A nearly optimal and deterministic summary structure for update data streams

Sumit Ganguly

Indian Institute of Technology, Kanpur

Abstract. We present a deterministic summary structure over update streams that enables deterministic and the first space-optimal algorithms for a variety of problems, including, estimating frequencies, finding approximate frequent items, finding approximate quantiles, finding hierarchical heavy hitters, approximately optimal B-bucket histograms, estimating inner product sizes, etc..

1 Introduction

The data streaming model presents a computational abstraction for applications (e.g., IP network monitoring, financial market monitoring, etc.), where, data arrives rapidly and continuously and has to be processed in an online fashion using sub-linear space. Formally, a data stream is modelled as a sequence of arrivals of the form (i, δ), where, i is the identity of an item belonging to the domain $[n] = \{1, \ldots, n\}$ and $\delta \in \{+1, -1\}$ depicts the increment/decrement to the frequency of i. The frequency f_i of an item i is defined as $f_i = \sum_{(i,\delta) \in \text{ stream}} \delta$. If $f_i \geq 0$ for all i then the stream is said to be strict; if $f_i \lesseqgtr 0$, then it is referred to as a general update stream. Some problems of practical relevance over data streams are, (P1) estimating the frequency of items (point queries), (P2) estimating frequencies of ranges (range-sum queries), (P3) finding approximate frequent items, (P4) finding approximate quantiles, (P5) finding approximate hierarchical heavy hitters, (P6) constructing approximately optimal B-bucket histograms, (P7) estimating inner-product sizes, etc..

Deterministic summary structures for update streams. Deterministic algorithms for these problems are often indispensable in practice. 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 [19], a false positive results in an innocent victim. For update streams, the CR-precis structure [12] presents a deterministic, sub-linear space algorithms for problems (P1) through (P7) above. A drawback of the CR-precis technique is that it requires space that is approximately the square of the space required by existing randomized algorithms for the same accuracy parameters. For example, the problem of finding approximate frequent items with parameter ϕ and tolerance ϵ that is,

return all items i with $|f_i| \geq \phi L_1$ and not return j where $|f_j| \leq (1-\epsilon)\phi L_1$) requires space $O(\frac{1}{\phi^2 \epsilon} \log^2(mn))$ [12] and analogously for the other problems. Here, $L_1 = L_1(f)$ is the L_1 norm of the frequency vector f defined as $\sum_{i \in [n]} |f_i|$. In comparison, existing randomized algorithms require space that is linear in $\frac{1}{\phi\epsilon}$ (up to logarithmic factors) for the same problem.

Problem Definitions. We now define problems (P1) through (P7). Algorithms for solving the above problems are reviewed in Appendix A.

(P1) The *point query* problem (or, frequency estimation) with parameter $0 < \epsilon < 1$ is— given $i \in [n]$, obtain an estimate $\hat{f}_i$ such that $|\hat{f}_i - f_i| \leq \epsilon L_1$.

(P2) Given a *range* $[l, r]$ from the domain $[n]$, the range frequency is defined as $f_{[l,r]} = \sum_{x=l}^{r} |f_x|$. The *range-sum* problem with parameter ϵ is: given a range $[l, r]$, return an estimate $\hat{f}_{[l,r]}$ such that $|\hat{f}_{[l,r]} - f_{[l,r]}| \leq \epsilon L_1$ [7,16].

(P3) Given parameters $0 < \epsilon, \phi < 1$, the *ϵ-approximate ϕ frequent items* problem is to (1)return all items with $|f_i| \geq \phi L_1$ and, (2) not return items such that $|f_i| < (1-\epsilon)\phi L_1$.

(P4) 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 $(j\phi - \epsilon)L_1 \leq \sum_{i=a_j}^{n-1} |f_i| \leq (j\phi + \epsilon)L_1$. Given ϵ, ϕ and j, the problem is to return an ϵ-approximate j^{th} quantile.

(P5) Given a hierarchy over the domain, the absolute frequency of a node X is defined as the sum of the absolute values of the frequencies of the leaf nodes (i.e., items) in the sub-tree rooted at X. A leaf node x is an *HHH* node provided $|f_x| > \frac{L_1}{s}$. An internal node is an *HHH* node provided that its absolute frequency, after discounting the absolute frequencies of all its descendant *HHH* nodes, is at least $\frac{L_1}{s}$. The ϵ-approximate ϕ *HHH* problem is to (1) find all *HHH* nodes, and, (2) not output any node output any node whose frequency, after discounting the absolute frequencies of descendant *HHH* nodes, is below $(1-\epsilon)\phi L_1$ [5,7,10,19].

(P6) A B-bucket histogram h is an n-dimensional vector that divides the domain $[n]$ into B non-overlapping intervals, say, $I_1, I_2, \ldots, I_B$ and associates a value v_j for each I_j. The cost of a B-bucket histogram h with respect to the frequency vector f is defined as $\|f - h\| = \sum_{j=1}^{B} \sum_{i \in I_j} \|f_i - v_j\|_k^k$, where, $k = 1$ denotes the L_1 norm and $k = 2$ denotes the L_2 norm. 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 $||f - \hat{h}|| \leq (1+\epsilon)||f - h^{opt}||$ [14,18].

(P7) Given streams R and S with item frequency vectors f and g respectively, the *inner product* $f \cdot g$ is defined as $\sum_{i \in [n]} f_i g_i$. The problem is to return $\hat{P}$ satisfying $|\hat{P} - f \cdot g| \leq \Delta$. Given space s bits, $\Delta = \Omega(s^{-1} L_1(f) L_1(g))$ [1].

Contributions. We present a deterministic summary structure for update streams that results in nearly space optimal algorithms for the problems defined above and is competitive with the space requirement of the best known randomized algorithms for the problems.

2 Deterministic summary structure

In this section, we present our summary structure and an algorithm for estimating the frequency of an item. Consider the following problem. Given $0 < \epsilon < 1$ and for a fixed n, design an $s \times n$ real matrix A with columns $[a_1, a_2, \ldots, a_n]$, where, s is as small as possible, satisfying the following conditions:

$$\|a_j\| = 1, \text{ for } j = 1, 2, \ldots, n, \text{ and } |a_i^T a_j| \leq \epsilon, \text{ for } 1 \leq i < j \leq n \ . \qquad (1)$$

Lemma 1 constructs such a matrix.

Lemma 1. *For $\epsilon < \frac{1}{36 \log n}$, there exists an $\frac{16 \log n}{\epsilon} \times n$ matrix A satisfying* (1).

Proof. Let C be an $s \times n$ random boolean matrix C whose entries C_{ij} is 1 with probability p and independent of each other. Let $c_1, c_2, \ldots, c_n$ denote the columns of C. Then,

$$\mathsf{E}\big[\|c_i\|\big] = sp \text{ and } \mathsf{E}\big[c_i^T c_j\big] = sp^2 \ .$$

If $p \geq \frac{24 \log(4n)}{s}$ and $T \geq \frac{2 \log(2n)}{\log \log n}$, then, by Chernoff's bounds,

$$\mathsf{Pr}\left\{ \mid \|c_i\| - sp \mid < \frac{sp}{2} \right\} \geq 1 - 2e^{-\frac{sp}{12}} \geq 1 - \frac{1}{8n^2}, \text{ and}$$
$$\mathsf{Pr}\left\{ |c_i^T c_j| > sp^2 + T \right\} \leq \left(\frac{esp^2}{T} \right)^T < \frac{1}{2n^2},$$

Define the matrix A whose columns a_i are normalization of c_i, that is, $a_i = \frac{c_i}{\|c_i\|}$, $i = 1, 2, \ldots, n$. Let $p = \frac{1}{\sqrt{s}}$ and $s \geq \frac{16(\log 2n)}{\epsilon}$. For $i < j$ with probability $1 - \frac{3}{4n^2}$,

$$a_i^T a_j = \frac{c_i^T c_j}{\|c_i\| \|c_j\|} \leq \frac{sp^2 + 2\log(2n)/(\log \log n)}{(sp)^2/4} \leq \frac{4}{s} + \frac{8 \log(2n)}{(sp)^2}$$
$$\leq \frac{\epsilon}{16 \log(2n)} + \frac{\epsilon}{2} < \epsilon \ .$$

Since, $\frac{1}{\sqrt{s}} = p \geq \frac{24 \log 2n}{s}$, we have, $s \geq 576 \log^2(2n)$. Letting $s = \frac{16(\log 2n)}{\epsilon} \geq 576 \log^2(2n)$, or $\epsilon \geq \frac{1}{36 \log(2n)}$. Using union bound, the probability that $a_i^T a_j \leq \epsilon$ for every $1 \leq i < j \leq n$ and $\|c_i\| \geq \frac{sp}{2}$ is at least $1 - \frac{3}{4n^2}\binom{n}{2} \geq 1 - \frac{3}{8} = \frac{5}{8}$. □

Reconsidering the above proof, we note that the process of random construction can be de-randomized to use a random hash function that is $O(\log n)$-wise independent [24]. Thus, there exists (a significant fraction of) an $O(\log^2 n)$ random bit seed that can be used to generate the entries of the matrix C and hence can be used to generate the columns a_i of A. The summary $y = Af$ can therefore be maintained in low space as follows. Upon receiving an input record (i, δ) on the stream, $y \ := \ y + \delta a_i$.

Estimator. The estimated frequency $\hat{f}_i$ is given as follows: $\hat{f}_i = a_i^T y$. For $i \in [n]$,

$$|\hat{f}_i - f_i| = |a_i^T(\sum_{j=1}^{n} f_j a_j) - f_i| = |\sum_{j \neq i} f_j a_i^T a_j| \leq \sum_{j \neq i} |f_j||a_i^T a_j| \leq (L_1 - |f_i|)\epsilon \ .$$

We summarize the discussion in the following lemma.

Lemma 2. *For $\epsilon < \frac{1}{36\log(2n)}$, there exists a deterministic summary structure that can be updated corresponding to stream updates. Further, for each $i \in [n]$, it can return an estimate $\hat{f}_i$ satisfying $|\hat{f}_i - f_i| \leq \epsilon(L_1 - |f_i|)$. The space required by the algorithm is $O(\frac{1}{\epsilon}(\log n)(\log mn))$ bits.* □

Note that the COUNT-MIN sketch structure requires space $O(\frac{1}{\epsilon}\log(mn)(\log\frac{1}{\delta}))$ and the space lower bound for deterministic algorithm is $\Omega(\frac{1}{\epsilon}(\log m)(\log \epsilon n))$ bits [12]. Lemma 2 can be used to derive near-optimal space algorithms for problems (P2) through (P7) using well-known techniques that reduce the respective problems to the problem of estimating frequencies. A review of the techniques is presented in A (see also [12] or [7] for a survey). Corollary 3 summarizes these properties.

Corollary 3. *There exist deterministic algorithms for each of the following problems with the following space requirement.*

1. *ϵ-approximate ϕ-frequent items using space $O((\phi\epsilon)^{-1}\log L_1 \log(n)\log(n\phi))$.*
2. *ϵ-approximate range-sum using space $O(\epsilon^{-1}\log(L_1)(\log n)(\log \epsilon n)$.*
2. *ϵ-approximate ϕ-quantiles using space $O((\phi\epsilon)^{-1}\log L_1 \log(n)\log(n\phi))$.*
4. *ϵ-approximate ϕ-hierarchical heavy hitters using space $O((\epsilon\phi)^{-2}h\log L_1(\log n))$, where, h is the height of the hierarchy,*
5. *$(1-\epsilon)$-optimal B-bucket histograms using space $O(B\epsilon^{-1}\log L_1(\log^2 n))$ (with respect to L_1 norm), and*
6. *inner product of streams R and S with frequency vectors f and g satisfying $|\hat{P} - f \cdot g| \leq \epsilon L_1(f)L_1(g)$ using space $O(\epsilon^{-1}\log n \log L_1)$.*

Proof. Statements 1 through 5 are proved using standard techniques that reduce each of the respective problems to the problem of estimating frequencies (and using the dyadic intervals technique–see Appendix A or [7,12] for review). We now consider statement (6) of the lemma. Let $y = Af$ and $z = Ag$ be the states of the summary structures for streams R and S respectively. Define $\hat{P} = y^T z$. Then,

$$\begin{aligned} |y^T z - f^T g| &= \big|(\sum_{i=1}^{n} f_i a_i^T)(\sum_{j=1}^{n} g_j a_j) - f^T g\big| \\ &= \big|\sum_{i=1}^{n} f_i g_i a_i^T a_i + 2\sum_{i<j} f_i g_j a_i^T a_j^T - f^T g\big| \leq \epsilon L_1(R)L_1(S) - f \cdot g \ . \end{aligned}$$ □

3 Conclusions

We present the first space-optimal (up to logarithmic factors) deterministic synopsis structure for estimating frequencies (i.e., $|\hat{f}_i - f_i| \leq \epsilon(L_1 - |f_i|$, for all $i \in [n]$). The technique enables the design of nearly space-optimal deterministic algorithms for a variety of problems including finding approximate frequent items, approximate quantiles, approximately optimal B-bucket histograms, etc..

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. P. Bose, E. Kranakis, P. Morin, and Y. Tang. "Bounds for Frequency Estimation of Packet Streams". In *SIROCCO*, pages 33–42, 2003.
3. M. Charikar, K. Chen, and M. Farach-Colton. "Finding frequent items in data streams". In *Proc. ICALP, 2002*, pages 693–703.
4. G. Cormode and M. Garofalakis. "Sketching Streams Through the Net: Distributed Approximate Query Tracking". In *Proc. VLDB*, September 2005.
5. G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. "Finding Hierarchical Heavy Hitters in Data Streams". In *Proc. VLDB*, 2003.
6. G. Cormode and S. Muthukrishnan. "What's New: Finding Significant Differences in Network Data Streams". In *IEEE INFOCOM*, 2004.
7. 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.
8. 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.
9. E. D. Demaine, A. López-Ortiz, and J. I Munro. "Frequency estimation of internet packet streams with limited space". In *Proc. ESA*, 2002.
10. C. Estan, S. Savage, and G. Varghese. "Automatically inferring patterns of resource consumption in network traffic". In *Proc. ACM SIGCOMM*, pages 137–148, 2003.
11. S. Ganguly, D. Kesh, and C. Saha. "Practical Algorithms for Tracking Database Join Sizes". In *Proc. FSTTCS*, 2005.
12. S. Ganguly and A. Majumder. "CR-precis: A deterministic summary structure for update data streams". Manuscript: http://arxiv.org/abs/cs.DS/0609032, July 2005.
13. P. B. Gibbons and Y. Matias. "New Sampling-Based Summary Statistics for Improving Approximate Query Answers". In *Proc. ACM SIGMOD*, 1998.
14. 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.
15. A. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss. "How to Summarize the Universe: Dynamic Maintenance of Quantiles". In *Proc. VLDB*, 2002.
16. 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.
17. M. Greenwald and S. Khanna. "Space-efficient online computation of quantile summaries". In *SIGMOD*, 2001.
18. S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. "Histogramming Data Streams with Fast Per-Item Processing". In *Proc. ICALP*, pages 681–692, 2002.

19. 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.
20. 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.
21. G. Manku and R. Motwani. "Approximate Frequency Counts over Data Streams". In *Proc. VLDB*, pages 346–357, August 2002.
22. 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.
23. J. Misra and Gries. D. "Finding repeated elements". *Sci. Comput. Programm.*, 2:143–152, 1982.
24. J. Schmidt, A. Siegel, and A. Srinivasan. "Chernoff-Hoeffding Bounds with Applications for Limited Independence". In *Proc. ACM SODA*, pages 331–340, 1992.
25. 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.

A Review

In this section, we briefly review some basic problems over data streams.

Preliminaries A *dyadic interval* at level l is an interval of size 2^l from the family of intervals $\{[i2^l + 1, (i+1)2^l], 0 \leq i \leq \lceil \frac{n}{2^l} \rceil - 1\}$, for $0 \leq l \leq \log n$, assuming that n is a power of 2. The set of dyadic intervals of levels 0 through $\log n$ form a complete binary tree as follows. The root of the tree is the single dyadic interval $[1, n]$. The nodes at distance h from the root are the set of dyadic intervals at level $\log N - h$. Moreover, for $0 \leq h < \log N$, each dyadic interval at level h is of the form $I_h = [i\frac{n}{2^h} + 1, (i+1)\frac{n}{2^h}]$ and has two children at level $h-1$, namely, the left and the right halves of I_h. The left child of I_h is the interval $[2i\frac{n}{2^{h+1}} + 1, (2i+1)\frac{n}{2^{h+1}}]$ and the right child is the interval $[(2i+1)\frac{n}{2^{h+1}} + 1, (2i+2)\frac{n}{2^{h+1}}]$.

Point queries or Frequency Estimator. For *insert-only streams*, a number of algorithms are known for estimating frequencies such that $|\hat{f}_i - f_i| \leq \epsilon L_1$. The Misra-Gries algorithm [23], refined in [9,2,20], uses $\epsilon^{-1} \log m$ bits and returns $\hat{f}_i$ such that $f_i \leq \hat{f}_i \leq f_i + \epsilon L_1$. The deterministic *Lossy Counting* algorithm [21] uses $O(\epsilon^{-1} \log(\epsilon L_1) \log L_1)$ bits to present similar guarantees as [23]. The randomized *Sticky Sampling* algorithm [21] extends the randomized *Counting Samples* algorithm [13] to return an estimate satisfying $f_i - \epsilon L_1 \leq \hat{f}_i \leq f_i$ with probability $1 - \delta$ using space $O(\epsilon^{-1} \log \frac{1}{\delta} \log m)$ bits. For *strict update streams*, the Count-Min sketch algorithm satisfies $f_i \leq \hat{f}_i \leq f_i + \epsilon L_1$ with probability $1 - \delta$ using space $O(\epsilon^{-1} \log \frac{1}{\delta} \log L_1)$ bits. For *general update streams*, the Count-Min sketch algorithm satisfies $|\hat{f}_i - f_i| \leq \epsilon L_1$ using the same order of space. The CountSketch algorithm [3] is applicable for *general update streams* and satisfies $|\hat{f}_i - f_i| \leq (\epsilon^{-1} F_2^{res}(\epsilon^{-1}))^{1/2} \leq \epsilon L_1$ with probability $1 - \delta$ using space $O(\epsilon^{-1}(\log \delta^{-1}) \log L_1)$, where, $F_2^{res}(s)$ is the sum of the squares of all but

the top-s frequencies in the stream. [2] show that any algorithm that returns $\hat{f}_i$ satisfying $|\hat{f}_i - f_i| \leq \epsilon L_1$ must use $\Omega(\epsilon^{-1} \log(\epsilon n))$ bits. [12] presents a deterministic algorithm for for this problem using space $O(\epsilon^{-1}(\log n)^2(\log \epsilon^{-1})^{-2}(\log L_1))$.

Approximate frequent items. The problem of efficiently finding ϵ-approximate ϕ-frequent items can be solved by keeping a point query estimator corresponding to each dyadic level $l = 0, \ldots, \log(N\phi)$ [7]. By construction, each item i belongs to a unique dyadic interval at level l, namely, the l^{th} level ancestor of the interval $[i, i]$ in the dyadic tree. The "items" at level l are the set of dyadic intervals $\{[j2^l, (j+1)2^l - 1]\}_{0 \leq j \leq 2^{d-l}}$ and are identifiable with the domain $\{0, 1, \ldots, 2^{d-l}\}$. With this interpretation, an arrival over the stream of the form (i, v) is processed as follows: update the item $((i \,\%\, 2^l), v)$ for each level $l = 0, 1, \ldots, \lfloor \log(\phi n) \rfloor$. The frequency of a dyadic interval I is defined as the sum of the individual frequencies of items in I, and is denoted as f_I. Since each level 0 item belongs to one and only one dyadic interval at a given level l, the sum of the interval frequencies at level l is the same as the sum of the item frequencies at level 0, which is L_1 for strict update streams. If an item i is frequent (i.e., $f_i \geq \phi L_1$), then the dyadic interval that contains i at any level l has frequency at least f_i and is therefore also frequent at level l. Hence, at each level l starting from $\lfloor \log(\phi n) \rfloor$ and decrementing down to 1, it suffices to consider only those dyadic intervals that are frequent at level l. The procedure begins by enumerating $O(\lceil \phi^{-1} \rceil)$ dyadic intervals at level $\lfloor \log(\phi n) \rfloor$ and keeping as candidate intervals whose estimated frequency is at least ϕL_1. In general, at level l, there are $O(\phi^{-1})$ candidate intervals. For each candidate interval at level l, we consider its left and right child intervals at level $l - 1$, and repeat the procedure. Since, at any level, the number of candidate intervals is $O(\phi^{-1})$, the total number of intervals considered in the iterations is $O(\phi^{-1} \log(\phi n))$. Using the Count-Min sketch algorithm at each dyadic level with total space $O((\epsilon\phi)^{-1} \log((\phi\delta)^{-1} \log(\phi n)) \log(\phi n))$ counters, one can return all frequent items with probability 1 and not return any item with frequency $(1 - \epsilon)\phi L_1$ with probability $1 - \delta$. The CR-precis structure [12] presents a deterministic algorithm for for this problem using space $O((\phi\epsilon)^{-2}(\log((\phi\epsilon)^{-1})(\log n)^2 \log(\phi^{-1}\epsilon^{-1}) \log(\phi n)\ (\log L_1))$.

Hierarchical heavy hitters. [5] presents a solution to the ϵ-approximate ϕ-*HHH* problem 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. The space required is $O((\epsilon^{-1}\phi^{-2} \log(\phi\delta^{-1}n)(\log n)(\log L_1))$ using the Count-Min sketch summary structure. [19] presents an $\Omega(\phi^{-2})$ space lower bound for constant ϵ $(= 0.01)$. The CR-precis structure [12] presents a deterministic algorithm for update streams requiring $O(\epsilon^{-2}\phi^{-4}h^2 \log(\phi^{-2}\epsilon^{-1}h) \log L_1)$ bits.

Range queries and approximate quantiles. A standard approach for solving range queries is to decompose a given interval as a canonical disjoint sum of at most $2 \log N$ dyadic intervals [16]. [7] uses Count-Min sketches to estimate range-sums using space $O(\epsilon^{-1} \log\log(n\delta^{-1}) \log n \log L_1)$ bits with probability $1-\delta$. The

ϵ-approximate ϕ-quantile problem has been studied in [7,17,15,22]. For insert-only streams, [17] presents an algorithm requiring space $O((\log \epsilon^{-1}) \log(\epsilon L_1))$ for insert-only streams. For strict update streams, the problem of finding approximate quantiles can be reduced to that of estimating range sums [15] as follows.

The range-sum query problem, that is, estimating the frequency of a given range, can be solved by using the technique of dyadic intervals [16]. Any range can be uniquely decomposed into the disjoint union of at most $2 \log N$ dyadic intervals of maximum size (for example, 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 $O(\epsilon^{-1} \log(\delta^{-1} \log n) \log n \log L_1)$ bits with probability $1 - \delta$ [7]. The problem of finding ϵ-approximate ϕ-quantiles can be reduced to range-sum queries 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 range sum $f_{[a_k, N-1]}$ lies between $(k\phi - \epsilon)L_1$ and $(k\phi + \epsilon)L_1$. [7] uses COUNT-MIN sketch es to find ϵ-approximate ϕ-quantiles with confidence $1 - \delta$ using space $O(\epsilon^{-1} \log^2 n \log(\phi^{-1} \delta^{-1} \log n))$.

Approximately optimal histograms and inner products. An algorithm for this problem is presented in a seminal paper [14] using space and time poly $(B, \frac{1}{\epsilon}, \log m, \log N)$ and improved in [18]. Given two streams R and S with item frequency vectors f and g respectively, the *inner product* $f \cdot g$ is defined as $\sum_{i \in [n]} f_i \cdot g_i$. The problem is to return an estimate $\hat{P}$ satisfying $|\hat{P} - f \cdot g| \leq \epsilon L_1(f) L_1(g)$. The work in [1] presents a space lower bound of $s = \Omega(\epsilon^{-1})$. Randomized algorithms [1,4,11] match the space lower bound, up to poly-logarithmic factors. The CR-precis structure presents an algorithm for estimating inner products satisfying $|\hat{P} - f \cdot g| = \epsilon L_1(f) L_1(g)$ using space $O(\epsilon^{-2} (\log^2 n)(\log^2 \epsilon^{-1})(\log \epsilon^{-1})(\log L_1)$.