Sublinear Space Private Algorithms Under the Sliding Window Model

Jalaj Upadhyay [1]

Abstract

The Differential privacy overview of Apple states, "Apple retains the collected data for a maximum of three months." Analysis on recent data is formalized by the *sliding window model*. This begs the question: what is the price of privacy in the sliding window model. In this paper, we study heavy hitters in the sliding window model with window size w. Previous works of Chan et al. (2012) estimates heavy hitters incur an error of order θw for a constant $\theta > 0$. In this paper, we give an efficient differentially private algorithm to estimate heavy hitters in the sliding window model with $\widetilde{O}(w^{3/4})$ additive error and using $\widetilde{O}(\sqrt{w})$ space.

1. Introduction

Many real-world applications involve processing time-evolving data. For example, consider a health agency tracking the spread of flu, or a network service provider monitoring the traffic to be able to allocate resources and preempt congestion, or a recommendation system following the trending events to be able to better serve the users. Given the temporal, streaming nature of data, a primary requirement from a computational perspective is to enable analysis in real time using small space. Furthermore, these systems often rely on highly sensitive, private data and therefore, privacy is of utmost concern in such applications. The focus, therefore, in this paper, is to study sublinear space algorithms for private analysis of temporal data.

Several computing models have been studied for processing temporal data. Perhaps, the most popular and the simplest of all is the *streaming model*, wherein, at every time epoch new data is received and processed but the output is produced only at the end of the stream. This model has been studied extensively in many privacy-preserving data analysis applications (Mir et al., 2011; Blocki et al., 2012; Upadhyay, 2014a;b; Joseph et al., 2018). A related model that has been studied with privacy considerations is that of *continual release model*, where the output is produced continually at every time epoch as the data streams, in Chan et al. (2011); Dwork et al. (2010a). Both these models consider the entire stream of data useful (and hence, also referred to as the unbounded stream models). This is in contrast with the *sliding window model* where the goal is to maintain a fixed length window over the stream and process queries based on the most recent data (Datar et al., 2002).

In this paper, we are primarily interested in the sliding window model since many real-world applications (including the ones listed above) value recent data more than the historical data. For example, Apple retains the data collected from a user for a maximum of three months, anything older than that is perhaps considered stale from the perspective of making recommendations (Apple, 2017; Thakurta et al., 2017a;b). A secondary motivation stems from privacy concerns – naturally, if Apple retains only recent data for any user, it limits the exposure to private data. From an algorithmic perspective, this poses an interesting question regarding the privacy-vs-utility tradeoff since working with a sliding window allows one to reclaim the privacy budget at every epoch. Therefore, given a fixed privacy budget, one can give a stronger privacy guarantee for a shorter sliding window but at the expense of less accurate analysis.

Unlike privacy in the unbounded streaming model, privacy in the sliding window model has largely been unexplored. The only private algorithm in the sliding window model, that we are aware of, is due to Chan et al. (2012) for finding heavy hitters, i.e., returning a list of elements in a stream that occur more than a certain number of times. The algorithm of Chan et al. (2012) incurs an additive error of θw (for a constant $\theta > 0$) over a sliding window of size w, and requires $O(w)$ space. We, on the other hand, are interested in sublinear space algorithm with $o(w)$ additive error.

The requirement of sublinear space is very natural. For example, consider network monitoring with a high-speed router working at 100 Gbps line speed. At every epoch, one of the objectives is to identify the IP address that injects the highest traffic to the network. A trivial solution to this problem requires 10^{11} bits of space. However, in the scenario of high-speed routers, where on-chip memory is expensive

[1] Department of Computer Science, Johns Hopkins University, Baltimore, MD - 21218. Correspondence to: Jalaj Upadhyay <jalaj@jhu.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

and multiple queries are made continuously, it is prohibitive to use 10^{11} bits of memory for each query.

Our main contribution in this paper is a differentially private algorithm for finding heavy hitters in the sliding window model with space complexity $o(w)$, and an additive error guarantee of $o(w)$ over the entire window.

1.1. Formal Problem Description

We first give an abstraction of problem description before stating the specific problem we study in this paper. For this, we fix the following notations. The stream consists of elements from a set $\mathscr{S}$. We use $\mathscr{Y}$ (and $\mathscr{R}$, respectively) to denote the domain (range, respectively) of the function to be evaluated. We use $\mathsf{id} \in \mathscr{Y}$ to denote the initialization value. We use $\mathscr{N}(\mu, \sigma^2)$ to denote a Gaussian distribution with mean μ and variance σ^2 and $\mathsf{Lap}(b)$ to denote a mean zero Laplace distribution with scale b. For a vector x, we use the notation $a_i(x)$ to denote its i-th coordinate.

An algorithm in the streaming model takes the following as input: (update function) $\mathscr{U} : \mathscr{S} \times \mathscr{Y} \to \mathscr{Y}$; (evaluation function) $f : \mathscr{Y} \to \mathscr{R}$; and (stream) a sequence $(s_t)_{t=1}^{\infty}$, where $s_t \in \mathscr{S}$.

The update function defines how the stream should be interpreted to define the input to the desired evaluation function. For an update function, we denote its output on a subsequence of stream $(s_n, \ldots, s_m)$ by $x_{[n,m]} := \mathscr{U}(s_m, \mathscr{U}(s_{m-1}, \ldots \mathscr{U}(s_n, \mathsf{id})) \ldots) \in \mathscr{Y}$. The goal of a streaming algorithm is to evaluate $f(x_{[1,t]})$ at any time t.

Let's consider a simple example to make the setup clear. An important problem from both numerical linear algebra perspective as well as learning theory, is the problem of principal component analysis.

Example 1 (Principal component analysis)**.** *Suppose* $\mathscr{X} = \mathbb{R}^d, \mathscr{Y} = \mathbb{R}^{d\times d}$*,* $\mathscr{R} = \mathbb{R}^{d\times k}$*, and* $\mathscr{U}(x_1, \ldots, x_t) = \sum_i x_i x_i^\top$*. Let* $\|\cdot\|_F$ *denotes the Frobenius norm of the matrix. Then* $f(\mathscr{U}(x_1, \ldots, x_t)) = \operatorname{argmin}_{U \in \mathbb{R}^{d\times k}} \left\| UU^\top - \sum_{i=1}^t x_i x_i^\top \right\|_F$.

The streaming model do not lend themselves to settings where the data are time-sensitive. In such settings, recent data are considered more accurate than data that arrived prior to a certain time window.

To model such settings, Datar et al. (2002) introduced the *sliding window model*. An algorithm in the sliding window model also takes the size of window w as input. The objective of the algorithm is to output $f(x_{[t-w+1,t]})$ at any time t, where $x_{[t-w+1,t]} \in \mathscr{Y}$ is formed by the last w updates. More formally, we define the sliding window model of computation as follows.

Definition 1. *Given a paramter w for the size of window, an input stream of updates, and a function of interest $f(\cdot)$, a randomized algorithm in the sliding window model yields a (T, α, β, τ) accurate estimate if, in every execution, with probability at least $1-\beta$, simultaneously, for all $1 \leq t \leq T$, after processing a prefix $(x_1, \ldots, x_t)$, the current output contains an estimate, f^*, such that $|f^* - f(x_{[t-w+1,t]})| \leq \alpha f(x_{[t-w+1,t]}) + \tau$.*

We use differential privacy as the notion of privacy. At a high level, differential privacy requires that the output distribution of computation on two "adjacent" dataset does not differ by "much." In the context of the sliding window, the dataset is given as a stream and notion of adjacent streams dictates what is private and what is not private. In this paper, the dataset is represented in the form of a vector.

We consider two streams as adjacent if they differ in a single time epoch in the current window by at most one. More precisely, we consider two streams $S = \{s_1, \ldots, s_t\}$ and $S' = \{s'_1, \ldots, s'_t\}$ adjacent at time t if the resulting n-dimensional vector x and x' formed by the last w updates differ in at most one entry by 1. That is, there is an $1 \leq i \leq n$ such that $|a_i(x) - a_i(x')| \leq 1$ and $a_j(x) = a_j(x')$ for $j \neq i$. This can be seen as the extension of *event-level privacy* (Dwork et al., 2010a; Chan et al., 2011) to the sliding window model. We now give the formal definition of differential privacy.

Definition 2 (Differential privacy)**.** *Let $\mathfrak{M} : \mathscr{X}^* \to \mathscr{R}$ be a randomized algorithm. $\mathfrak{M}$ is ε-differentially private if for every adjacent streams S and S' and every measurable set $\mathscr{C} \subseteq \mathscr{R}$, $\mathsf{Pr}[\mathfrak{M}(S) \in \mathscr{C}] \leq e^{\varepsilon}\mathsf{Pr}[\mathfrak{M}(S') \in \mathscr{C}]$.*

As mentioned earlier, one of the main problems studied in this paper is that of a heavy hitter defined by Cormode & Muthukrishnan (2005). They considered (ϕ, ρ)-heavy hitter, where we are required to output all elements whose frequency is at least $\phi \|x\|_1$ and not output any element whose frequency is less than $(\phi - \rho) \|x\|_1$, where x is the vector formed by the stream. An equivalent definition of heavy hitters also used in the literature is as follows:

Definition 3 ((ζ, γ)-sliding heavy hitters)**.** *Given a publicly known bounded universe $\mathscr{X} = \{1, \ldots, u\}$ and a stream of input $(x_t)_{t\geq 1}$, where each $x_t \in \{0,1\}^{|\mathscr{X}|}$, the problem of* (ζ, γ)-heavy hitters *is to return all $1 \leq i \leq u$ such that $a_i(x) \geq (1+\zeta)\gamma \|x\|_1$ at time T, where $x := \sum_{t=1}^T x_t$, along with the estimates of their frequency. Further, it should not output any $1 \leq j \leq u$ with $a_j(x) < (1 - c\zeta)\gamma \|x\|_1$ for some constant $c > 0$.*

It is easy to see that by setting ϕ and ρ appropriately, the above formulation is equivalent to the one considered by Cormode & Muthukrishnan (2005).

As one can observe, we use the set-up similar to Chan et al. (2012). In our set up, the universe and the individuals are separate. Every update can be seen as a user reporting an element of the universe. That is, we consider the underlying universe is publicly known. As such revealing the identity of the heavy hitter does not reveal the identity of the user. Our results are easily extendable to the setting where more than one users contribute to an update.

Our Contributions. Current state-of-the-art non-private algorithms in the sliding window model uses the *smooth histogram framework* of Braverman & Ostrovsky (2010) (see Section 2 for formal definition). A natural question is whether we can extend this framework to differential privacy. We give a counterexample showing that one cannot extend this framework to the private setting even for a single release of the output without incurring a prohibitively large additive error (Claim 1). On the positive side, we give an algorithm to estimate heavy hitters with accuracy $\widetilde{O}(w^{3/4}/\varepsilon)$ over the entire window (Theorem 4 and Theorem 5) that uses space sublinear in w. This significantly improves the previous result by Chan et al. (2012), which incurs θw additive error (for a constant $\theta > 0$). We then show that dependence of space on w is inevitable for a constant multiplicative approximation by giving a space lower bound for estimating heavy-hitters by any differentially private algorithm (Theorem 6). The lower bound on the space is for the case when we produce the output just once.

Challenges of the Sliding Window Model. It is not clear if existing techniques used in privacy-preserving algorithms can be used in the sliding window model. There are two techniques used in the previous works to deal with data in the streaming model: private version of Bentley & Saxe (1980)'s method and random projection based method. The private version of Bentley & Saxe (1980)'s static to dynamic data-structure transformation has been a key component of many previous works (Chan et al., 2011; 2012; Dwork et al., 2010a). Unfortunately, the construction of Bentley & Saxe (1980) does not allow "automatic" deletion that happens in the sliding window model (once a stream element is expired). One can still use the construction to output an estimate in the window of size w at the expense of using $O(N)$ space, where N is the total length of the stream. However, this is highly undesirable when $N \gg w$. In concise, it is unclear if we can use techniques based on Bentley & Saxe (1980) to provide even $O(w)$ space sliding window algorithm.

Another workhorse of low-space differentially private algorithms is randomized projections (Arora et al., 2018; Bassily & Smith, 2015; Bassily et al., 2017; Blocki et al., 2012; 2013; Kenthapadi et al., 2012; Liu et al., 2006; Upadhyay, 2013; 2018). All these techniques use linear sketches. However, once an input has been sketched, there is no way we can find out its influence on the sketch at a later point if we do not store the update. Since the sliding window has automatic deletion and not stream based deletion, it is unclear how to update the sketch due to the automatic deletion of the update that lies beyond the current window.

Finally, one may argue that we can use existing non-private algorithms in the sliding window model for heavy hitters and then produce a private estimate of their output. The only known (non-private) algorithm, that we are aware of, for heavy hitters in the sliding window model is that by Braverman et al. (2018b). However, while their algorithm can be used to estimate heavy hitters once, it is not clear how to extend it to estimate heavy hitters continually over the entire window because of some of the subroutines used in their algorithm.

Our Approach. Our approach is different from previous algorithms. At a high level, at any time T, we store a set of time-stamps that is defined using a known algorithm in the sliding window model for estimating frequency moment, F_1. Suppose these time stamps are $t_1 < t_2 < \cdots < t_s = T$. We privately compute F_1 of the vector formed during the time interval $[t_1, T]$ (say this is ℓ^*). We use the checkpoints corresponding to the time stamps $t_1, \ldots, t_s$ to execute s independent instances of private COUNT-MIN algorithm, say $CM_1, \ldots, CM_s$. These invocations results in a set of s vectors, $v_1, \ldots, v_s$. Here v_i is the vector formed by executing COUNT-MIN sketch on the n-dimensional vector formed during the time interval $[t_i, T]$. To compute the heavy hitter in the updates formed during the time interval $[T-w+1, T]$, we use the private estimation of the norm, i.e., ℓ^* with the sketch generated by the private COUNT-MIN instance CM_1. The analysis requires a delicate balance due to the combination of streaming algorithm and sliding window algorithm as well as the noise required to preserve privacy. The linearity of COUNT-MIN and estimation of ℓ_1 norm allows us to extend the result to continual observation over the entire window.

We believe this approach of combining techniques in the sliding window model along with the techniques in the streaming model of computation can be used to give private algorithms for many other practically important functions.

2. Prior related work

Without any space constraints, one can compute any computable function exactly in space $O(w)$ using the generic transformation based on the technique of Bentley & Saxe (1980). However, such algorithms are infeasible in the practical setting discussed in Section 1. Therefore, one primary objective in the sliding window model is to perform the computation using $o(w)$ space.

There are two main techniques that yield sublinear space algorithms in the sliding window model, one is the *exponential histogram* based approach for subadditive functions and the other is the *smooth histogram* based approach for (α, β)-*smooth functions*. Our algorithms use the smooth histogram framework for estimating certain functions of the streamed vector. To understand the framework, we first need the definition of a smooth function.

Let A, B, C be a subsequence of the stream such that B is a subsequence of A and C follows A. By abusing the notation, we denote by $f(A)$ the computation of the function f on the variable formed due to the updates in the subsequence A.

Definition 4 (Smooth Function)**.** *A function $f : \mathcal{X}^* \to \mathbb{R}$ is (α_1, α_2)-smooth (with $0 < \alpha_2 \leq \alpha_1 < 1$) if it has the following properties: (a) $\frac{1}{\mathsf{poly}(w)} \leq f(A) \leq \mathsf{poly}(w)$ for any subsequence A; (b) $f(B) \leq f(A)$ for $B \subseteq A$; and (c) For any subsequence A, B and C such that $B \subseteq C$, if $(1-\alpha_2)f(A) \leq f(B)$, then $(1-\alpha_1)f(A \cup C) \leq f(B \cup C)$.*

The smooth histogram data structure maintains a number of "checkpoints" throughout the data stream. Each checkpoint corresponds to a sketch of all the elements observed from the time of the checkpoint until the most recently arrived element. The most recent element received influences all the sketches in the smooth histogram. A checkpoint is created with the arrival of each new element and checkpoints are discarded when their corresponding sketches get "too close" to the next checkpoint. Braverman & Ostrovsky (2010) constructed a smooth histogram data structure for smooth functions and showed the following.

Theorem 1 (Braverman & Ostrovsky (2010))**.** *Let α be the desired approximation parameter. Let $f : \mathcal{X} \to \mathbb{R}$ be an (α_1, α_2)-smooth function for α_1 and α_2 some function of α. Let w be the sliding window parameter and $(d_t)_{t=1}^{\infty}$ be a data stream. Assume that there exists an algorithm in the insertion only model that computes the function f within $(1-\alpha, 1+\alpha)$ factor using S_f space and U_f update time. Then there is an efficient algorithm that computes the function $f(\cdot)$ in the sliding window model using $O(\frac{1}{\alpha_2}(S_f + \log w) \log w)$ space and $O(\frac{1}{\alpha_2} U_f \log w)$ operations.*

A direct corollary of Theorem 1 is the estimation of ℓ_1 norm. Braverman & Ostrovsky (2010) maintain $s = O(\frac{1}{\alpha} \log w)$ checkpoints to estimate frequency moment of a stream within $(1-\alpha, 1+\alpha)$ factor. The output of their algorithm is $\ell := \left\| x_{[t_1,t]} \right\|_1$, where $x_{[t_1,t]}$ is the vector formed between the first checkpoint and the current time. We refer the interested readers to Braverman & Ostrovsky (2010) for more details. They showed the following result.

Lemma 1 (Braverman & Ostrovsky (2010))**.** *There is a sliding window algorithm that on input a stream that forms a vector x in the current window and approximation parameter α, outputs an estimate $\ell \in \mathbb{R}$ such that $(1-\alpha)\ell \leq \|x\|_1 \leq (1+\alpha)\ell$ at time t. The space required by this algorithm is $O(\frac{1}{\alpha} \log w \log(1/\beta))$.*

Our algorithm uses an instantiation of COUNT-MIN algorithm by Cormode & Muthukrishnan (2005). We use the following result:

Theorem 2 ((Cormode & Muthukrishnan, 2005))**.** *For a stream x, let $a_{q_i}(x)$ be the actual frequency of element q_i and $\widehat{a}_{q_i}(x)$ is the frequency estimated by* COUNT-MIN *algorithm executed with parameters $(\gamma, \zeta/4, \beta)$. Then with probability at least $1-\beta$, it outputs all elements q_i with frequency $a_{q_i}(x) \geq (1+\zeta)\gamma \|x\|_1$ such that $|a_{q_i}(x) - \widehat{a}_{q_i}(x)| \leq \frac{3}{2}\gamma\zeta \|x\|_1$.*

In the context of privacy, Dwork et al. (2010a) were the first to study the continual observation model within the framework of differential privacy. They gave a matching upper and lower bound on the achievable accuracy for a large class of functions; however, they do not place any space restriction. Further, their bound does not address the setting where we may be willing to allow a small multiplicative approximation.

3. Sliding Window and Privacy

In view of Theorem 1, one may ask whether such reduction works even with privacy, i.e., can we just take the off-the-shelf algorithm for (α_1, α_2)-smooth functions and privatize it. Unfortunately, this is not the case as we show next. Following is a simple example when the sensitivity of a smooth function can be arbitrarily large.

Example 2 (Frequency moment)**.** *Consider the problem of privately computing the p-th* frequency moment, *$F_p(x) := \|x\|_1^p$, of an input stream $x := (x_t)_{t \geq 1}$ with integer entries. Braverman & Ostrovsky (2010) showed that $F_p(x)$ is $(\alpha, \alpha^p/p^p)$-smooth for $p \geq 1$. Consider two adjacent streams $S = (x_1, \cdots, x_i, \cdots, x_t)$ and $S' = (x_1, \cdots, x_i', \cdots, x_t)$. $F_p(\cdot)$ on the last w inputs for the streams S and S' would be*

$$F_p = \|x\|_1^p \,;\, F_p' = (a_i(x)+1)^p + \|x\|_1^p - (a_i(x))^p,$$

where x is the vector formed by the last w updates. The sensitivity of this function is $|F_p - F_p'| = (a_i(x)+1)^p - a_i(x)^p$, which can be arbitrariliy large if the vector has entries in the set of natural numbers, $\mathbb{N}$.

In fact, the following claim based on a standard packing argument shows that we can have an arbitrary large additive error. A detailed proof appears in the supplementary material.

Claim 1. *Let $\varepsilon < 1$ and $p > 2^{1+\varepsilon}$. Consider a set of data streams $x_{[t-w+1,t]}^{(1)}, \ldots, x_{[t-w+1,t]}^{(p)} \in \mathbb{R}^w$ such that*

$$|F_p(x_{[t-w+1]}^{(\ell)}) - F_p(x_{[t-w+1]}^{(\ell')})| \geq \xi$$

for some ξ. Any ε-differentially private algorithm for computing $F_p(\cdot)$ must add noise at least $\xi/4$.

Therefore, in addition to the smoothness, we require some additional assumption. One such assumption that is common in the literature of differential privacy is that is that of Lipschitzness. Making these assumptions, we can privately compute the ℓ_2 and ℓ_1 norm of the streamed vector in the sliding window (see supplementary material).

Theorem 3. *Let $(x_t)_{t\geq 1}$ be the stream. For a window of size w, let x be the vector formed by last w updates. Then there is an efficient ε-differentially private algorithm in the sliding window model that uses $O(\frac{1}{\alpha}\log w)$ space and outputs an estimate $\widetilde{\ell}_1$ at any time t such that, with probability at least $1-\beta$, $\left|\widetilde{\ell}_1 - \|x\|_1\right| \leq \alpha\|x\|_1 + O\left(\frac{1}{\epsilon}\log(1/\beta)\right)$.*

3.1. Private Heavy Hitter in the Sliding Window Model

We now turn our attention to privately estimate the heavy hitters in the sliding window model. Chan et al. (2012) also studied this problem under event-level privacy as studied in this paper. Let $\mathcal{X}$ be the universe from which the elements are picked. They consider that every update x_t increases the count of one element in the universe by *one*. In the vector notation, every update is a vector $x_t \in \{0,1\}^{|\mathcal{X}|}$ and we are interested in a $|\mathcal{X}|$ dimensional vector formed by the last w updates. In this set-up, their algorithm incur an additive error θw for a constant $\theta > 0$.

In this section, we first present an algorithm with space requirement sublinear in w while providing accuracy for one-time release of the estimate (Theorem 4). We then show that we can improve the total additive error to be sublinear in w over the entire window while using sublinear space (Theorem 5). We show the following result for Algorithm 1.

Theorem 4. *Let $(x_t)_{t\geq 1}$ be the streamed vector with $x_t \in \{0,1\}^{|\mathcal{X}|}$. For a window of size w, let $x := x_{[t-w+1,t]}$ be the vector formed by last w updates, i.e., $x = \sum_{t=T-w+1}^{T} x_t$. Let $\rho = \frac{3\gamma\zeta}{2}$ and $\phi = (1+\zeta)\gamma$, $k = O(\frac{1}{\phi})$. Then the following holds:*

1. PRIVATE-L1-HEAVY *(described in Algorithm 1) is ε-differentially private in the sliding window model.*

2. PRIVATE-L1-HEAVY *satisfies the following with probability at least $1-3\beta$: If $a_{q_i}(x) \geq (1+\zeta)\gamma\|x\|_1$, then $(q_i, a_{q_i}(x)) \in$ LIST, then*

$$|\widetilde{a}_{q_i}(x) - a_{q_i}(x)| \leq \frac{3\zeta\gamma}{2}\|x\|_1 + O\left(\frac{1}{\varepsilon}\log\left(\frac{k}{\beta}\right)\right).$$

   Further, LIST *does not include any element $j \in \{1,\ldots,n\}$ such that its frequency $x_i < \left(1-\frac{\zeta}{2}\right)\gamma\|x\|_1 - (2-\frac{13}{4}\zeta)\gamma\tau_1$ for τ_1 as in Algorithm 1; and (b)*

3. *The space required by* PRIVATE-L1-HEAVY *is $O\left(\frac{1}{\gamma\zeta}\log w\log^2(k/\beta)\log n\right)$ and the update time per new input is $O(\frac{1}{\zeta}\log w\log^2(k/\beta))$.*

```
Algorithm 1 PRIVATE-L1-HEAVY $((x_t)_{t\geq 1}; w; \varepsilon; (\gamma,\zeta,\beta)$
Require: A stream $((x_t)_{t\geq 1}$ with $x_t \in \mathbb{R}$, COUNT-MIN algorithm for estimating heavy hitter in the streaming model, window size $w$, privacy parameter $\varepsilon$, parameters $(\gamma,\zeta,\beta)$ for COUNT-MIN algorithm.
Ensure: LIST $:= \{(q_1, \widetilde{a}_{q_1}(x)), \ldots, (q_k, \widetilde{a}_{q_k}(x))\}$ of estimates of heavy hitters, where $k \leq 1/\gamma$.
Maintain checkpoints $t_1,\ldots,t_s$ and corresponding estimates of $\|x_{[1,t]}\|_1, \ldots, \|x_{[s,t]}\|_1$ using the sliding window algorithm of Braverman & Ostrovsky (2010) for $\ell_1$ norm with $\alpha = \zeta/2$ as an approximation parameter. Here $s = O(\frac{1}{\zeta}\log w\log(1/\beta))$. Set $k = O(\frac{1}{\gamma})$.
Compute $\|\widehat{x}_{[i,t]}\|_1 = \|x_{[i,t]}\|_1 + \mathsf{Lap}(2/\varepsilon)$ for all $1 \leq i \leq s$.
Execute COUNT-MIN with parameters $(\gamma,\zeta/4,\beta)$ on $x_{[1,t]}$ to compute a list $\widehat{L} := \{(q_1, \widehat{a}_{q_1}(x)), \ldots, (q_k, \widehat{a}_{q_k}(x))\}$.
Update: LIST $\leftarrow$ LIST $\cup \{q_i, \widetilde{a}_{q_i}(x)\}$ if $\widehat{a}_{q_i}(x) \geq \left((1-\frac{\zeta}{2})\gamma\|\widehat{x}_{[1,s]}\|_1 - \tau_1\right)$, where $\widetilde{a}_{q_i}(x) = \widehat{a}_{q_i}(x) + \mathsf{Lap}(2/\varepsilon)$ and $\tau_1 := O\left(\frac{1}{\varepsilon}\log(1/\beta)\right)$.
Output LIST.
```

We note that the multiplicative approximation in the frequency estimate of heavy hitter is the same as in Cormode & Muthukrishnan (2005) (see Theorem 2). The cost of privacy is in the form of additive error in the estimate of the frequencies of heavy hitters and in the guarantee for rejecting elements that have low frequencies. The cost of the algorithm in the sliding window is in the form of an $s = O(\frac{1}{\zeta}\log w\log(\beta/k))$ factor increase in the space requirement.

To compare with the non-private sliding window algorithm for heavy hitters by Braverman et al. (2018b), we achieve the same space bound (see Theorem 3 in Braverman et al. (2018b)) and more flexibility in the approximation parameter than their result. Our algorithm is arguably simpler than theirs while affording us to extend easily to the continual observation over the entire window (see Theorem 5). However, their algorithm is more general than ours. That is, their algorithm can estimate lp-heavy hitters for $0 < p < 2$.

For the accuracy bound, note that we can pick ζ and γ to be a small constant. In particular, we can pick $\rho = \zeta\gamma = \frac{1}{w^{1-\kappa}}$ for a small constant $\kappa > 0$ that would lead to a sublinear in w accuracy bound while making sure that the space required is also sublinear in w. This is in contrast with the non-private algorithm of Braverman et al. (2018b).

As a final remark, we note that all the constructions in this paper can be extended to the case when $x_t \in \mathbb{N}^{|\mathcal{X}|}$, where $\mathbb{N}$ denotes the set of natural numbers. However, for the ease of presentation, we only present it for binary vector.

Proof. We first give the correctness proof. Let $x := x_{[t-w+1,t]}$ be the vector formed by the window. Let $\left\|x_{[1,t]}\right\|_1, \dots, \left\|x_{[s,t]}\right\|_1$ be the value of the norm of the checkpoints and $\left\|\widehat{x}_{[1,t]}\right\|_1, \dots, \left\|\widehat{x}_{[s,t]}\right\|_1$ be the private estimates of these vectors. Then, with probability at least $1-\beta$, $\forall i \in [s], \ \left|\left\|x_{[i,s]}\right\|_1 - \left\|\widehat{x}_{[i,s]}\right\|_1\right| \leq \tau_1$.

The correctness proof now relies on two lemmata: Lemma 2 which states that the heavy elements are included in LIST and Lemma 3 which guarantees that non-heavy elements are discared with high probability.

Lemma 2. *Let* $L = \{(q_1, a_{q_1}(x)), \dots, (q_k, a_{q_k}(x))\}$ *be the true list of heavy hitters,* $\widehat{L} := \{(q_1, \widehat{a}_{q_1}(x)), \dots, (q_k, \widehat{a}_{q_k}(x))\}$ *be the list of heavy hitters (and their estimated frequency) returned by the* COUNT-MIN *algorithm, and* $\widetilde{L} := \{(q_1, \widetilde{a}_{q_1}(x)), \dots, (q_k, \widetilde{a}_{q_k}(x))\}$ *be the private computation of the heavy hitters. Then if* $a_{q_i} \geq (1+\zeta)\gamma \|x\|_1$*, then* $(q_i, \widetilde{a}_{q_i}(x)) \in \mathsf{LIST}$ *with probability at least* $1-2\beta$*.*

Proof. We prove Lemma 2 using two claims. Claim 2 guarantees that heavy elements are returned in Step 2 of the algorithm and Claim 3 asserts that these elements are included in LIST. The proofs of these claims rely on Theorem 2 and the approximation guarantee of Lemma 1.

Claim 2 (Heavy hitters are included in $\widehat{L}$)**.** *If* $a_{q_i}(x) \geq (1+\zeta)\gamma \|x\|_1$*, then* $q_i \in \widehat{L}$ *with probability at least* $1-\beta/k$*.*

Proof. First note that the output, ℓ, by the algorithm of Braverman & Ostrovsky (2010) is $x_{[1,t]}$. Therefore, $x_{[1,t]}$ is such that

$$(1-\alpha)\left\|x_{[1,t]}\right\|_1 \leq \|x\|_1 \leq (1+\alpha)\left\|x_{[1,t]}\right\|_1. \qquad (1)$$

Moreover, by the construction of Braverman & Ostrovsky (2010), the window starts after the first checkpoint and before the second check-point. That is x is a subsequence of $x_{[1,t]}$. Therefore, if $a_{q_i}(x) \geq (1+\zeta)\gamma\|x\|_1$, then we have the following set of inequalities:

$$\begin{aligned} a_{q_i}(x_{[1,t]}) &\geq a_{q_i}(x) \geq (1+\zeta)\gamma\|x\|_1 \\ &\geq (1+\zeta)(1-\alpha)\gamma\left\|x_{[1,t]}\right\|_1. \end{aligned}$$

Since $\alpha = \zeta/2$ and $\zeta < 1/2$, we have

$$(1+\zeta)(1-\alpha) = (1+\zeta)(1-\zeta/2) \geq (1+\frac{\zeta}{2}-\frac{\zeta^2}{2}) \geq (1+\frac{\zeta}{4}).$$

This implies $a_{q_i}(x_{[1,t]}) \geq (1+\zeta/4)\gamma\left\|x_{[1,t]}\right\|_1$ and using Theorem 2, it will be detected with probablity at least $1-\beta/k$. This completes the proof of the Claim 2. □

Claim 3 (Heavy hitters are included in LIST)**.** *If* $a_{q_i}(x) \geq \phi\|x\|_1$*, then the algorithm includes* q_i *in* LIST *with probability at least* $1-\beta/k$*.*

Proof. Using equation (1), if $a_{q_i}(x) \geq (1+\zeta)\gamma\|x\|_1$, then we have the following:

$$\begin{aligned} \widehat{a}_{q_i}(x_{[1,t]}) &\geq a_{q_i}(x_{[1,t]}) - \rho\left\|x_{[1,t]}\right\|_1 && \text{(Theorem 2)} \\ &\geq a_{q_i}(x) - \rho\left\|x_{[1,t]}\right\|_1 && \text{(by definition)} \\ &\geq a_{q_i}(x) - \frac{\rho}{(1-\alpha)}\|x\|_1 && \text{(Lemma 1)} \\ &\geq \left(1+\zeta-\frac{3\zeta}{2(1-\alpha)}\right)\gamma\|x\|_1. \end{aligned}$$

Since $\frac{\zeta}{(1-\alpha)} > 0$ and $\left\|\widehat{x}_{[1,t]}\right\|_1 \leq \left\|x_{[1,t]}\right\|_1 + \tau_1 \leq \frac{(1+\zeta)\gamma}{(1-\alpha)}\|x\|_1 + \tau_1$, we have that q_i is included in LIST with probability at least $1-\beta$. Claim 3 follows. □

Combining Claim 2 and Claim 3 completes the proof of Lemma 2 □

Claim 4. *If* $(q_i, \widetilde{a}_{q_i}(x)) \in \mathsf{LIST}$*, then* $|a_{q_i}(x) - \widetilde{a}_{q_i}(x)| \leq \frac{3\rho}{2}\|x\|_1 + \tau_2$ *with probability at least* $1-\beta/k$*, where* $\tau_2 := O\left(\frac{1}{\varepsilon}\log(1/\beta)\right)$.

Proof. Using triangle inequality, we have $|a_{q_i}(x) - \widetilde{a}_{q_i}(x)| \leq |a_{q_i}(x) - \widehat{a}_{q_i}(x)| + |\widehat{a}_{q_i}(x) - \widetilde{a}_{q_i}(x)| \leq \frac{3\rho}{2}\|x\|_1 + \tau_2$ with probability at least $1-\beta/k$. □

We next prove that with high probability, elements that do not occur with considerable frequency are not included.

Lemma 3 (Non-heavy elements are not included)**.** *Algorithm 1 does not include any element with frequency smaller than* $(1-c_1\zeta)\gamma\|x\|_1 - (2-\frac{13}{4}\zeta)\gamma\tau_1$ *with probability at least* $1-\beta$*.*

Proof. Let $(a_{q_1}(x), \dots, a_{q_k}(x)), (\widehat{a}_{q_1}(x), \dots, \widehat{a}_{q_k}(x))$, and $(\widetilde{a}_{q_1}(x), \dots, \widetilde{a}_{q_k}(x))$ be as before. Let $\phi = (1+\zeta)\gamma$ and $\rho = \frac{3}{2}\zeta\gamma$. Then $(1-\zeta/2)\gamma = (1+\zeta)\gamma - \frac{3}{2}\zeta\gamma = \phi - \rho$. We want to prove that no element q_i is included in LIST if its frequency is below $(\phi - 3\rho)\|x\|_1 - (2\phi - \frac{7}{2}\rho)\tau_1$. If q_i was included in LIST, then we have $\widehat{a}_{q_i}(x_{[t_1,t]}) \geq (\phi-\rho)(\left\|\widehat{x}_{[t_1,t]}\right\|_1 - \tau_1)$. That is,

$$\begin{aligned} a_{q_i}(x_{[t_1,t]}) &\geq \widehat{a}_{q_i}(x_{[t_1,t]}) - \frac{3\rho}{2}\left\|\widehat{x}_{[t_1,t]}\right\|_1 \\ &\geq \left(\phi - \rho - \frac{3\rho}{2}\right)\left\|\widehat{x}_{[t_1,t]}\right\|_1 - (\phi-\rho)\tau_1 \\ &= \left(\phi - \frac{5\rho}{2}\right)\left\|\widehat{x}_{[t_1,t]}\right\|_1 - (\phi-\rho)\tau_1 \\ &\geq \left(\phi - \frac{5\rho}{2}\right)\left\|x_{[t_1,t]}\right\|_1 - \left(2\phi - \frac{7\rho}{2}\right)\tau_1 \end{aligned}$$

with probability at least $1 - \beta/k$. Further, we have the following:

$$\|x\|_1 = a_{q_i}(x) + \sum_{j \neq q_i} a_j(x),$$
$$\left\|x_{[t_1,t]}\right\|_1 = a_{q_i}(x_{[t_1,t]}) + \sum_{j \neq q_i} a_j(x_{[t_1,t]}).$$

Now using equation (1), $\|x\|_1 \geq (1-\alpha)\left\|x_{[t_1,t]}\right\|_1$. Combined with $a_j(x_{[t_1,t]}) - a_j(x) \geq 0$, this implies that $a_{q_i}(x) \geq a_{q_i}(x_{[t_1,t]}) - \frac{\rho}{3}\left\|x_{[t_1,t]}\right\|_1$. That is,

$$\begin{aligned} a_{q_i}(x) &\geq \left(\phi - \frac{5\rho}{2} - \frac{\rho}{3}\right)\left\|x_{[t_1,t]}\right\|_1 - \left(2\phi - \frac{7\rho}{2}\right)\tau_1 \\ &= \left(\phi - \frac{17\rho}{6}\right)\left\|x_{[t_1,t]}\right\|_1 - \left(2\phi - \frac{7\rho}{2}\right)\tau_1 \\ &\geq (\phi - 3\rho)\|x\|_1 - \left(2\phi - \frac{7\rho}{2}\right)\tau_1. \end{aligned}$$

This completes the proof of Lemma 3 by putting the value of ϕ and ρ. □

Combining Lemmata 2 and 3 with Claim 4 completes the proof of accuracy guarantee of Theorem 4.

For the privacy proof, recall that we consider event level privacy and the universe size is publicly known. Every user at a time epoch picks an element in the universe; thereby, contributing a counter of one to the frequency x_t. As such, preserving the privacy of an event (or user if a user is restricted to contribute just once) means preserving the privacy of the estimates of the heavy hitters. Note that we use a private estimate of $\|x\|_1$. This is reminiscent of the Numeric- Sparse algorithm (Dwork & Roth, 2014). The privacy proof follows just like Numeric-Sparse algorithm using the fact that every entry of the COUNT-MIN can change by at most 1 under the adjacency relation considered in this paper. For the space bound, Cormode & Muthukrishnan (2005) showed that any instance of COUNT-MIN uses $O(\frac{1}{\gamma}\log(1/\beta)\log(|\mathscr{X}|))$ space. Using Lemma 1, the total space required is $s \cdot O\left(\gamma^{-1}\log(1/\beta)\log(|\mathscr{X}|)\right)$ as required. The update time follows from the fact that the update algorithm of Braverman & Ostrovsky (2010) is $O(s)$ and that of every COUNT-MIN sketch is $O(\log(1/\beta))$.

This completes the proof of Theorem 4. □

Theorem 4 assumes that the estimate of the frequencies of heavy hitters is produced just once. However, there are scenarios when we would like to estimate the frequencies of heavy hitters more than once. The following result follows as a simple corollary to Theorem 4 and the composition theorem of differential privacy (Dwork & Roth, 2014).

Corollary 1. *Let x be as in Theorem 4. Suppose Algorithm 1 releases the estimate of heavy hitters at most r times. Let $\rho = \frac{3\gamma\zeta}{2}$ and $\phi = (1+\zeta)\gamma$. Algorithm 1 is ε-differentially private in the sliding window model, and it outputs* $\mathsf{LIST} = \{(q_1, \widetilde{a}_1(x)), \ldots, (q_k\widetilde{a}_k(x))\}$ *in each of the release such that with probability at least $1 - O(\beta)$,*

$$|\widetilde{a}_{q_i}(x) - a_{q_i}(x)| \leq \rho\|x\|_1 + O\left(\frac{1}{\varepsilon}\log(1/\beta)\right)$$

for all $a_{q_i}(x) \geq \phi\|x\|_1$. Further, it does not include any element q_i such that its frequency $a_{q_i}(x) < (\phi - 3\rho)\|x\|_1 - (2\phi - \frac{7\rho}{2})\tau_1$ for τ_1 as in Algorithm 1 and a universal constant c_1. Furtheremore, the space required by Algorithm 1 is $O\left(\frac{1}{\rho}\log w\log^2(1/\beta)\log|\mathscr{X}|\right)$ and the update time per new input is $O(\frac{1}{\zeta}\log w\log^2(1/\beta))$.

3.2. Private Heavy Hitter Over the Entire Window

To reduce the additive error over the entire window, we divide the window into equal disjoint sub-windows of size $\sqrt{w}$ and run the first three steps of Algorithm 1 on each of the sub-window. We then output the list by using the aggregate of all the $\sqrt{w}$ COUNT-MIN sketches and the estimate of the norms of the vector formed by the partitioned window. Here, we exploit the linearity of COUNT-MIN sketch and ℓ_1 norm. We present the details of this algorithm in the supplementary material. We refer to this algorithm as PRIVATE-HEAVY. Using the analysis as in Section 3.1 and the result of Chan et al. (2011) (that is, every estimate requires evaluation of $\sqrt{w}$ sub-windows and every element occurs exactly $\sqrt{w}$ times over all the estimates – corresponding to the sub-window it is in), we arrive at the following result.

Theorem 5. *Let $(x_t)_{t\geq 1}$ be the streamed vector with $x_t \in \{0,1\}^{|\mathscr{X}|}$ and $x = \sum_{t=T-w+1}^{T} x_t$. Let $\rho = \frac{3\gamma\zeta}{2}$ and $\phi = (1+\zeta)\gamma$, $k = O(\frac{1}{\phi})$. Then the following holds:*

1. PRIVATE-HEAVY *is ε-differentially private algorithm in the sliding window model.*
2. *It outputs* $\mathsf{LIST} = \{(q_1, \widetilde{x}_{q_1}), \ldots, (q_k, \widetilde{x}_{q_k})\}$ *such that, with probability at least $1 - O(\beta)$, simultaneously over the entire window, following holds: For all $x_{q_i} \geq (1+\zeta)\gamma\|x\|_1$, where x_{q_i} is the value of q_i-th coordinate in x,*

$$|\widetilde{x}_{q_i} - x_{q_i}| \leq \frac{3\zeta\gamma}{2}\|x\|_1 + O\left(\frac{w^{3/4}}{\varepsilon}\log(w/\beta)\right).$$

   Further, LIST *does not include any element q_i such that $x_{q_i} < \left(1 - \frac{\zeta}{2}\right)\gamma\|x\|_1 - O(\frac{w^{3/4}}{\varepsilon}\log(w/\beta))$*

3. *The space required by* PRIVATE-HEAVY *is $O\left(\frac{\sqrt{w}}{\rho}\log w\log^2(w/\beta)\right)$.*

Algorithm 2 PRIVATE-HEAVY $((x_t)_{t\geq 1}; w; \varepsilon; \gamma; \beta; k)$

```
Require: A stream $(x_t)_{t\geq 1}$, where $x_t \in \mathbb{R}$, COUNT-MIN algorithm for estimating heavy hitter in the streaming model, window size $w$, privacy parameter $\varepsilon$, parameters $(\gamma, \zeta, \beta)$ for COUNT-MIN .
Ensure: $\mathsf{LIST} := \{(q_1, \widetilde{a}_{q_1}(x)), \dots, (q_k, \widetilde{a}_k(x))\}$.
Divide the window in to $\sqrt{w}$ smaller window, $W_1, W_2, \dots, W_{\sqrt{w}}$.
for $i = 1 \dots, \sqrt{w}$ do
    Maintain checkpoints $t_1, \dots, t_s$ and corresponding estimates of $\left\|x^{(i)}_{[1,t]}\right\|_1, \dots, \left\|x^{(i)}_{[s,t]}\right\|_1$ using the sliding window algorithm of Braverman & Ostrovsky (2010) for $\ell_1$ norm in the window $W_i$ with $\alpha = \zeta/2$ as an approximation parameter. Here $s = O(\frac{1}{\gamma} \log w \log(1/\beta))$. Set $k = 1/\gamma$.
    Compute $\left\|\widehat{x}^{(i)}_{[j,t]}\right\|_1 = \left\|x^{(i)}_{[j,t]}\right\|_1 + \mathsf{Lap}(2\sqrt{w}\log W/\varepsilon)$ for all $1 \leq j \leq s$.
    Form COUNT-MIN sketch of $x^{(i)}_{[j,t]}$ for $1 \leq j \leq s$ with parameters $(\gamma, \zeta/4, \beta/k)$ to form a vector $y^{(i)}_{[j,t]}$.
    Privatize by adding $\mathsf{Lap}(2\sqrt{w}/\epsilon)$ to each entry of the vector $y^{(i)}_{[j,t]}$. Let the resulting vector be $z^{(i)}_{[j,t]}$.
end for
Compute $\left\|\widehat{x}_{[1,t]}\right\|_1 = \sum_i \left\|\widehat{x}^{(i)}_{[1,t]}\right\|_1$. Use $y_{[1,s]} := \sum_i y^{(i)}_{[1,t]}$, the combined COUNT-MIN, to construct a list $\widehat{L} = \{(q_1, \widehat{a}_1(x)), \dots, (q_k, \widehat{a}_k(x))\}$ of estimated non-private heavy hitters. Let $\widetilde{L} = \{(q_1, \widetilde{a}_1(x)), \dots, (q_k, \widetilde{a}_k(x))\}$.
Include: $(q_i, \widetilde{a}_{q_i}(x))$ in LIST if $\widehat{a}_{q_i}(x) \geq \left((1 - \frac{\zeta}{2})\gamma \left\|\widehat{x}_{[1,s]}\right\|_1 - O\left(\frac{w^{3/4}}{\varepsilon}\log(1/\beta)\right)\right)$.
Output LIST.
```

To make a fair comparison with Chan et al. (2012), we set $\rho = \frac{1}{\sqrt{w}}$ and $c = O(1)$. When $\|x\|_1 = \Omega(w^{3/4})$, this gives a non-trivial accuracy guarantee. In particular, in the setting of Chan et al. (2012), we achieve non-trivial accuracy in $O(\sqrt{w})$ space if $\frac{1}{w^{1/4}}$ fraction of the updates are non-zero.

3.3. Space Lower Bound for Single Output

In the previous section, we gave an efficient differentially private algorithm for heavy hitters in the sliding window model using sublinear space. A natural question one can ask is whether it is possible to estimate heavy hitters in space independent of window size? We show that it is not the case and we need at least $O(\log w)$ space even if we want a constant multiplicative approximation. We do this by reducing the problem of computing private heavy hitter in the sliding window model to one-way communication complexity of *augmented index* problem (AIND):

Definition 5 (Augmented Index Problem (AIND))**.** *Alice has a string $x \in \{0,1\}^p$ and Bob has an index $i \in [p]$ along with $x_1, \dots, x_i$. The players wish to compute the value of x_i, the i-th coordinate of x.*

Miltersen et al. (1995) showed that solving the AIND using one round of communication is hard, even if the players have shared randomness, requires $\Omega(p)$ bits. We show a reduction showing that if we can estimate heavy hitters in space $o(\frac{1}{\gamma}\log w)$, then it can be used to solve AIND with communication $O(p)$ for appropriate choice of p, thereby contradicting the result of Miltersen et al. (1995). This gives us the following theorem; a detailed proof can be found in the supplementary material.

Theorem 6. *Any differentially private algorithm that returns the $(\zeta, \gamma, 2/3, w)$-heavy hitters in the sliding window model requires $\Omega\left(\frac{1}{\gamma\zeta}\log w\right)$ bits.*

4. Discussion and Future Work

Our technique gives a new way to design and analyse differentially private algorithm in the sliding window model to give a non-trivial accuracy bound while using just $o(w)$ space. This significantly improves the best known result for heavy hitters. We also showed a lower bound on the space required by differentially private algorithm in the sliding window model.

Our results pose several interesting open questions. One important question is to understand the optimal space requirement of private heavy hitter algorithm simultaneously over the entire window. Another question is to understand whether we can improve the accuracy guarantee to be polylogarithmic in w.

Acknowledgements. This research was supported in part by NSF BIGDATA grants IIS-1546482 and IIS-1838139. The author would like to thank Raman Arora for discussion and his useful feedback on the presentation of the paper. The author would like to thank Nikita Ivkin for helpful discussions and directing to the paper of (Braverman et al., 2018b). A part of the idea came during a discussion on matrix multiplication with David Woodruff and Samson Zhou on a different project (Braverman et al., 2018a)..

References

Apple. Differential privacy, overview `https://images.apple.com/privacy/docs/Differential_Privacy_Overview.pdf`. 2017.

Arora, R., Braverman, V., and Upadhyay, J. Differentially private robust low-rank approximation. In *Advances in*

Neural Information Processing Systems, pp. 4137–4145, 2018.

Bassily, R. and Smith, A. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 127–135. ACM, 2015.

Bassily, R., Nissim, K., Stemmer, U., and Thakurta, A. G. Practical locally private heavy hitters. In *Advances in Neural Information Processing Systems*, pp. 2288–2296, 2017.

Bentley, J. L. and Saxe, J. B. Decomposable searching problems i. static-to-dynamic transformation. *Journal of Algorithms*, 1(4):301–358, 1980.

Blocki, J., Blum, A., Datta, A., and Sheffet, O. The johnson-lindenstrauss transform itself preserves differential privacy. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pp. 410–419. IEEE, 2012.

Blocki, J., Blum, A., Datta, A., and Sheffet, O. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pp. 87–96. ACM, 2013.

Braverman, V. and Ostrovsky, R. Effective computations on sliding windows. *SIAM J. Comput.*, 39(6):2113–2131, 2010. doi: 10.1137/090749281. URL `https://doi.org/10.1137/090749281`.

Braverman, V., Drineas, P., Upadhyay, J., and Zhou, S. Numerical linear algebra in the sliding window model. *arXiv preprint arXiv:1805.03765*, 2018a.

Braverman, V., Grigorescu, E., Lang, H., Woodruff, D. P., and Zhou, S. Nearly optimal distinct elements and heavy hitters on sliding windows. *arXiv preprint arXiv:1805.00212*, 2018b.

Bun, M., Nelson, J., and Stemmer, U. Heavy hitters and the structure of local privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 435–447. ACM, 2018.

Chan, T. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26:1–26:24, 2011. doi: 10.1145/2043621.2043626. URL `http://doi.acm.org/10.1145/2043621.2043626`.

Chan, T.-H. H., Li, M., Shi, E., and Xu, W. Differentially private continual monitoring of heavy hitters from distributed streams. In *International Symposium on Privacy Enhancing Technologies Symposium*, pp. 140–159. Springer, 2012.

Cormode, G. and Muthukrishnan, S. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

Datar, M., Gionis, A., Indyk, P., and Motwani, R. Maintaining stream statistics over sliding windows. *SIAM J. Comput.*, 31(6):1794–1813, 2002. doi: 10.1137/S0097539701398363. URL `https://doi.org/10.1137/S0097539701398363`.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pp. 715–724, 2010a. doi: 10.1145/1806689.1806787. URL `http://doi.acm.org/10.1145/1806689.1806787`.

Dwork, C., Naor, M., Pitassi, T., Rothblum, G. N., and Yekhanin, S. Pan-private streaming algorithms. In *ICS*, pp. 66–80, 2010b.

Erlingsson, Ú., Pihur, V., and Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067. ACM, 2014.

Hsu, J., Khanna, S., and Roth, A. Distributed private heavy hitters. In *International Colloquium on Automata, Languages, and Programming*, pp. 461–472. Springer, 2012.

Joseph, M., Roth, A., Ullman, J., and Waggoner, B. Local differential privacy for evolving data. *arXiv preprint arXiv:1802.07128*, 2018.

Kenthapadi, K., Korolova, A., Mironov, I., and Mishra, N. Privacy via the johnson-lindenstrauss transform. *arXiv preprint arXiv:1204.2606*, 2012.

Liu, K., Kargupta, H., and Ryan, J. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on knowledge and Data Engineering*, 18(1):92–106, 2006.

Miltersen, P. B., Nisan, N., Safra, S., and Wigderson, A. On data structures and asymmetric communication complexity. In *STOC*, pp. 103–111. ACM, 1995.

Mir, D., Muthukrishnan, S., Nikolov, A., and Wright, R. N. Pan-private algorithms via statistics on sketches. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 37–48. ACM, 2011.

Thakurta, A. G., Vyrros, A. H., Vaishampayan, U. S., Kapoor, G., Freudiger, J., Sridhar, V. R., and Davidson, D. Learning new words, March 14 2017a. US Patent 9,594,741.

Thakurta, A. G., Vyrros, A. H., Vaishampayan, U. S., Kapoor, G., Freudinger, J., Prakash, V. V., Legendre, A., and Duplinsky, S. Emoji frequency detection and deep link frequency, July 11 2017b. US Patent 9,705,908.

Upadhyay, J. Random Projections, Graph Sparsification, and Differential Privacy. In *ASIACRYPT (1)*, pp. 276–295, 2013.

Upadhyay, J. Differentially private linear algebra in the streaming model. *arXiv preprint arXiv:1409.5414*, 2014a.

Upadhyay, J. Randomness efficient fast-johnson-lindenstrauss transform with applications in differential privacy and compressed sensing. *arXiv preprint arXiv:1410.2470*, 2014b.

Upadhyay, J. The price of privacy for low-rank factorization. In *Advances in Neural Information Processing Systems*, pp. 4180–4191, 2018.