Supplementary Material to “Sublinear Space Private Algorithms Under the Sliding Window Model”

5. Vector Norm Estimation

Note that the negative result of the previous section rules out privatizing a generic reduction given by Braverman & Ostrovsky (2010). However, there may exist smooth functions that can still be compute privately under the sliding window model. As a warm up, we show that computing the ℓ_2 norm is possible under privacy in the sliding window.

Algorithm 3 L2-ESTIMATE $((x_t)_{t\geq 1}; w; (\varepsilon, \delta); (\alpha, \beta))$

```
Require: A stream $(x_t)_{t\geq 1}$, window size $w$, privacy parameter $(\varepsilon, \delta)$, approximation parameter $\alpha$, and confidence parameter $\beta$.
Ensure: An estimate, $\widetilde{\mathsf{L}}_2$, of $\ell_2$ norm.
  Maintain checkpoints $t_1, \dots, t_s$ and corresponding estimates of $\left\|x_{[1,t]}\right\|_1, \dots, \left\|x_{[s,t]}\right\|_1$ using the sliding window algorithm of Braverman & Ostrovsky (2010) for $\ell_2$ norm to get $(1 \pm \alpha)$ estimate of $\mathsf{L}_2$. Here $s = O(\frac{1}{\alpha^2}\log w)$.
  Set $\sigma^2 := \frac{4\log(1/\delta)}{\varepsilon^2}$.
  Privatize $\|x_{[1,t]}\|_2, \dots, \|x_{[s,t]}\|_2$ by using Gaussian mechanism, i.e, $\|\widehat{x}_{[i,t]}\|_2 = \|x_{[i,t]}\|_2 + e_i$ for all $1 \leq i \leq s$, where $e_i \sim \mathcal{N}(0, \sigma^2)$.
  Output $\widetilde{\ell}_2 := \|\widehat{x}_{[1,t]}\|_2$.
```

The proposed algorithm simply adds noise with appropriate variance to the sketch at each checkpoint computed by Braverman & Ostrovsky (2010). Algorithm 3 describes this in more detail for one release of ℓ_2 norm. For this algorithm, we show the following.

Theorem 7. *Let $(x_t)_{t\geq 1}$ be the stream. For a window of size w, let $x = x_{[t-w+1,t]} := \mathscr{U}(x_{t-w+1}, \dots, x_t)$. Then Algorithm 3 is (ε, δ)-differentially private algorithm in the sliding window model that uses $O(\frac{1}{\alpha^2}\log w)$ space and outputs an estimate $\widetilde{\ell}_2$ at any time t such that, with probability at least $1-\beta$,*

$$\left|\widetilde{\ell}_2 - \|x\|_2\right| \leq \alpha\|x\|_2 + O\left(\frac{1}{\epsilon}\sqrt{\log(1/\delta)}\log(1/\beta)\right).$$

In fact, by instantiating Algorithm 3 with $e_i \sim \mathsf{Lap}(2/\varepsilon)$ and using the framework of Braverman & Ostrovsky (2010) on ℓ_1 norm, we can privately estimate the ℓ_1 norm. Let us call this algorithm L1-ESTIMATE. The same proof technique as for Theorem 7 and using the concentration result for Laplace distribution gives us the following result.

Theorem 8. *Let $(x_t)_{t\geq 1}$ be the streamed vector. For a window of size w, let $x = \mathscr{U}(x_{[t-w+1,t]}) := (x_{t-w+1}, \dots, x_t)$. Then* L1-ESTIMATE *is $(\varepsilon, 0)$-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 $1-\beta$,*

$$\left|\widetilde{\ell}_1 - \|x\|_1\right| \leq \alpha\,\|x\|_1 + O\left(\frac{1}{\epsilon}\log(1/\beta)\right).$$

6. Missing Proofs

Proof of Claim 1. Consider any private algorithm $P(\cdot)$ for computing $F_p(\cdot)$ and incurs additive error less than $w/2c$ for some constant c. For the sake of brevity, let us write $x^{(\ell)}$ instead of $x^{(\ell)}_{[t-w+1,t]}$ Consider any $x^{(\ell)}$ and a subset T that is within ℓ_∞ ball of radius $\xi/4c$ around $F_p(x^{(\ell')})$ (see, Figure 1). That is,

$$T \subseteq \mathscr{B}_\infty\left(F_p(x^{(\ell')}), \frac{\xi}{4c}\right).$$

Since all ℓ_∞ balls are disjoint, i.e.,

$$|F_p(x^{(\ell)}) - F_p(x^{(\ell)})| \geq \xi/c,$$

there is an $\ell' \neq \ell$ such that for all $\tau \in T$ such that $|\tau - F_p(x_{[t-w+1],\ell'})| \leq \xi/2c$ and

$$\begin{aligned} \mathsf{Pr}_{\tau \in P(x^{(\ell)})}[p \in T] \leq \frac{1}{p} \quad \text{and} \\ \mathsf{Pr}_{\tau \in P(x^{(\ell')})}[p \in T] \geq 1/2. \end{aligned}$$

This implies that

$$\frac{\mathsf{Pr}_{\tau \in P(x^{(\ell)})}[\tau \in T]}{\mathsf{Pr}_{\tau \in P(x^{(\ell')}}[\tau \in T]} \leq \frac{2}{p}.$$

Further, from differential privacy and the fact that $\left\| x^{(\ell)} - x^{(\ell')} \right\|_1 \leq 1$, we have

$$\frac{\mathsf{Pr}_{\tau \in P(x^{(\ell)})}[\tau \in T]}{\mathsf{Pr}_{\tau \in P(x^{(\ell')})}[\tau \in T]} \geq 2^{-\varepsilon}.$$

If $p > 2^{1+\varepsilon}$, then we have a contradiction. □

Proof of Theorem 7. Let $x = \mathcal{U}(x_{[t-w+1,t]})$. Braverman & Ostrovsky (2010) showed the ℓ_2 norm is $(\alpha, \alpha^2/2)$-smooth. The space bound follows directly from $(\alpha, \alpha^2/2)$-smoothness and Theorem 1. Using Theorem 1, we also have

$$(1-\alpha)\|x_{[1,t]}\|_2 \leq \|x\|_2 \leq \|x_{[1,t]}\|_2 \leq (1+\alpha)\|x_{[1,t]}\|_2.$$

Using the concentration of Gaussian distribution, we have

$$\left| \|x_{[1,t]}\|_2 - \|\widehat{x}_{[1,t]}\|_2 \right| \leq O(\sigma \log(1/\beta))$$

with probability $1-\beta$. Combining the two, we get the following:

$$(1-\alpha)\|x\|_2 - \sigma \leq \|\widehat{x}_{[1,t]}\|_2 \leq (1+\alpha)\|x\|_2 + \sigma$$

for small α as required. □

Towards proving the accuracy guarantee of heavy hitter in the sliding window model, we first show that we can compute $\mathfrak{P}(f, \mathcal{U}, i, j)$ when $f = \mathsf{L}_1$. Using monotonicity of L_1 and the smooth histogram framework of Braverman & Ostrovsky (2010), we can show the following lemma; a proof can be found in the supplementary material:

Lemma 4 (Braverman & Ostrovsky (2010))**.** *There is an efficient data-structure that can be used to estimate $\mathsf{L}_1(x_{[i,t]})$ for every index $i \in [t-w+1, t]$ for time epoch t. That is, for a constant $\alpha \in (0,1)$, we can compute ℓ_1^* such that $(1-\alpha)\ell_1^* \leq \mathsf{L}_1(\mathcal{U}(x_{[i,t]})) \leq \ell_1^*$.*

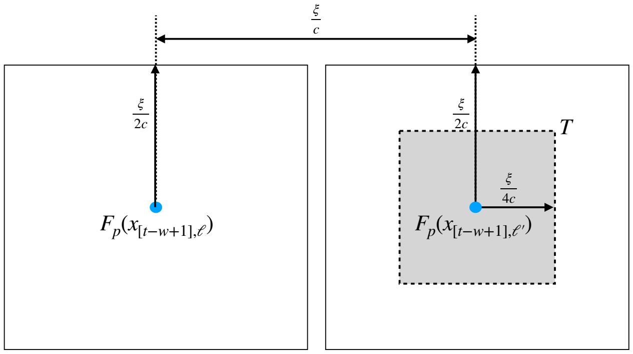

Figure 1. Lower bound proof

Proof Sketch of Lemma 4. The data structure used to derive the above lemma is shown diagrammatically in Figure 2. We use the same algorithm as Braverman & Ostrovsky (2010) to store a smooth histogram with checkpoints $t_1, \dots, t_s = t$ and the corresponding function evaluation such that the following invariance is maintained: (i) $(1-\beta) f(x_{[t_j,t]}) \geq f(x_{[t_{j+2},t]})$ for $1 \leq j \leq s-2$; (ii) $(1-\alpha) f(x_{[t_j,t]}) \leq f(x_{[t_{j+1},t]})$ for $1 \leq j \leq s-1$; and (iii) $x_s = x_t$ and the start of the window is sandwiched between t_1 and t_2. The output at time t is $\ell_1^* = f(x_{[t_1,t]})$. Now once we have this invariance, we have the following derivation: Let $t' \leq t$.Then we know from point (iii) above, $f_t(x_{[t_2,t]}) \leq f(x_{[t-w+1,t]}) \leq f_t(x_{[t_1,t]}) = \ell_1^*$. Now the value of function on successive event points are close in the past, that is, $f_{t'}(x_{[t_{i+1},t']}) \geq (1-\beta) f_{t'}(x_{[t_i,t']})$. This implies that $f_{t'}(x_{[t_2,t']}) \geq (1-\beta) f_{t'}(x_{[t_1,t']})$ and $f_t(x_{[t_2,t]}) \geq (1-\alpha) f_t(x_{[t_1,t]}) \geq (1-\alpha) f(x_{[t-w+1,t]})$. This completes the proof of the lemma. □

Using Lemma 4, we can show the following.

Lemma 5. *Let $(x_t)_{t \geq 1}$ be the streamed vector. For a window of size w, let $x = x_{[t-w+1,t]} := \mathscr{U}(x_{t-w+1}, \dots, x_t)$. Then Algorithm 3 is (ε, δ)-differentially private algorithm in the sliding window model can be used to compute $\mathfrak{P}(\mathsf{L}_1, \mathscr{U}, i, j)$ for any $t - w + 1 \leq i \leq j \leq t$ with additive error at most $O(\frac{\log(\beta/w)}{\varepsilon})$.*

Proof of Lemma 5. Let i, j be indices such that $t - w + 1 \leq i \leq j \leq t$. Find the index a and b such that $t_a \leq i < t_{a+1}$ and $t_{b-1} < j + 1 \leq t_b$. Compute $f^* := \left\| \widehat{x}_{[a,t]} \right\|_1 - \left\| \widehat{x}_{[b,t]} \right\|_1$. Lemma 4 implies that

$$\left\| x_{i,t} \right\|_1 \leq \left\| x_{[a,t]} \right\|_1, \quad \left\| x_{[b,t]} \right\|_1 \leq \left\| x_{j+1,t} \right\|_1.$$

Using the concentration of Gaussian distribution, and $\left\| x_{[i,j]} \right\|_1 = \left\| x_{i,t} \right\|_1 - \left\| x_{[j,t]} \right\|_1$, we have

$$f^* \leq \left\| x_{[i,j]} \right\|_1 + 2\sigma$$

for every range $[i, j]$. This completes the proof. □

Proof of Theorem 6. The proof is a simple extension of the idea of lower bound proofs for heavy hitters in the unbounded streaming model to the sliding window model by carefully describing the stream. We set $p = \frac{1}{2\gamma(1+\zeta)} \log(\sqrt{w}) \log w$. Give its input $a \in \{0,1\}^p$, Alice divides it in to $\log w$ blocks, each of $\frac{1}{2\gamma(1+\zeta)} \log(\sqrt{w})$ bits. That is

$$x = (x^{(1)}, \cdots, x^{(\log w)}),$$

where $x^{(i)} \in \{0,1\}^{\log \sqrt{w}/(2\gamma(1+\zeta))}$. Let us denote every $x^{(i)}$ further as $\log \sqrt{w}$ bit value

$$x^{(i)} = (x^{(i,1)}, \dots, x^{(i,1/2\gamma(1+\zeta))}).$$

Alice and Bob construct the following heavy hitter instance. The universe is

$$\mathscr{X} = \left\{ (j, x^{(i,j)}) : i \in [\log w], j \in \left[\frac{1}{2\gamma(1+\zeta)} \right] \right\}.$$

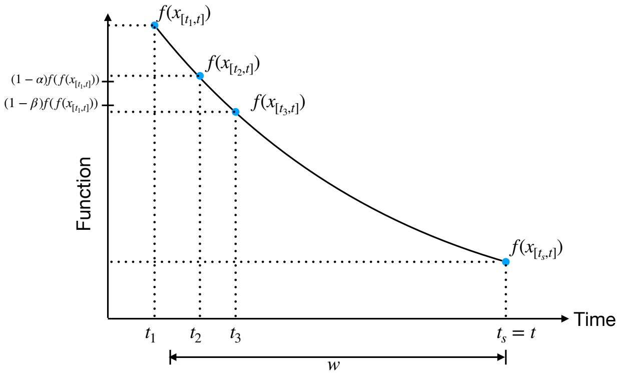

Figure 2. Smooth histogram data structure for window of size w.

Alice constructs the following stream, $(y_t)_{t\geq 1}$. The frequency of the value $(j, x^{(i,j)}) \in \mathcal{X}$ is set to $\nu^{(i,j)} = 0$ at the start of the stream. During the stream, the value of $\nu^{(i,j)}$ is updated just once by a value $\nu^{(i,j)} \leftarrow \nu^{(i,j)} + 2^{\log w - i}$. Alice runs its sliding window algorithm to produce a sketch of the stream and send its state to Bob. Bob expires all the updates before $x^{(i)}$. It then uses the output of the heavy hitter algorithm to compute the value of $x^{(i)}$.

Now let us see how this can be accomplished. The L_1 norm of the updates in the window considered by Bob is

$$\begin{aligned}\left\|y_{[t-w+1,w]}\right\|_1 &= \frac{1}{2\gamma(1+\zeta)}(2^0 + 2^1 + 2^2 + \ldots + 2^{\log w - i}) \\ &\leq \frac{2^{\log w - i}}{\gamma(1+\zeta)}.\end{aligned}$$

The manner in which the stream is updated, the frequency of the vector $x^{(i)}$ is

$$2^{\log w - i} \geq \gamma(1+\zeta)\frac{2^{\log w - i}}{\gamma(1+\zeta)} \geq \gamma(1+\zeta)\left\|y_{[t-w+1,w]}\right\|_1.$$

From the construction of the universe, these heavy hitters corresponds to elements, $e \in \mathcal{X}$ with first $\log(1/2\gamma(1+\zeta))$ bits corresponding to $j \in [1/2\gamma(1+\zeta)]$ and the remaining bits of e corresponding to $x^{(i,j)}$. Bob can thus concatanate the elements to construct $x^{(i)}$ as required. Using the lower bound of Miltersen et al. (1995) and the fact that any differentially private algorithm has to be randomized completes the proof of Theorem 6. □

6.1. Improved Accuracy Over the Entire Window

In the previous section, we gave an algorithm to compute heavy hitters with additive error $\tau_2 = O(\log(1/\beta)/\varepsilon)$ and over the entire window $w\tau_2$ (by setting $r = w$ in Corollary 1). In this section, we show an algorithm that computes heavy hitter with small additive error over the entire window, i.e., $o(w\tau_2)$.

To get an intuition of our new algorithm, we first analyze Algorithm 1 in a slightly different manner. For this, it is helpful to revisit the concept of partial sums defined by Chan et al. (2011). In partial sum, the goal is to evaluate a function on the values streamed in a particular time interval.

Definition 6 (Partial sum (Chan et al., 2011))**.** *A* partial sum*, denoted by $\mathfrak{P}(t_1, t_2)$-sum, is the sum of consecutive items streamed between time t_1 and t_2. That is, for $1 \leq t_1 \leq t_2$ and stream $(x_t)_{t\geq 1}$, the partial sum of $x_{[t_1,t_2]}$ is $\sum_{k=t_1}^{t_2} x_k$.*

We extend this definition with respect to any given input function, f, and update function, $\mathcal{U}$.

Definition 7 (Partial evaluation)**.** *A $\mathfrak{P}(f, \mathcal{U}, i, j)$-function corresponding to a function f and update function $\mathcal{U}$ is a partial evaluation of the function on updates caused by consecutive items. Let $1 \leq i \leq j \leq t$ and $(x_t)_{t\geq 1}$ be the stream, then the partial sum for substream $x_{[i,j]}$ is denoted by $\mathfrak{P}(f, \mathcal{U}, i, j) := f(\mathcal{U}(x_i, \ldots, x_j))$.*

Now we can cast Algorithm 1 in terms of partial sums. Every single element in the window can appear in w release of the output. Therefore, the sensitivity of the output is w, and to preserve privacy, each partial evaluation must be perturbed with noise $\mathsf{Lap}(w/\varepsilon)$. This leads to the accuracy bound in Corollary 1. Now consider an alternate algorithm that uses w space: we first privatize and store every entry in the stream and then use it to compute the heavy hitters. In this case, we need w partial evaluation with every element having sensitivity 1. Using the concentration result of Laplace distribution, the additive error incurred would be $O(\frac{\sqrt{w}\log(w/\beta)}{\varepsilon})$.

More precisely, if an algorithm uses p partial evaluations, such that every element in the stream can appear in at most s number of times in any of these sums, then it is easy to see that we can bound the total additive error to be

$$\tau = O\left(\frac{\sqrt{p}s\log(p/\beta)}{\varepsilon}\right).$$

The idea in Algorithm 2 to reduce the additive error by reducing s in the above expression at the expense of increasing p. The increase in p would result in an increase in the space required by the algorithm, thereby a tradeoff between the space and accuracy.

For the ease of presentation, we divide the window in to equal window of size $\sqrt{w}$ and run the first three steps of Algorithm 1. 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. Let us call this algorithm PRIVATE-HEAVY-HITTER. Using the analysis as in Section 3.1 and the fact that $s = p = \sqrt{w}$, we arrive at Theorem 5.

7. Related Works

Exponential Histogram. Given a function $f : \mathscr{X} \to \mathscr{R}$ to approximate in the sliding window model and update function $U(x_{[i,j]}) := \sum_{\ell=i}^{j} x_\ell$, the exponential histogram partitions the data stream into "buckets" of exponentially increasing size. Each bucket corresponds to a time interval. For example, suppose we are given a data stream of integers and we want to approximate the number of ones in the sliding window within a factor of 2. In the exponential histogram data structure, the smallest bucket consists of all elements in the data stream from the most recent element to the most recent element whose value is one. The next bucket would consist of all previous elements until two elements whose value is one are seen. Similarly, the i-th bucket consists of the previous elements until 2^i instances of ones are seen. They showed the following:

Theorem 9 (Datar et al. (2002))**.** *Given a stream with $\mathscr{X} = \{0, 1\}$, $\mathscr{Y} = \{0, 1\}^w$, there is an algorithm that uses $O(\frac{1}{\alpha} \log^2 w)$ bits of memory and provides an estimate of counts of 1 at every instant that is within a $(1 + \alpha)$ factor of the actual answer.*

The above theorem can be extended to any subadditive functions as well, i.e., function with the following property: evaluating the function for the sum of two elements of the domain is always less than or equal to an absolute constant factor the sum of the function's values at each element.

There are two variants of private heavy hitters studied in the literature. The first variant of the private heavy hitters problem has been considered in the setting of *pan-private streaming algorithms* (Dwork et al., 2010b; Mir et al., 2011), wherein the authors consider a setting in which a stream of elements is presented to the algorithm, and the algorithm must estimate the approximate count of heavy hitters. In this setting, the input domain is the set of individuals appearing in the stream, and so it is not possible to reveal the identity of the heavy hitter. On the other hand, we differentiate the universe from the individual, i.e., individual picks from the universe.

Recently, there has been a surge of interest in computing heavy hitters in the more restricted setting of *local model of privacy* (Bassily & Smith, 2015; Bassily et al., 2017; Bun et al., 2018; Erlingsson et al., 2014; Hsu et al., 2012) with n users. In this setting, the input domain and the set of users are separate, so one can output the estimate of heavy hitters (and not just the count of the heavy hitters). There are known lower and upper bounds of $\Theta(\sqrt{\frac{1}{\varepsilon^2 n}})$ in the local setting (Bassily & Smith, 2015; Bun et al., 2018).