uniformly at random from the set of all rank $n/2$ subspaces of $\{0,1\}^n$, and then set $X_f = E_{x \leftarrow S}[f(x)]$.*

$$\Pr[|X_f - \mu| > \frac{1}{2^{n/8}}] = O(1/2^{n/4}).$$

Proof. First note that $\mathrm{E}[X_f] \in [\mu - 1/2^{n/2}, \mu + 1/2^{n/2}]$, as the distribution obtained by selecting a random rank $n/2$ subspace S, then choosing a random $x \in S$ places probability $1/2^{n/2}$ on the zero vector $\vec{0}$, and the remaining $2^n - 1$ vectors have equal weight. We will now upper bound $\mathrm{Var}[X_f]$, and then apply Chebyshev's inequality. In the following calculations, $\#S$ denotes the number of rank $n/2$ subspaces of $\{0,1\}^n$.

$$\begin{aligned}\mathrm{E}[X_f^2] &= \mathrm{E}_S\left[\left(\frac{1}{2^{n/2}}\sum_{x\in S} f(x)\right)^2\right] = \mathrm{E}_S\left[\frac{1}{2^n}\sum_{x\in S} f(x)^2 + \frac{1}{2^n}\sum_{x,x'\in S, x\neq x'} f(x)f(x')\right] \\ &\leq 1/2^{n/2} + \frac{1}{2^n}\frac{1}{\#S}\sum_{S}\sum_{x,x'\in S, x\neq x'} f(x)f(x'),\end{aligned}$$

which is equal to

$$2^{-n/2} + \frac{2^{-n}}{\#S}\left(2f(\vec{0})\frac{\#S}{2^{n/2}-1}\sum_{x'\neq 0} f(x') + \sum_{x\neq\vec{0}} f(x)\frac{\#S}{(2^{n/2}-1)(2^{n/2}-2)}\sum_{x'\notin\{\vec{0},x\}} f(x')\right).$$

Noting that $|f(x)| \leq 1$, and $\sum_{x\neq\vec{0}} f(x) \in [2^n\mu - 1, 2^n\mu + 1]$, and $\sum_{x'\notin\{\vec{0},x\}} f(x') \in [2^n\mu - 2, 2^n\mu + 2]$, the above expression lies in the range $\mu \pm O(1/2^{n/2})$. Hence $\mathrm{Var}[X_f] = O(1/2^{n/2})$, and $\Pr[|X_f - \mu| \geq 1/2^{n/8}] \leq O(1/2^{n/4})$, as desired. ◀

Proof of Proposition 10. Let distribution D be defined to be the uniform distribution over $\{0,1\}^n$, and let S be a rank $n/2$ subspace that has been chosen uniformly at random from the set of rank $n/2$ subspaces of $\{0,1\}^n$, and define D_S to be the uniform distribution over S. Let $f_1, \ldots, f_n$ be a sequence of $n = \Theta(2^{n/4})$ statistical queries, corresponding to the responses $r_1, \ldots, r_n$ with $r_i = \mathrm{E}_{x\leftarrow D}[f(x)]$. We will now show that, with probability greater than $1/2$ over the randomness of the choice of subspace S, it will hold that $|r_i - \mathrm{E}_{x\leftarrow D_S}[f_i(x)]| \leq 1/2^{n/8}$. Indeed, this follows immediately from Lemma 11 via a union bound over the n events. To conclude, this shows that after n queries of tolerance at most $1/2^{n/8}$, with probability at least $1/2$, no information is given regarding whether the responses correspond to D versus D_S for some S, hence the probability of correctly guessing "D" versus "D_S" can be at most $3/4$ in one of the two cases. ◀

For the purposes of our lower bound, Lemma 3.3 from [39] (restated below) immediately translates the statistical query lower bound of Proposition 10 into a lower bound on the number of samples required by any algorithm in the following bounded communication model: there is one player per example, and the t players each speak once, one after the other. The i-th player sees the message sent of each of the first $i-1$ players, and then sends its message.

▶ **Lemma 12** (Lemma 3.3 from [39] (restated)). *If a given testing (or learning) task can be solved with probability $> 1 - \delta$ using t examples via a communication bounded protocol that employs s bits of communication per example then it can be solved with probability $> 1 - 2\delta$ via a statistical query algorithm that asks $2st$ statistical queries of tolerance $\tau = \delta/(t2^s)$.*

Proposition 10 and Lemma 12 yield the following:

▶ **Theorem 13.** *Any bounded-communication protocol that distinguishes samples from a rank $n/2$ subspace of $\{0,1\}^n$ with constant probability of success above $1/2$ that uses $s \leq n/8$ bits of communication per example requires at least $\Theta(2^{n/8-s})$ samples.*

Streaming Lower Bound: This communication bound implies an equivalent bound on the bits of memory required by any streaming algorithm, via the standard technique of noting that the memory contents of the streaming algorithm after seeing each example "communicates" information between examples of the input stream. Note that this lower bound holds even if the lower bound of Theorem 13 were instead only to hold in the communication model in which the i-th player only sees the message sent of the $(i-1)$-st player.

Nearly Matching Upper Bound: Our lower bound is tight to within constant factors for the natural case of $s = O(\log n)$ and $t = O(2^n)$, where a naïve algorithm suffices: choose a random coordinate unit vector $x = e_i$. Note that for any rank $n/2$ space, at most $n/2$ of these can lie in it. Take $O(2^n)$ samples, looking to see if any of them equal x. If we are sampling from a rank-n space, then we will find x with high constant probability; if we are sampling from a rank $n/2$ space, we will find it with probability at most $1/2$.

We note that very recently the work of Raz [34] improves upon the framework of [39]. It may be possible to apply it to strengthen the above theorem to require $b = \Omega(n^2)$ bits of space with fewer than $2^{\Omega(n)}$ samples, though it is not immediate. In either case, our result shows an exponential number of samples is needed to achieve polylogarithmic memory by a data stream algorithm, and we do not know how to prove this in a different way, e.g., by using more standard reductions from communication complexity.

5 Conclusions

We have introduced a new model for analyzing tradeoffs between sample and space complexity of streaming algorithms.

There are a number of open questions, a few of which we list here:

1. (Collision Probability) For the collision probability question, what is the optimal space complexity? We conjecture that $s \cdot t = \tilde{\Omega}(n)$ should hold. After resolving the dependence on n, a next natural question would be to understand the dependence on ϵ and δ. Recent work [1] may be helpful in this regard.
2. (Frequeny Moments) Can one obtain optimal tradeoffs for other frequency moments $F_k = \sum_j p_j^k$? In this work we focused solely on F_2. Perhaps more generally one could provide a general set of techniques for analyzing various distribution learning problems in this framework.
3. (Connectivity) What lower bounds can one show for testing connectivity? It seems we can show some preliminary lower bounds via a reduction from the set disjointness problem, though currently they are far from the upper bounds.
4. (General) The techniques used for the different problems studied here are not directly related to each other. Is it possible to develop a more general framework which unifies the results for these problems?

Acknowledgments: We thank the anonymous referees for their helpful comments.

References

1 Jayadev Acharya, Alon Orlitsky, Ananda Theertha Suresh, and Himanshu Tyagi. The complexity of estimating rényi entropy. *CoRR*, abs/1408.1000, 2014.

2 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

3 Alexandr Andoni, Andrew McGregor, Krzysztof Onak, and Rina Panigrahy. Better bounds for frequency moments in random-order streams. *CoRR*, abs/0808.2222, 2008.

4 Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. Sampling algorithms: lower bounds and applications. In *STOC*, pages 266–275, 2001.

5 Greg Barnes and Uriel Feige. Short Random Walks on Graphs. *SIAM Journal on Discrete Mathematics*, 9(1):19, 1996. URL: http://dblp.uni-trier.de/db/journals/siamdm/siamdm9.html#BarnesF96, doi:10.1137/S0895480194264988.

6 Marc Bury and Chris Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. *CoRR*, abs/1505.02019, 2015. URL: http://arxiv.org/abs/1505.02019.

7 Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. In *STOC*, pages 641–650, 2008.

8 Amit Chakrabarti, T. S. Jayram, and Mihai Patrascu. Tight lower bounds for selection in randomly ordered streams. In *SODA*, pages 720–729, 2008.

9 Steve Chien, Katrina Ligett, and Andrew McGregor. Space-efficient estimation of robust statistics and distribution testing. In *ICS*, pages 251–265, 2010.

10 Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 205–214, 2009.

11 Thomas M. Cover, Michael A. Freedman, and Martin E. Hellman. Optimal finite memory learning algorithms for the finite sample problem. *Information and Control*, 30(1):49–85, 1976.

12 Klim Efremenko and Omer Reingold. How Well Do Random Walks Parallelize? In *APPROX-RANDOM*, volume 5687, pages 476–489, Berlin, Heidelberg, 2009. URL: http://dblp.uni-trier.de/db/conf/approx/approx2009.html#EfremenkoR09, doi:10.1007/978-3-642-03685-9.

13 Uriel Feige. A fast randomized logspace algorithm for graph connectivity. *Theor. Comput. Sci.*, 169(2):147–160, 1996.

14 Uriel Feige. A Spectrum of Time-Space Trade-offs for Undirected s-t Connectivity. *Journal of Computer and System Sciences*, 54(2):305–316, April 1997. URL: http://dblp.uni-trier.de/db/journals/jcss/jcss54.html#Feige97, doi:10.1006/jcss.1997.1471.

15 R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368, 1922.

16 I.J. Good. Surprise indexes and p-values. *J. Statistical Computation and Simulation*, 32:90–92, 1989.

17 Michael Greenwald and Sanjeev Khanna. Efficient online computation of quantile summaries. In *ACM International Conference on Management of Data*, pages 58–66, 2001.

18 Sudipto Guha and Zhiyi Huang. Revisiting the direct sum theorem and space lower bounds in random order streams. In *ICALP*, pages 513–524, 2009.

19 Sudipto Guha and Andrew McGregor. Space-efficient sampling. In *AISTATS*, pages 169–176, 2007.

20 Sudipto Guha and Andrew McGregor. Stream order and order statistics: Quantile estimation in random-order streams. *SIAM Journal on Computing*, 38(5):2044–2059, 2009. URL: http://link.aip.org/link/?SMJ/38/2044/1, doi:10.1137/07069328X.

21 Martin E Hellman and Thomas M Cover. Learning with finite memory. *The Annals of Mathematical Statistics*, pages 765–782, 1970.

22 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *PODS*, pages 41–52, 2010.

23 Jack Koplowitz. Necessary and sufficient memory size for m-hypothesis testing. *Information Theory, IEEE Transactions on*, 21(1):44–46, 1975.

24 Frank Thomson Leighton and Ronald L. Rivest. Estimating a probability using finite memory. *IEEE Trans. Information Theory*, 32(6):733–742, 1986.

25 Yi Li, Huy L. Nguyen, and David P. Woodruff. On sketching matrix norms and the top singular vector. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1562–1581, 2014.

26 Andrew McGregor, A. Pavan, Srikanta Tirthapura, and David P. Woodruff. Space-efficient estimation of statistics over sub-sampled streams. In *PODS*, pages 273–282, 2012.

27 Andrew McGregor and Paul Valiant. The shifting sands algorithm. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 453–458, 2012.

28 Michael Mitzenmacher and Eli Upfal. *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

29 J. Ian Munro and Mike Paterson. Selection and sorting with limited storage. *Theor. Comput. Sci.*, 12:315–323, 1980.

30 S. Muthukrishnan. Stochastic data streams. In *MFCS*, page 55, 2009.

31 AS Nemirovsky and DB Yudin. *Problem Complexity and Method Efficiency in Optimization*. J. Wiley @ Sons, New York, 1983.

32 Yurii Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

33 Piyush Rai, Hal Daumé III, and Suresh Venkatasubramanian. Streamed learning: One-pass SVMs. In *IJCAI*, pages 1211–1216, 2009.

34 Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. *CoRR*, abs/1602.05161, 2016.

35 Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4):1–24, September 2008. URL: `http://dblp.uni-trier.de/db/journals/jacm/jacm55.html#Reingold08`, `doi:10.1145/1391289.1391291`.

36 Atri Rudra and Steve Uurtamo. Data stream algorithms for codeword testing. In *Automata, Languages and Programming*, pages 629–640. Springer, 2010.

37 Florin Rusu and Alin Dobra. Sketching sampled data streams. In *ICDE*, pages 381–392, 2009.

38 Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 163–171, 2014.

39 J. Steinhardt, G. Valiant, and S. Wager. Memory, communication, and statistical queries. Manuscript, 2015.

40 TheJournal.ie. 50 billion instant messages expected to be sent each day in 2014. URL: `http://businessetc.thejournal.ie/sms-messaging-falls-1262017-Jan2014/`.

41 David P. Woodruff. The average-case complexity of counting distinct elements. In *ICDT*, pages 284–295, 2009.