CR-PRECIS: A deterministic summary structure for update data streams

Sumit Ganguly[1] and Anirban Majumder[2⋆]

[1] Indian Institute of Technology, Kanpur
[2] Lucent Technologies, Bangalore

Abstract. We present deterministic sub-linear space algorithms for a number of problems over update data streams, including, estimating frequencies of items and ranges, finding approximate frequent items and approximate ϕ-quantiles, estimating inner-products, constructing near-optimal B-bucket histograms and estimating entropy. We also present new lower bound results for several problems over update data streams.

1 Introduction

The data streaming model [2, 26] presents a computational model for a variety of monitoring applications, for example, network monitoring, sensor networks, etc., where data arrives rapidly and continuously and has to be processed in an online fashion using sub-linear space. Some examples of fundamental data streaming primitives include, estimating the frequency of items (point queries) and ranges (range-sum queries), finding approximate frequent items, approximate quantiles and approximate hierarchical heavy hitters, estimating inner-product, constructing approximately optimal B-bucket histograms, estimating entropy, etc.. We view a data stream as a sequence of arrivals of the form (i, v), where, i is the identity of an item belonging to the domain $\mathcal{D} = \{0, 1, \ldots, n-1\}$ and v is a non-zero integer that depicts the change in the frequency of i. $v \geq 1$ signifies v insertions of the item i and $v \leq -1$ signifies $|v|$ deletions of i. The frequency of an item i is denoted by f_i and is defined as the sum of the changes to its frequency since the inception of the stream, that is, $f_i = \sum_{(i,v) \text{ appears in stream}} v$. If $f_i \geq 0$ for all i (i.e., deletions correspond to prior insertions) then the corresponding streaming model is referred to as the *strict update* streaming model, whereas, the model in which frequencies can take arbitrary positive, zero or negative values is called the *general update* streaming model. The *insert-only* model refers to data streams with no deletions, that is, $v > 0$.

Randomized algorithms dominate the landscape of sub-linear space algorithms for problems over update streams. There are no deterministic sub-linear space algorithms known for a variety of basic problems over update streams, including, estimating the frequency of items and ranges, finding approximate frequent items and approximate ϕ-quantiles, finding approximate hierarchical

⋆ Work done while at IIT Kanpur.

heavy hitters, constructing approximately optimal B-bucket histograms, estimating inner-products, estimating entropy, etc.. Deterministic algorithms are often indispensable, for example, in a marketing scenario where frequent items correspond to subsidized customers, a false negative would correspond to a missed frequent customer, and conversely, in a scenario where frequent items correspond to punishable misuse [21], a false positive results in an innocent victim.

We now review a data structure introduced by Gasieniec and Muthukrishnan [26] (page 31) that we use later. We refer to this structure as the CR-PRECIS structure, since the Chinese Remainder theorem plays a crucial role in the analysis. The structure is parameterized by a height k and width t. Choose t consecutive prime numbers $k \leq p_1 < p_2 < \ldots < p_t$ and keep a collection of t tables T_j, for $j = 1, \ldots, t$, where, T_j has p_j integer counters, numbered from $0, 1, \ldots, p_j - 1$. Each stream update of the form (i, v) is processed as follows.

$$\textbf{for } j := 1 \textbf{ to } t \textbf{ do } \{ \; T_j[i \mod p_j] := T_j[i \mod p_j] + v \; \}$$

Lemma 1 presents the space requirement of a CR-PRECIS structure and is implicit in [26] (pp. 31) and is proved by a direct application of the prime number theorem [27]. Let $L_1 = \sum_i |f_i|$.

Lemma 1. *The space requirement of a* CR-PRECIS *structure with height parameter* $k \geq 12$ *and width parameter* $t \geq 1$ *is* $O(t(t + \frac{k}{\ln k}) \log(t + \frac{k}{\ln k})(\log L_1))$ *bits. The time required to process a stream update is* $O(t)$ *arithmetic operations.* □

[26] (pp. 31) uses the CR-PRECIS structure to present a k-set structure [15] using space $O(k^2(\log k)(\log L_1))$ bits. k-set structures using space $O(k(\log N)(\log N + \log L_1))$ bits are presented in [15].

Contributions. We present deterministic, sub-linear space algorithms for each of the problems mentioned above in the model of update streams using the CR-PRECIS structure. We also present improved space lower bounds for some problems over update streams, namely, (a) the problem of estimating frequencies with accuracy ϕ over strict update streams is shown to require $\Omega(\phi^{-1}(\log m) \log(\phi n))$ space (previous bound was $\Omega(\phi^{-1} \log(\phi n))$ [4]), and, (b) all the above problems except for the problems of estimating frequencies of items and range-sums, are shown to require $\Omega(n)$ space in the general update streaming model.

2 Review

In this section, we review basic problems over data streams. For strict update streams, let $m = \sum_i f_i$ and for general update streams, $L_1 = \sum_i |f_i|$.

The *point query* problem with parameter ϕ is the following: given $i \in \mathcal{D}$, obtain an estimate $\hat{f}_i$ such that $|\hat{f}_i - f_i| \leq \phi L_1$. For *insert-only streams*, the Misra-Gries algorithm [25], rediscovered and refined in [12, 4, 22], uses $\lceil \phi^{-1} \rceil \log m$ bits and satisfies $f_i \leq \hat{f}_i \leq f_i + \phi m$. The *Lossy Counting* algorithm [23] for insert-only streams returns $\hat{f}_i$ satisfying $f_i \leq \hat{f}_i \leq f_i + \phi m$ using $\lceil \phi^{-1} \rceil \log(\phi m) \log m$ bits. The *Sticky Sampling* algorithm [23] extends the *Counting Samples* algorithm [16] and returns $\hat{f}_i$ satisfying $f_i - \phi m \leq \hat{f}_i \leq f_i$ with probability $1 - \delta$

using space $O(\lceil \phi^{-1} \rceil \log(\delta^{-1}) \log m)$ bits. The COUNT-MIN sketch algorithm returns $\hat{f}_i$ that satisfies (a) $f_i \leq \hat{f}_i \leq f_i + \phi m$ with probability $1 - \delta$ for strict update streams, and, (b) $|\hat{f}_i - f_i| \leq \phi L_1$ using $O(\lceil \phi^{-1} \rceil (\log \delta^{-1}) \log L_1)$ bits. The COUNTSKETCH algorithm [6] satisfies $|\hat{f}_i - f_i| \leq (\lceil \phi^{-1} \rceil F_2^{res}(\lceil \phi^{-1} \rceil))^{1/2} \leq \phi L_1$ with probability $1 - \delta$ using space $O(\lceil \phi^{-1} \rceil (\log \delta^{-1} \log L_1)$ bits, where, $F_2^{res}(s)$ is the sum of the squares of all but the top-s frequencies in the stream. [4] shows that any algorithm satisfying $|\hat{f}_i - f_i| \leq \phi m$ must use $\Omega(\lceil \phi^{-1} \rceil \log \phi n)$ bits.

Given a parameter $0 < \phi < 1$, an item i is said to be ϕ-*frequent* if $|f_i| \geq \phi L_1$. [11, 22] show that finding all and only frequent items requires $\Omega(n)$ space. Therefore low-space algorithms find ϵ-approximate frequent items, where, $0 < \epsilon < 1$ is another parameter: return i such that $|f_i| \geq \phi L_1$ and do not return any i such that $|f_i| < (1-\epsilon)\phi L_1$ [6, 11, 9, 12, 16, 22, 25, 23, 28]. Algorithms for finding frequent items with parameters ϕ and ϵ typically use point query estimators with parameter $\frac{\epsilon\phi}{2}$ and return all items i such that $\hat{f}_i > (1 - \frac{\epsilon}{2})\phi L_1$. A superset of ϵ-approximate frequent items is typically found using the technique of dyadic intervals [10, 9], reviewed below.

A *dyadic interval* at level l is an interval of size 2^l from the family of intervals $\{[i2^l, (i+1)2^l - 1], 0 \leq i \leq \frac{n}{2^l} - 1\}$, for $0 \leq l \leq \log n$, assuming that n is a power of 2. The set of dyadic intervals at levels 0 through $\log n$ form a complete binary tree, whose root is the level $\log n$ dyadic interval $[0, n-1]$ and whose leaves are the singleton intervals. Each dyadic interval I_l with level $1 \leq l \leq \log n$ has two children that are dyadic intervals at levels $l - 1$. If $I_l = [i2^l, (i+1)2^l - 1]$, for $0 \leq i \leq \frac{n}{2^l}$, then, the left child of I_l is $[2i\frac{n}{2^{l+1}}, (2i+1)\frac{n}{2^{l+1}} - 1]$ and the right child is $[(2i+1)\frac{n}{2^{l+1}}, (2i+2)\frac{n}{2^{l+1}} - 1]$. Given a stream, one can naturally extend the notion of item frequencies to dyadic interval frequencies. The frequency of a dyadic interval I_l at level l is the aggregate of the frequencies of the level 0 items that lie in that interval , that is, $f_{I_l} = \sum_{x \in I_l} f_x$. The efficient solution to a number of problems over *strict update* streams, including the problem of finding approximate frequent items, is facilitated by using summary structures for each dyadic level $l = 0, 1, \ldots, \lceil \log \phi n \rceil$ [10]. For the problem of ϵ-approximate ϕ-frequent items, we keep a point query estimator structure corresponding to accuracy parameter $\epsilon\phi$ for each dyadic level $l = 0, \ldots, \lceil \log(\phi n) \rceil$. An arrival over the stream of the form (i, v) is processed as follows: for each $l = 0, 1, \ldots, \lfloor \log \phi n \rfloor$, propagate the update $((i \ \% \ 2^l), v)$ to the structure at level l. Since, each item i belongs to a unique dyadic interval at each level l, the sum of the interval frequencies at level l is m. If an item i is frequent (i.e., $f_i \geq \phi m$), then for each $1 \leq l \leq \log n$, the unique dyadic interval I_l that contains i at level l has frequency at least f_i and is therefore also frequent at level l. To find ϵ-approximate ϕ-frequent items, we start by enumerating $O(\lceil \phi^{-1} \rceil)$ dyadic intervals at level $\lfloor \log \phi n \rfloor$. Only those candidate intervals are considered whose estimated frequency is at least $(1 - \frac{\epsilon}{2})\phi n$. We then consider the left and the right child of these candidate intervals, and repeat the procedure. In general, at level l, there are $O(\lceil \phi^{-1} \rceil)$ candidate intervals, and thus, the total number of intervals considered in the iterations is $O(\lceil \phi^{-1} \rceil \log(\phi n))$.

The *hierarchical heavy hitters* problem [8, 13] is a useful generalization of the frequent items problem for domains that have a natural hierarchy (e.g., domain of IP addresses). Given a hierarchy, the frequency of a node X is defined as the sum of the frequencies of the leaf nodes (i.e., items) in the sub-tree rooted at X. The definition of hierarchical heavy hitter node (*HHH*) is inductive: a leaf node x is an *HHH* node provided $f_x > \phi m$. An internal node is an *HHH* node provided that its frequency, after discounting the frequency of all its descendant *HHH* nodes, is at least ϕm. The problem is, (a) to find all nodes that are *HHH* nodes, and, (b) to not output any node whose frequency, after discounting the frequencies of descendant *HHH* nodes, is below $(1-\epsilon)\phi m$. This problem has been studied in [8, 10, 13, 21]. As shown in [8], this problem can be solved by using a simple bottom-up traversal of the hierarchy, identifying the frequent items at each level, and then subtracting the estimates of the frequent items at a level from the estimated frequency of its parent [8]. Using COUNT-MIN sketch, the space complexity is $O((\epsilon\phi^2)^{-1}(\log((\delta\epsilon\phi^2)^{-1}\log n))(\log n)(\log m))$ bits. [21] presents an $\Omega(\phi^{-2})$ space lower bound for this problem, for fixed $\epsilon \leq 0.01$.

Given a *range* $[l, r]$ from the domain $\mathcal{D}$, the range frequency is defined as $f_{[l,r]} = \sum_{x=l}^{r} f_x$. The *range-sum query* problem with parameter ϕ is: return an estimate $\hat{f}_{[l,r]}$ such that $|\hat{f}_{[l,r]} - f_{[l,r]}| \leq \phi m$. The range-sum query problem can be solved by using the technique of dyadic intervals [19]. Any range can be uniquely decomposed into the disjoint union of at most $2\log n$ dyadic intervals of maximum size (for e.g., over the domain $\{0, \ldots, 15\}$, the interval $[3, 12] = [3, 3] + [4, 7] + [8, 11] + [12, 12]$). The technique is to keep a point query estimator corresponding to each dyadic level $l = 0, 1, \ldots, \log n - 1$. The range-sum query is estimated as the sum of the estimates of the frequencies of each of the constituent maximal dyadic intervals of the given range. Using COUNT-MIN sketch at each level, this can be accomplished using space $O(\phi^{-1}\log(\log(\delta^{-1}n))(\log n)(\log m))$ bits and with probability $1-\delta$ [10].

Given $0 \leq \phi \leq 1$ and $j = 1, 2, \ldots, \lceil \phi^{-1} \rceil$, an *$\epsilon$-approximate j^{th} ϕ-quantile* is an item a_j such that $(j\phi - \epsilon)m \leq \sum_{i=a_j}^{n-1} f_i \leq (j\phi + \epsilon)m$. The problem has been studied in [10, 20, 18, 24]. For insert-only streams, [20] presents an algorithm requiring space $O((\log(\epsilon\phi)^{-1})\log(\epsilon\phi m))$. For strict update streams, the problem of finding approximate quantiles can be reduced to that of estimating range sums [18] as follows. For each $k = 1, 2, \ldots, \phi^{-1}$, a binary search is performed over the domain to find an item a_k such that the estimated range sum $\hat{f}_{[a_k, n-1]}$ lies between $(k\phi - \epsilon)m$ and $(k\phi + \epsilon)m$. [10] uses COUNT-MIN sketches and the above technique to find ϵ-approximate ϕ-quantiles with confidence $1-\delta$ using space $O(\epsilon\phi^{-1}\log^2 n((\log(\epsilon\phi\delta)^{-1}) + \log\log n))$.

A B-bucket histogram h divides the domain $\mathcal{D} = \{0, 1, \ldots, n-1\}$ into B non-overlapping intervals, say, $I_1, I_2, \ldots, I_B$ and for each interval I_j, chooses a value v_j. Then $h[0 \ldots n-1]$ is the vector defined as $h_i = v_j$, where, I_j is the unique interval containing i. The cost of a B-bucket histogram h with respect to the frequency vector f is defined as $||f - h||$. Let h^{opt} denote an optimal B-bucket histogram satisfying $||f - h^{opt}|| = \min_{B\text{-bucket histogram } h} ||f - h||$. The problem is to find a B-bucket histogram $\hat{h}$ such that $||f - \hat{h}|| \leq (1+\epsilon)||f - h^{opt}||$.

An algorithm for this problem is presented in a seminal paper [17] using space and time poly $(B, \frac{1}{\epsilon}, \log m, \log n)$ (w.r.t. L_2 distance $\|f-h\|_2$).

Given two streams R and S with item frequency vectors f and g respectively, the *inner product* $f \cdot g$ is defined as $\sum_{i \in \mathcal{D}} f_i \cdot g_i$. The problem is to return an estimate $\hat{P}$ satisfying $|\hat{P} - f \cdot g| \leq \phi m_R m_S$. The problem finds applications in database query processing. The work in [1] presents a space lower bound of $s = \Omega(\phi^{-1})$ for this problem. Randomized algorithms [1, 7, 14] match the space lower bound, up to poly-logarithmic factors. The *entropy* of a data stream is defined as $H = \sum_{i \in \mathcal{D}} |f_i| \log \frac{L_1}{|f_i|}$. The problem is to return an ϵ-approximate estimate $\hat{H}$ satisfying $|\hat{H} - H| \leq \epsilon H$. For insert-only streams, [5] presents a randomized estimator that uses space $O(\epsilon^{-2}(\log \delta^{-1}) \log^3 m)$ bits and also shows an $\Omega(\epsilon^{-2}(\log(\epsilon^{-1}))^{-1})$ space lower bound for estimating entropy. [3] presents a randomized estimator for update streams using space $O(\epsilon^{-3} \log^5 m(\log \epsilon^{-1})(\log \delta^{-1}))$.

We note that sub-linear space deterministic algorithms over update streams are not known for any of the above-mentioned problems.

3 CR-PRECIS structure for update streams

In this section, we use the CR-PRECIS structure to present algorithms for a family of problems over update streams.

An application of the Chinese Remainder Theorem. Consider a CR-PRECIS structure with height k and width t. Fix $x, y \in \{0, \ldots, n-1\}$ where $x \neq y$. Suppose x and y collide in the tables indexed by J, where, $J \subset \{1, 2, \ldots, t\}$. Then, $x \equiv y \mod p_j$, for each $j \in J$. By the Chinese Remainder theorem, $x \equiv y \mod \left(\prod_{j \in J} p_j\right)$. Therefore, $|J| < \log_k n$, otherwise, $\prod_{j \in J} p_j \geq k^{\log_k n} = n$, which is a contradiction, since, $x, y \in \{0, 1, \ldots, n-1\}$ and are distinct. Therefore, for any given $x, y \in \{0, 1, \ldots, n-1\}$ such that $x \neq y$,

$$|\{j \mid y \equiv x \mod p_j \text{ and } 1 \leq j \leq t\}| \leq \log_k n - 1 \ . \tag{1}$$

3.1 Algorithms for strict update streams

In this section, we use the CR-PRECIS structure to design algorithms over strict update streams.

Point Queries. Consider a CR-PRECIS structure with height k and width t. The frequency of $x \in \mathcal{D}$ is estimated as: $\hat{f}_x = \min_{j=1}^{t} T_j[x \mod p_j]$. The accuracy guarantees are given by Lemma 2.

Lemma 2. *For* $0 \leq x \leq n-1$, $0 \leq \hat{f}_x - f_x \leq \frac{(\log_k n - 1)}{t}(m - f_x)$.

Proof. Clearly, $T_j[x \bmod p_j] \geq f_x$. Therefore, $\hat{f}_x \geq f_x$. Further,

$$t\hat{f}_x \leq \sum_{j=1}^{t} T_j\left[x \mod p_j\right] \quad = \quad tf_x \; + \; \sum_{j=1}^{t} \quad \sum_{\substack{y \neq x \\ y \equiv x \mod p_j}} f_y \; .$$

$$\text{Thus,} \quad t(\hat{f}_x - f_x) = \sum_{j=1}^{t} \quad \sum_{\substack{y \neq x \\ y \equiv x \bmod p_j}} f_y \quad = \quad \sum_{y \neq x} \quad \sum_{j : y \equiv x \bmod p_j} f_y$$

$$= \sum_{y \neq x} f_y |\{j : y \equiv x \bmod p_j\}| \quad \leq (\log_k n - 1)(m - f_x), \quad \text{by} \quad (1) \quad . \qquad \Box$$

If we let $k = \lceil \phi^{-1} \rceil$ and $t = \lceil \phi^{-1} \rceil \log_{\lceil \phi^{-1} \rceil} n$, then, the space requirement of the point query estimator is $O(\phi^{-2}(\log_{\lceil \phi^{-1} \rceil} n)^2 (\log m))$ bits. A slightly improved guarantee that is often useful for the point query estimator is given by Lemma 3, where, $m^{res}(s)$ is the sum of all but the top-s frequencies [3, 6].

Lemma 3. *Consider a* CR-PRECIS *structure with height* s *and width* $2s \log_s n$. *Then, for any* $0 \leq x \leq n-1$, $0 \leq \hat{f}_x \leq \frac{m^{res}(s)}{s}$.

Proof. Let $y_1, y_2, \ldots, y_s$ denote the items with the top-s frequencies in the stream (with ties broken arbitrarily). By (1), x conflicts with each $y_j \neq x$ in at most $\log_s n$ buckets. Hence, the total number of buckets at which x conflicts with any of the top-s frequent items is at most $s \log_s n$. Thus there are at least $t - s \log_s n$ tables where, x does not conflict with any of the top-s frequencies. Applying the proof of Lemma 2 to only these set of $t - s \log_s n \geq s \log_s n$ tables, the role of m is replaced by $m^{res}(s)$. $\Box$

We obtain deterministic algorithms for estimating range-sums, finding approximate frequent items, finding approximate hierarchical heavy hitters and ϵ-approximate quantiles over strict update streams, by using the corresponding well-known reductions to point query estimators. The only change is that the use of randomized summary structures is replaced by a CR-PRECIS structure. Theorem 4 summarizes the space versus accuracy guarantees for these problems.

Theorem 4. *There exist deterministic algorithms over the strict update streaming model for the problems mentioned in Figure 1 using the space and per-update processing time depicted there.* $\Box$

Estimating inner product. Let $m_R = \sum_{i \in \mathcal{D}} f_i$ and let $m_S = \sum_{i \in \mathcal{D}} g_i$. We maintain a CR-PRECIS structure for each of the streams R and S, that have the same height k, same width t and use the same prime numbers as the table sizes. For $j = 1, 2, \ldots, t$, let T_j and U_j respectively denote the tables maintained for streams R and S corresponding to the prime p_j respectively. The estimate $\hat{P}$ for the inner product is calculated as $\hat{P} = \min_{j=1}^{t} \quad \sum_{b=1}^{p_j} T_j[b] U_j[b]$.

Problem	Space	Time
1. ϵ-approx. ϕ-frequent items ($\lambda = \lceil(\epsilon\phi)^{-1}\rceil$)	$O(\lambda^2(\log_\lambda n)(\log\lambda)$ $\log(\lambda^{-1}n)\ (\log m))$	$O(\lambda\log_\lambda n\log n)$
2. Range-sum: parameter ϕ, ($\rho = \lceil\phi^{-1}\rceil$)	$O(\rho^2(\log_\rho n)\ (\log\rho$ $+\log\log_\rho n))(\log m)\log n)$	$O(\rho(\log_\rho n)\ (\log n))$
3. ϵ-approx. ϕ-quantile ($\lambda = \lceil(\epsilon\phi)^{-1}\rceil$)	$O(\lambda^2(\log^5 n)(\log m)$ $(\log\log n + \log\lambda)^{-1})$	$O\big(\lambda(\log^2 n)$ $(\log\log n + \log\lambda)^{-1}\big)$
4. ϵ-approx. ϕ-hierarchical heavy hitters. h = height, ($\tau = (\epsilon\phi^2)^{-1}h$)	$O(\tau^2(\log_\tau n)^2$ $(\log\tau + \log\log n)\log m)$	$O(\tau(\log n)(\log_\tau n))$
5. ϵ-approx. B-bucket histogram	$O\big((\epsilon^{-2}B^2)(\log^3 n)$ $(\log^{-1}(\epsilon^{-1}B))(\log m)$	$O((\epsilon^{-1}B)(\log^2 n)$ $(\log^{-1}(\epsilon^{-1}B))$

Fig. 1. Space and time requirement for problems using CR-PRECIS technique.

Lemma 5. $f\cdot g \le \hat{P} \le f\cdot g + \left(\frac{\log_k n - 1}{t}\right) m_R m_S$.

Proof. For $j = 1,\ldots,t$, $\sum_{b=0}^{p_j-1} T_j[b]U_j[b] \ge \sum_{b=0}^{p_j-1}\sum_{x\equiv b \bmod p_j} f_x g_x = f\cdot g$. Thus, $\hat{P} \ge f\cdot g$. Further,

$$t\hat{P} \le \sum_{j=1}^{t}\sum_{b=1}^{p_j} T_j[b]U_j[b] = t(f\cdot g) + \sum_{j=1}^{t}\sum_{\substack{x\neq y\\ x\equiv y \bmod p_j}} f_x g_y$$
$$= t(f\cdot g) + \sum_{x,y:x\neq y} f_x g_y \sum_{j:x\equiv y \bmod p_j} 1$$
$$\le t(f\cdot g) + (\log_k n - 1)(m_R m_S - f\cdot g),\ \text{by}\ \ (1). \qquad \square$$

Estimating entropy. We use the CR-PRECIS structure to estimate the entropy H over a strict update stream. For parameters k and t to be fixed later, a CR-PRECIS structure of height $k \ge 2$ and width t is maintained. Also, let $0 < \epsilon < 1$ be a parameter. First, we use the point query estimator to find all items x such that $\hat{f}_x \ge \frac{m}{\epsilon t}$. The contribution of these items, called *dense* items, to the estimated entropy is given by $\hat{H}_d = \sum_{x:\hat{f}_x > \frac{m}{\epsilon t}} \hat{f}_x \log\frac{m}{\hat{f}_x}$. Next, we remove the estimated contribution of the dense items from the tables. To ensure that the residues of dense frequencies remain non-negative, the estimated frequency is altered. Since, $0 \le \hat{f}_x - f_x \le \frac{(m-f_x)}{t}$, $f_x \ge \hat{f}_x - \frac{m-\hat{f}_x}{t-1} = f'_x$ (say), the tables are modified as follows: $T_j[x \mod p_j]$ **:=** $T_j[x \mod p_j] - f'_x$, for each x s.t. $\hat{f}_x \ge \frac{m}{\epsilon t}$ and $j = 1,\ldots,t$. $\hat{H}_s$ estimates the contribution to H by the non-dense or *sparse* items: $\hat{H}_s = \text{avg}_{j=1}^{t}\sum_{1\le b\le p_j \text{ and } T_j[b]\le\frac{m}{\epsilon^2 t}} T_j[b]\log\frac{m}{T_j[b]}$. The final estimate is returned as $\hat{H} = \hat{H}_d + \hat{H}_s$.

Analysis. The main part of the analysis concerns the accuracy of the estimation of H_s. Let H_d and H_s denote the (true) contributions to entropy due to dense and sparse items respectively, that is, $H_d = \sum_{x \text{ dense}} f_x \log\frac{m}{f_x}$ and

$H_s = \sum_{x \text{ sparse}} f_x \log \frac{m}{f_x}$. A standard case analysis [3, 5] (Case 1: $f_x \le m/e$ and Case 2: $f_x > m/e$, where, $e = 2.71828..$) is used to show that $|\hat{H}_d - H_d| \le \frac{2}{t} H_d$. Define $g_x = f_x$, if x is a sparse item and $g_x = f_x - f'_x$, if x is a dense item. Let $m' = \sum_x g_x$ and let $H(g) = \sum_{x: f_x > 0} g_x \log \frac{m}{g_x}$. Suppose x maps to a bucket b in table T_j. Then, $f_x \le T_j[b]$ and

$$\sum_{b=1}^{p_j} T_j[b] \log \frac{m}{T_j[b]} \le \sum_{b=1}^{p_j} \sum_{x \bmod p_j = b} g_x \log \frac{m}{T_j[b]} = H(g) \ .$$

Thus, $\hat{H}_s \le H(g)$. Since, $H(g)$ may contain the contribution of the residues of the dense items, $H_s \le H(g)$. The contribution of the residues of dense items to $H(g)$ is bounded as follows. Let x be a dense item. Define $h(a) = y \log \frac{m}{y}$. For $t > \frac{1}{\epsilon^2}$, and since, $f_x \ge \frac{m}{\epsilon t}$, $\quad h(\frac{g_x}{m}) \le h\left(\frac{m - f_x}{mt}\right) \le \epsilon h(f_x)$. Since, Therefore,

$$H(g) = \sum_x g(x) \log \frac{m}{g_x} = H_s + \sum_{x \text{ dense}} m h(g_x) \le H_s + \sum_{x \text{ dense}} m \epsilon h(f_x) = H_s + \epsilon H_d \ .$$

Further, for $0 \le x \le n-1$, let $S_x = \sum_{j=1}^{t} (T_j[x \mod p_j] - g_x)$. Then

$$S_x = \sum_{j=1}^{t} \sum_{\substack{y \ne x \\ y \equiv x \bmod p_j}} g_y = \sum_{y \ne x} f_y |\{j : y \equiv x \bmod p_j\}| \le (m' - g_x)(\log n - 1) \quad (2)$$

Define a bucket b in a table T_j to be dense if $T_j[b] > \frac{m}{\epsilon^2 t}$ and $c = \epsilon^2 t$. Then,

$$\begin{aligned}
t\hat{H}_s &= \sum_{j=1}^{t} \sum_{\substack{1 \le b \le p_j \\ b \text{ not dense}}} T_j[b] \log \frac{m}{T_j[b]} \\
&\ge \sum_{j=1}^{t} \sum_{\substack{1 \le b \le p_j \\ b \text{ not dense}}} \sum_{x : x \equiv b \bmod p_j \text{ and } g_x \ge 1} g_x \log c \\
&= tm' \log c - \sum_x g_x \log c \ |\{j : T_j[x \mod p_j] \text{ is dense }\}| \\
&= tm' \log c - \sum_x g_x (\log c) \ \lfloor S_x / (c^{-1} m - g_x) \rfloor \\
&\ge tm' \log c - \sum_x g_x c (1-\epsilon)^{-1} (\log c)(\log_k N) \quad \text{by } \ (2) \text{ and since, } g_x \le \epsilon c^{-1} m \\
&\ge tm' \log c - m' c (1-\epsilon)^{-1} (\log c)(\log_k N) \\
&\ge tm' \log c \left(1 - \epsilon^2 (1-\epsilon)^{-1} \log_k N\right) \qquad (3)
\end{aligned}$$

Lemma 6. *For each $0 < \varepsilon < 1$ and $\alpha > 1$, there exists a deterministic algorithm over strict update streams that returns $\hat{H}$ satisfying $\frac{H(1-\varepsilon)}{\alpha} \le H \le (1 + \frac{\varepsilon}{\sqrt{\log N}}) H$ using space $O(\frac{\log^2 N}{\varepsilon^4} m^{\frac{2}{\alpha}} (\log m + \log \varepsilon^{-1})(\log m))$ bits.*

Proof. By earlier argument,

$$\hat{H}_d + \hat{H}_s \le (1 + 2t^{-1})H_d + H(g) \le (1 + 2t^{-1})H_d + \epsilon H_d + H_s \\ \le (1 + 2t^{-1} + \epsilon)(H_d + H_s) \ .$$

Further, since, $H(g) \le m' \log m$, using (3) we have,

$$\hat{H}_d + \hat{H}_s \ge (1 - 2t^{-1})H_d + H_s \frac{\log c}{\log m}(1 - \epsilon^2(1-\epsilon)^{-1} \log_k N) \ .$$

The lemma follows by letting $\epsilon = \frac{\varepsilon}{2\sqrt{\log N}}$ and $t = \frac{m^{1/\alpha}}{\epsilon^2}$. ◻

Lower bounds for computation over strict update streams. In this section, we improve on the existing space lower bound of $\Omega(\phi^{-1}\log(\phi n))$ for point query estimation with accuracy parameter ϕ [4].

Lemma 7. *For $\phi > \frac{8}{\sqrt{n}}$, a deterministic point query estimator with parameter ϕ over strict update streams requires $\Omega(\phi^{-1}(\log m)\log(\phi n))$ space.*

Proof. Let $s = \lceil \phi^{-1} \rceil$. Consider a stream consisting of s^2 distinct items, organized into s levels $1, \ldots, s$ with s items per level. The frequency of an item at level l is set to $t_l = 2^{l-1}$. Let $\phi' = \frac{\phi}{16}$ and let A be a deterministic point query estimator with accuracy parameter ϕ'. We will apply A to obtain the identities of the items, level by level. Initially, the stream is inserted into the data structure of A. At iteration $r = 1, \ldots, s$ in succession, we maintain the invariant that items in levels higher than $s - r + 1$ have been discovered and their exact frequencies are deleted from A. Let $l = s - r + 1$. At the beginning of iteration r, the total frequency is $m = m_l = \sum_{u=1}^{l}(st_u) \le s\sum_{u=1}^{l} 2^{l-1} < s2^l$. At iteration r, we use A to return the set of items x such that $\hat{f}_x \ge U_l = 2^{l-1} - 2^{l-4}$. Therefore, (a) estimated frequencies of items in level l cross the threshold U_l, since, $\hat{f}_x \ge f_x - \phi' m \ge 2^{l-1} - \frac{\phi 2^l s}{16} \ge U_l$, and, (b) estimated frequencies of items in level $l-1$ or lower do not cross U_l, since, $\hat{f}_y \le f_y + \phi' m \le 2^{l-2} + 2^{l-3} < U_l$. After s iterations, the level by level arrangement of the items can be reconstructed. The number of such arrangements is $\binom{n}{s\ s \ldots\ s}$ and therefore $\mathcal{A}$ requires space $\log \binom{n}{s\ s \ldots\ s} = \Omega(s^2 \log \frac{n}{s}) = \Omega(s(\log m)(\log \frac{n}{s}))$, since, $n > 64s^2$ and $m = m_s = s2^{s+1}$. ◻

Lemma 8. *For $\phi > 8n^{-1/2}$, any point query estimator for strict update streams with parameter ϕ and confidence* 0.66 *requires $\Omega(\phi^{-1}(\log m)(\log(\phi n)))$ bits.*

Proof. We reduce the bit-vector indexing problem to a randomized point query estimator. In the bit-vector indexing problem, bit vector v of size $|v|$ is presented followed by an index i between 1 and $|v|$. The problem is to decide whether $v[i] = 1$. It is known to require space $\Omega(|v|)$ by a randomized algorithm that gives the correct answer with probability $\frac{2}{3}$. Let $s = \lceil \phi^{-1} \rceil, |v| = s^2 \lceil \log(\lceil \frac{n}{s} \rceil) \rceil$ and $\rho = \lceil \log \lceil \frac{n}{s} \rceil \rceil$. We can isomorphically view the vector $v[1 \ldots |v|]$ as a set of contiguous segments τ of size ρ each, starting at positions 1 modulo ρ. The s^2 possible starting positions can be represented as (a_τ, b_τ), where, $a_\tau, b_\tau \in \{0, 1, \ldots, s-1\}$. Map each such segment to an item from the domain $s^2 2^\rho$ with $2\lceil \log s \rceil + \rho$ bit binary address $a_\tau \circ b_\tau \circ \tau$, where, $\circ$ represents bit concatenation. The frequency of the item is set to 2^{a_τ}. The bit vector is isomorphically mapped to a set of s^2 items of frequencies between 1 and 2^{s-1}, such that there are exactly s items with frequency 2^l, for $l = 0, 1, \ldots, s-1$. If the error probability of the point estimator is at most $1 - \frac{1}{3s^2}$, then, using the argument of Lemma 7, the set of all the s^2 items and their frequencies are correctly retrieved with error probability bounded by $\frac{s^2}{3s^2} = \frac{1}{3}$. That is, the bit vector is effectively reconstructed and the original bit vector index query can be answered. The above argument holds for every choice of the 1-1 onto mappings of the s^2 starting positions of ρ-size segments to (a_τ, b_τ). In particular, it holds for the specific map when the query index i belongs to a segment τ_0 whose starting position is mapped to the highest level, i.e., $b_{\tau_0} = s-1$. In this case, a single invocation of the point query suffices. Thus the space required is $\Omega(s^2 \log \frac{n}{s}) = \Omega(\phi^{-1}(\log m)(\log \phi n))$. □

3.2 General update streaming model

In this section, we consider the general update streaming model. Lemma 9 presents the property of point query estimator for general update streams.

Lemma 9. *Consider a* CR-PRECIS *structure with height k and width t. For $x \in \mathcal{D}$, let $\hat{f}_x = \frac{1}{t}\sum_{j=1}^{t} T_j[x \mod p_j]$. Then, $|\hat{f}_x - f_x| \le \frac{(\log_k n - 1)}{t}(L_1 - |f_x|)$.*

Proof. $t\hat{f}_x = \sum_{j=1}^{t} T_j\left[x \mod p_j\right] = tf_x + \sum_{j=1}^{t}\sum\{f_y \mid y \neq x \text{ and } y \equiv x \bmod p_j\}$.

$$\text{Thus, } t|\hat{f}_x - f_x| = |\sum_{j=1}^{t} \sum_{\substack{y \neq x \\ y \equiv x \bmod p_j}} f_y| = |\sum_{y \neq x} \sum_{j: y \equiv x \bmod p_j} f_y|$$

$$\le \sum_{y \neq x} \sum_{j: y \equiv x \bmod p_j} |f_y| \le (\log_k n - 1)\left(F_1 - |f_x|\right), \text{ by } (1) \quad \square$$

Similarly, we can obtain an estimator for the inner-product of streams R and S. Let $L_1(R)$ and $L_1(S)$ be the L_1 norms of streams R and S respectively.

Lemma 10. *Consider a* CR-PRECIS *structure of height k and width t. Let $\hat{P} = \frac{1}{t}\sum_{j=1}^{t}\sum_{b=1}^{p_j} T_j[b]U_j[b]$. Then, $|\hat{P} - f \cdot g| \le \frac{(\log_k n - 1)}{t} L_1(R)L_2(S)$.* □

Lower bounds for computations over general update streams. We now present space lower bounds for problems over general update streams.

Lemma 11. *Deterministic algorithms for the following problems in the general update streaming model requires $\Omega(n)$ bits: (1) finding ϵ-approximate frequent items with parameter s for any $\epsilon < \frac{1}{2}$, (2) finding ϵ-approximate ϕ-quantiles for any $\epsilon < \phi/2$, (3) estimating the k^{th} norm $L_k = (\sum_{i=0}^{n-1}|f_i|^k)^{1/k}$, for any real value of k, to within any multiplicative approximation factor, and (4) estimating entropy to within any multiplicative approximation factor.*

Proof. Consider a family $\mathcal{F}$ of sets of size $\frac{n}{2}$ elements each such that the intersection between any two sets of the family does not exceed $\frac{n}{8}$. It can be shown[3] that there exist such families of size $2^{\Omega(n)}$. Corresponding to each set S in the family, we construct a stream $str(S)$ such that $f_i = 1$ if $i \in S$ and $f_i = 0$, otherwise. Denote by $str_1 \circ str_2$ the stream where the updates of stream str_2 follow the updates of stream str_1 in sequence. Let $\mathcal{A}$ be a deterministic frequent items algorithm. Suppose that after processing two distinct sets S and T from $\mathcal{F}$, the same memory pattern of $\mathcal{A}$'s store results. Let Δ be a stream of deletions that deletes all but $\frac{s}{2}$ items from $str(S)$. Since, $L_1(str(S) \circ \Delta) = \frac{s}{2}$, all remaining $\frac{s}{2}$ items are found as frequent items. Further, $L_1(str(T) \circ \Delta) \geq \frac{n}{2} - \frac{s}{2}$, since, $|S \cap T| \leq \frac{n}{8}$. If $s < \frac{n}{3}$, then, $\frac{F_1}{s} > 1$, and therefore, none of the items qualify as frequent. Since, $str(S)$ and $str(T)$ are mapped to the same bit pattern, so are $str(S) \circ \Delta$ and $str(T) \circ \Delta$. Thus $\mathcal{A}$ makes an error in reporting frequent items in at least one of the two latter streams. Therefore, $\mathcal{A}$ must assign distinct bit patterns to each $str(S)$, for $S \in \mathcal{F}$. Since, $|\mathcal{F}| = 2^{\Omega(n)}$, $\mathcal{A}$ requires $\Omega(\log(|\mathcal{F}|)) = \Omega(n)$ bits, proving part (1) of the lemma.

Let S and T be sets from $\mathcal{F}$ such that $str(S)$ and $str(T)$ result in the same memory pattern of a quantile algorithm $\mathcal{Q}$. Let Δ be a stream that deletes all items from S and then adds item 0 with frequency $f_0 = 1$ to the stream. now all quantiles of $str(S) \circ \Delta = 0$. $str(T) \circ \Delta$ has at least $\frac{7n}{8}$ distinct items, each with frequency 1. Thus, for every $\phi < \frac{1}{2}$ and $\epsilon \leq \frac{\phi}{2}$ the kth ϕ quantile of the two streams are different by at least $k\phi n$. Part (3) is proved by letting Δ be an update stream that deletes all elements from $str(S)$. Then, $L_k(str(S) \circ \Delta) = 0$ and $L_k(str(T) \circ \Delta) = \Omega(n^{1/k})$.

Proceeding as above, suppose Δ is an update stream that deletes all but one element from $str(S)$. Then, $H(str(S) \circ \Delta) = 0$. $str(T) \circ \Delta$ has $\Omega(n)$ elements and therefore $H(str(T) \circ \Delta) = \log n + \Theta(1)$. The multiplicative gap $\log n : 0$ is arbitrarily large—this proves part (4) of the lemma. $\square$

References

1. N. Alon, P. B. Gibbons, Y. Matias, and M. Szegedy. "Tracking Join and Self-Join Sizes in Limited Storage". In *Proc. ACM PODS*, 1999.
2. B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. "Models and Issues in Data Stream Systems". In *Proc. ACM PODS*, 2002.
3. L. Bhuvanagiri and S. Ganguly. "Estimating Entropy over Data Streams". In *Proc. ESA*, pages 148–159, 2006.
4. P. Bose, E. Kranakis, P. Morin, and Y. Tang. "Bounds for Frequency Estimation of Packet Streams". In *SIROCCO*, pages 33–42, 2003.
5. A. Chakrabarti, G. Cormode, and A. McGregor. "A Near-Optimal Algorithm for Computing the Entropy of a Stream". In *Proc. ACM SODA*, 2007.
6. M. Charikar, K. Chen, and M. Farach-Colton. "Finding frequent items in data streams". In *Proc. ICALP, 2002*, pages 693–703.

[3] number of sets that are within a distance of $\frac{n}{8}$ from a given set of size $\frac{n}{2}$ is $\sum_{r=0}^{\frac{n}{8}} \binom{n/2}{r}^2 \leq 2\binom{n/2}{n/8}^2$. Therefore, $|\mathcal{F}| \geq \frac{\binom{n}{n/2}}{2\binom{n/2}{n/8}^2} \geq \frac{2^{n/2}}{2(3e)^{n/8}} = \frac{1}{2}\left(\frac{16}{3e}\right)^{n/8}$.

7. G. Cormode and M. Garofalakis. "Sketching Streams Through the Net: Distributed Approximate Query Tracking". In *Proc. VLDB*, September 2005.
8. G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. "Finding Hierarchical Heavy Hitters in Data Streams". In *Proc. VLDB*, 2003.
9. G. Cormode and S. Muthukrishnan. "What's New: Finding Significant Differences in Network Data Streams". In *IEEE INFOCOM*, 2004.
10. G. Cormode and S. Muthukrishnan. "An Improved Data Stream Summary: The Count-Min Sketch and its Applications". *J. Algorithms*, 55(1):58–75, April 2005.
11. G. Cormode and S. Muthukrishnan. "What's hot and what's not: tracking most frequent items dynamically". *ACM Trans. Database Syst.*, 30(1):249–278, 2005.
12. E. D. Demaine, A. López-Ortiz, and J. I Munro. "Frequency estimation of internet packet streams with limited space". In *Proc. ESA*, 2002.
13. C. Estan, S. Savage, and G. Varghese. "Automatically inferring patterns of resource consumption in network traffic". In *Proc. ACM SIGCOMM*, pages 137–148, 2003.
14. S. Ganguly, D. Kesh, and C. Saha. "Practical Algorithms for Tracking Database Join Sizes". In *Proc. FSTTCS*, 2005.
15. S. Ganguly and A. Majumder. "Deterministic K-set Structure". In *Proc. ACM PODS*, 2006.
16. P. B. Gibbons and Y. Matias. "New Sampling-Based Summary Statistics for Improving Approximate Query Answers". In *Proc. ACM SIGMOD*, 1998.
17. A. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss. "Fast Small-space Algorithms for Approximate Histogram Maintenance". In *Proc. ACM STOC*, 2002.
18. A. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss. "How to Summarize the Universe: Dynamic Maintenance of Quantiles". In *Proc. VLDB*, 2002.
19. A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. "Surfing Wavelets on Streams: One-pass Summaries for Approximate Aggregate Queries". In *Proc. VLDB*, September 2001.
20. M. Greenwald and S. Khanna. "Space-efficient online computation of quantile summaries". In *SIGMOD*, 2001.
21. J. Hershberger, N. Shrivastava, S. Suri, and C.D. Toth. "Space Complexity of Hierarchical Heavy Hitters in Multi-Dimensional Data Streams". In *Proc. ACM PODS*, 2005.
22. R.M. Karp, S. Shenker, and C.H. Papadimitriou. "A Simple Algorithm for Finding Frequent Elements in Streams and Bags". *ACM TODS*, 28(1):51–55, 2003.
23. G. Manku and R. Motwani. "Approximate Frequency Counts over Data Streams". In *Proc. VLDB*, pages 346–357, August 2002.
24. G. Manku, S. Rajagopalan, and B. Lindsay. "Random sampling techniques for space efficient online computation of order statistics of large datasets". In *Proc. ACM SIGMOD*, 1999.
25. J. Misra and Gries. D. "Finding repeated elements". *Sci. Comput. Programm.*, 2:143–152, 1982.
26. S. Muthukrishnan. *"Data Streams: Algorithms and Applications"*. Foundations and Trends in Theoretical Computer Science, *Vol. 1, Issue 2*, 2005.
27. J.B. Rosser. "Explicit bounds on some functions on prime numbers". *Amer. J. Math.*, 63(1941).
28. R. Schweller, Z. Li, Y. Chen, Y. Gao, A. Gupta, Y. Zhang, P. Dinda, M-Y. Kao, and G. Memik. "Monitoring Flow-level High-speed Data Streams with Reversible Sketches". In *IEEE INFOCOM*, 2006.