Data Stream Algorithms for Vectors: Draft Chapter*

October 27, 2013

In this chapter, we study one of the common forms in which modern data problems arise. Traditional data problems consider data that is stored, say, the records of all employees in a company, students in an University, and so on. These databases change, albeit slowly, and data analyses often assume the data can be accessed repeatedly and it will not change during the analyses. In contrast, modern data arises as streams of measurements or observations arriving over time and describe an underlying signal in some high dimensional space. For example, the collection of transactions at an ATM or the photos of cars passing a traffic intersection or the descriptions of IP packets passing through a router are all examples of data streams. The underlying signals could be the current balance of each bank account or the number of times each car goes through the intersection or the number of bytes sent by each IP address, respectively. As is evident from these examples, the dimension of these signals — the number of bank accounts or cars or IP addresses — is potentially large. Also, for running the network of ATM or traffic system or the IP network, one needs to monitor these signals and analyze them for potential security reasons, or optimizing ones' operations or reporting and so on. These considerations motivate the study of problems in this chapter.

Formally, we consider a stream of m updates $S = \langle a_1, \ldots, a_m \rangle$ that determine a vector $\mathbf{x} \in \mathbb{R}^n$. We assume that $\mathbf{x} = (x_1, \ldots, x_n)$ is initially the zero vector. An update $a_t = (i_t, \Delta_t) \in [n] \times \mathbb{R}$ encodes the update

$$x_{i_t} \leftarrow x_{i_t} + \Delta_t \ .$$

Note that after m updates we have $x_j = \sum_{t \in [m]: j = i_t} \Delta_{i_t}$. For example, if $n = 4$ and $m = 5$, the stream $S = \langle (1,2), (2,-0.5), (4,1), (1,-1), (4,2) \rangle$ encodes the vector

$$\mathbf{x} = (1, -0.5, 0, 3)$$

In motivation examples earlier, n is the dimension of the signals and m is the number of transactions, both of which may be large in modern data application.

We will approach problems of analyzing such data streams, as is, typical, requiring that we use very little space to represent the streams. In particular, for a given function g, the goal is to return an approximation of $g(\mathbf{x})$ using space that is sub-linear in m and n, typically, polylogarithmic in these factors. A case has been made for such stringent space constraints in prior work over the past decade, primarily because the streams arrive rapidly and high speed memory is expensive. See [Mut06] for a detailed discussion.

*Draft of a chapter from the forthcoming textbook "Data Stream Algorithms and Sketches" by Andrew McGregor and S. Muthukrishnan. Do not distribute without permission of the authors. Latest version can be found at `http://people.cs.umass.edu/~mcgregor/book/book.html`. Please send comments and corrections to `mcgregor@cs.umass.edu`.

We will focus on three basic problems with signal analysis. These basic problems will let us introduce some of the powerful techniques invented in the past few decades. Ultimate, these problems by themselves will be of interest in some applications. In applications where modern data problems arise, like sparse signal recovery or entropy estimation or cascaded aggregates, these techniques will prove useful. The problems of interest are:

1. Frequency Moments: Estimating $F_k = \sum_{i\in[n]} |x_i|^k$
2. Distinct Elements: Estimating $F_0 = |\{i \in [n] : x_i \neq 0\}|$
3. Heavy Hitters: Finding all $i \in [n]$ with $|x_i| \geq \phi\,(F_k)^{1/k}$ for some $\phi \in (0, 1)$.

A special case is the *increment-only* model in which all Δ_t are assumed to equal 1 and are omitted from the stream. In this model x_j is referred to the frequency of j in the stream.

1 Increment-Only Streams: Sampling and Counting

In this section we will describe several simple sampling and counting algorithms that already help us solve interesting problems.

1.1 Misra-Gries Algorithm

We first consider a deterministic algorithm using k counters such that, when queried with $i \in [n]$, will return an estimate $\hat{x}_i$ of x_i such that

$$x_i - \sum_{j\neq i} \frac{x_j}{k-1} \leq \hat{x}_i \leq x_i\ .$$

The algorithm maintains k counters $c_1, \ldots, c_k$, initially zero, along with k elements $e_1, \ldots, e_k$ that are currently being "monitored." On the arrival of a new element e we do one of the following:

Case 1: If $e_j = e$ for some j: Increment c_j

Case 2: If $e_j \neq e$ for all $j \in [k]$ and $c_i = 0$ for some $i \in [k]$: Set $c_i \leftarrow 1$ and $e_i \leftarrow e$

Case 3: If $e_j \neq e$ for all $j \in [k]$ and $c_i > 0$ for all $i \in [k]$: Decrement c_i for all $i \in [k]$

Then, to estimate x_i we return:

$$\hat{x}_i = \begin{cases} c_j & \text{if } e_j = i \text{ for some } j \in [k] \\ 0 & \text{otherwise} \end{cases}\ .$$

Lemma 1. *For all i, $x_i - \sum_{j\neq i} \frac{x_j}{k-1} \leq \hat{x}_i \leq x_i$.*

Proof. The second inequality is clear since a counter corresponding to i will only be incremented when i appears in the stream. Define b to be the number of occurrences of Case 3 and note that $\hat{x}_i \geq x_i - b$. To establish the first inequality, consider the quantity $C = \sum_{j\in[k]:e_j\neq i} c_j$. **MUTHU: Do we need j to be in [k]?** Note that $0 \leq C \leq m - x_i$ since C is incremented at most $m - x_i$ times. Hence,

$$b \leq \frac{m - x_i}{k-1}$$

because each application of Case 3 decrements C by $k - 1$. □

1.2 Reservoir Sampling

A standard approach for estimating a function on a large data set is to sample from the data set and make an inference from the set of samples. In this section, we show how to sample uniformly at random from an increment-only stream even if we do not know the length of the data stream. We will later show that more powerful forms of sampling are possible.

- Algorithm: Given stream $\langle a_1, a_2, \ldots \rangle$.
  - Initially $s = a_1$
  - On seeing the t-th element, set $s \leftarrow a_t$ with probability $1/t$

For analysis, consider, what's the probability that $s = a_i$ at some time $t \geq i$? This is:

$$\Pr\left[s = x_i\right] = \frac{1}{i} \times \left(1 - \frac{1}{i+1}\right) \times \ldots \times \left(1 - \frac{1}{t}\right) = \frac{1}{t}$$

To get k samples we use $O(k \log n)$ bits of space, and get a precisely uniform sample with **MUTHU: What is the precise claim here?**

1.3 AMS Sampling

A more advanced sampling technique was introduced by Alon, Matias and Szegedy [AMS99]. It is particularly useful when trying to estimate aggregates of the form

$$f(\mathbf{x}) := \sum_{i \in [n]} f(x_i)$$

where f is some function with the property $f(0) = 0$.

The basic idea is to generate a random variable R defined thus: Pick $J \in [m]$ uniformly at random and let $R = |\{j : a_j = a_J,\ J \leq j \leq m\}|$. Let $\mathcal{D}$ be the distribution of R. Then we define the random variable

$$X = m(f(R) - f(R-1)) \ .$$

It can easily be shown that $\mathbb{E}\left[X\right] = f(A)$:

$$\begin{aligned}
\mathbb{E}\left[X\right] &= \sum_{j \in [n]} \Pr\left[a_J = j\right] \mathbb{E}\left[f(R) - f(R-1) | a_J = j\right] \\
&= \sum_{j \in [n]} \frac{x_j}{m} \cdot \left(\frac{m(f(x_j) - f(x_j - 1)) + \ldots + m(f(1) - f(0))}{x_j} \right) \\
&= \sum_{j \in [n]} f(x_j)
\end{aligned}$$

Hence, if the variance of X is low then by computing a "small" number of independent samples from $\mathcal{D}$ we can get a good approximation for $\sum_{j \in n} f(x_j)$.

There are several details in applying this sampling method, for example, R has to be generated using small space, and the variance of X has to be bounded, and so on. We demonstrate this via applications to estimating frequency moments and entropy.

1.3.1 Application: Frequency Moments

Recall $F_k = \sum_i x_i^k$ for $k \in \{1, 2, 3, \ldots\}$ and let $F_\infty = \max_i |f_i|$. Use AMS estimator with $X = m(r^k - (r-1)^k)$ and note that

$$\mathbb{E}[X] = F_k$$

Exercise 2. *Show that* $0 \leq X \leq mk\,(F_\infty)^{k-1}$.

Suppose we generate t independent copies of X in parallel and let $\hat{X}$ be the average value. By an application of the Chernoff bound,

$$\Pr\left[|\hat{X} - F_k| \geq \epsilon F_k\right] \leq 2\exp\left(-\frac{tF_k\epsilon^2}{3mkF_\infty^{k-1}}\right) .$$

Hence, taking $t = \frac{3mk\,(F_\infty)^{k-1}\log(\frac{2}{\delta})}{\epsilon^2 F_k}$ ensures that

$$\Pr\left[|\hat{X} - F_k| \geq \epsilon F_k\right] \leq \delta \; .$$

We next need to bound t in terms of n, ϵ, and δ.

Lemma 3. *For all* $k \geq 1$,

$$\frac{m(F_\infty)^{k-1}}{F_k} \leq n^{1-1/k}.$$

Proof. We consider two cases depending on the relative size of F_∞^k and $n(m/n)^k$. First suppose $F_\infty^k \geq n(m/n)^k$. Then,

$$\frac{m(F_\infty)^{k-1}}{F_k} \leq \frac{m(F_\infty)^{k-1}}{F_\infty^k} = \frac{m}{F_\infty} \leq \frac{m}{n^{1/k}(m/n)} = n^{1-1/k}$$

Alternatively suppose that $F_\infty^k \leq n(m/n)^k$. Then,

$$\frac{mF_\infty^{k-1}}{F_k} \leq \frac{mn^{1-1/k}(m/n)^{k-1}}{n(m/n)^k} = n^{1-1/k}$$

where the first inequality follows since $F_k \geq n(m/n)^k$ by appealing to the convexity of $g(x) = x^k$. □

Therefore, we have proved the following result.

Theorem 4. *We obtain an* ϵ *approximation to computing* F_k *which uses space* $O(\frac{kn^{1-1/k}\log\frac{1}{\delta}}{\epsilon^2})$ *and succeeds with probability at least* $1 - \delta$.

In particular, for $k = 2$, this gives an algorithm that uses $O(\sqrt{n})$ space unto polylogarithmic terms, and this is already sub linear in the dimension n of the underlying signal. We will obtain better bounds for this problem later.

1.3.2 Application: Entropy

Given a probability distribution $\mathbf{p}$ over $[n]$ the Shannon entropy is defined as

$$H(\mathbf{p}) := -\sum_{i\in[n]} p_i \log_2 p_i$$

It is a quantity that arises in numerous settings including monitoring network traffic. For our purposes, we consider $\mathbf{p}$ to be empirically defined by the data stream. In particular, we define $p_i = x_i/m$, i.e., we consider p_i to be the relative frequency of i in the stream.

The algorithm we present consists of two sub-algorithms which are run in parallel. The answer returned by the first algorithm is correct if $p_\ell \le 7/8$ where $\ell = \operatorname{argmax}_{i\in[n]} p_i$. The answer returned by the second algorithm is correct if $p_\ell \ge 3/4$.

Case 1: $p_\ell \le 7/8$: Use the AMS estimator with $X = (-r\log_2 \frac{r}{m} + (r-1)\log_2 \frac{r-1}{m})$.

$$\mathbb{E}[X] = H(\mathbf{p}) \ .$$

Exercise 5. *Prove that $-\log_2 e \le X \le \log_2 m$ and $H(p) \ge \frac{1}{8}\log_2\frac{1}{8} + \frac{7}{8}\log_2\frac{7}{8} = 0.543$ if $p_\ell \le 7/8$.*

As we did for frequency moments, suppose we generate t independent copies of X in parallel and let $\hat{X}$ be the average value. Unfortunately, this time we can not apply the Chernoff bound directly because X may be negative. However, the following simple lemma establishes that t need not be too large via an indirect application.

Lemma 6. *If $t > c\epsilon^{-2}\ln(2\delta^{-1})$ for some sufficiently large constant $c > 0$ then*

$$\Pr\left[|\hat{X} - H(\mathbf{p})| \ge \epsilon H(\mathbf{p})\right] \le \delta \ .$$

Proof. We apply the Chernoff bound to the estimate $Y = X + \log_2 e$ where $\hat{Y} = H(\mathbf{p}) + \log_2 e$. Since $0 \le Y \le \log_2 em$, we know

$$\Pr\left[|\hat{Y} - H(\mathbf{p}) - \log_2 e| \ge \gamma(H(\mathbf{p}) + \log_2 e)\right] \le 2\exp\left(-\frac{t(H(\mathbf{p})+\log_2 e)\gamma^2}{3\log_2 em}\right)$$

Setting

$$\gamma = \frac{0.543\epsilon}{0.543 + \log_2 e}$$

ensures that

$$\gamma(H(\mathbf{p}) + \log_2 e) \le \frac{\epsilon H(\mathbf{p})}{H(\mathbf{p}) + \log_2 e} \cdot (H(\mathbf{p}) + \log_2 e) = \epsilon H(\mathbf{p})$$

since $H(\mathbf{p}) \ge 0.543$. Therefore, if

$$t \ge \frac{3\ln(2/\delta)\log_2 em}{(0.543 + \log_2 e)\gamma^2}$$

ensures

$$\Pr\left[|\hat{X} - H(\mathbf{p})| \ge \epsilon H(\mathbf{p})\right] \le \delta \ .$$

$\square$

Case 2: $p_\ell \geq 3/4$**:** We can write $H(\mathbf{p})$ as

$$H(\mathbf{p}) = -p_\ell \log_2 p_\ell - \sum_{i \neq \ell} p_i \log p_i = -p_\ell \log_2 p_\ell - (1 - p_\ell) \sum_{i \neq \ell} \frac{x_i}{m - x_\ell} \log p_i$$

Using the Misra-Gries algorithm described in Section 1.1, in $O(\epsilon^{-1})$ space we can identify ℓ and find an estimate $\hat{p}_\ell$ such that

$$p_\ell - \frac{\epsilon(1 - p_\ell)}{4} \leq \hat{p}_\ell \leq p_\ell \ .$$

Exercise 7. *Prove that* $\frac{1-p_\ell}{1-\hat{p}_\ell} = 1 \pm \frac{\epsilon}{3}$ *and* $\frac{p_\ell \log_2 p_\ell}{\hat{p}_\ell \log_2 \hat{p}_\ell} = 1 \pm \frac{\epsilon}{3}$ *if* $p_\ell \geq \frac{3}{4}$.

Hence, it remains to show how to find a $(1 + \frac{\epsilon}{3})$ approximation of $-\sum_{i \neq \ell} \frac{x_i}{m - x_\ell} \log p_i$. The algorithm to do this is an extension of AMS where, rather than finding a single value R, we find two random variables R_1 and R_2 defined as follows.

1. Pick $J_1 \in_R [m]$ and let $R_1 = |\{j : a_j = a_{J_1}, J_1 \leq j \leq m\}|$.
2. Pick $J_2 \in_R \{j \in [m] : a_j \neq a_{J_1}\}$ and let $R_2 = |\{j : a_j = a_{J_2}, J_2 \leq j \leq m\}|$.

Observe that computing J_1, R_1, J_2, R_2 in small space is easy if we have two passes over the data: in the first pass we compute J_1 and R_1 and in the second pass, we compute J_2 and R_2. However, with a bit of care it is possible to compute J_1, R_1, J_2, R_2 in small space given only a single pass. With each stream element a_i associate a random value $c_i \in_R [0, 1]$ and at time t, let $J_{1,t} = \text{argmin}_{i \in [t]} c_i$, $J_{2,t} = \text{argmin}_{i \in [t] : a_i \neq a_{J_1}} c_i$, $R_{1,t} = |\{j : a_j = a_{J_1}, J_1 \leq j \leq t\}|$, $R_{2,t} = |\{j : a_j = a_{J_2}, J_2 \leq j \leq t\}|$, $a_{1,t} = a_{J_{1,t}}$, $a_{2,t} = a_{J_{2,t}}$, $c_{1,t} = c_{J_{1,t}}$, and $c_{2,t} = c_{J_{2,t}}$. Then, $J_{1,t+1}, J_{2,t+1}, R_{1,t+1}, R_{2,t+1}, a_{1,t+1}, a_{2,t+1}, c_{1,t+1}$, and $c_{2,t+1}$ can be computed from $a_{t+1}, c_{t+1}, J_{1,t}, J_{2,t}, R_{1,t}, R_{2,t}, a_{1,t}, a_{2,t}, c_{1,t}$, and $c_{2,t}$.

At the end of the stream, once $\ell, R_{1,n}, R_{2,n}, a_{1,n}, a_{2,n}$ have been computed, let

$$R = \begin{cases} R_1 & \text{if } a_{1,n} \neq \ell \\ R_2 & \text{otherwise} \end{cases}$$

and let

$$X = -R \log_2 \frac{R}{m} + (R - 1) \log_2 \frac{R - 1}{m}.$$

Exercise 8. *Prove that* $\mathbb{E}[X] = -\sum_{i \neq \ell} \frac{x_i}{m - x_\ell} \log p_i$ *and* $0 \leq X \leq \log_2 m$.

Therefore by averaging parallel repetitions of the AMS estimator and applying the Chernoff bound we get a (ϵ, δ) estimator $-\sum_{i \neq \ell} \frac{x_i}{m - x_\ell} \log p_i$. Putting together all the cases gives the following theorem:

Theorem 9. *The algorithm finds an* ϵ *approximation for* $H(\mathbf{p})$ *using space* $O(\frac{\log \frac{1}{\delta} \log m}{\epsilon^2})$ *and succeeds with probability at least* $1 - \delta$.

2 Basic Linear Sketches

In this section, we describe the *linear sketching* approach to stream computation. One can view specific sketches as comprising two components.

- *Projection:* A (random) *projection matrix* $A \in \mathbb{R}^{k\times n}$ is implicitly stored by the algorithm. As the stream is processed we compute $A\mathbf{x}$. It is possible to do this without materializing the length n vector $\mathbf{x}$ and instead only store the length $k \ll n$ vector $A\mathbf{x}$. If the stream increments the i-th coordinate of $\mathbf{x}$ by Δ then we update $A\mathbf{x}$ by:

$$A\mathbf{x} \leftarrow A\mathbf{x} + \Delta A\mathbf{e}_i^T$$

  where $\mathbf{e}_i$ is the i-th standard basis vector. It is natural to think of $\mathbf{x}$ being embedded into a smaller-dimensional space.

- *Post-Process:* The other component is an algorithm to post-process $\mathbf{Ax}$ and return an estimate for the quantity of interest.

For this to be useful in streaming algorithms, the entries of $\mathbf{A}$ should be computable in small space and time as $\mathbf{x}$ is updated by the stream. This is particularly important when the matrix is random since if we must store $\Omega(nk)$ random bits to express $\mathbf{A}$ then we would be better off materializing $\mathbf{x}$. We can get around this in various ways, e.g., by using pseudo-random generators or hash functions that are not fully independent.

2.1 Distinct Items

A large amount of work has been done on estimating $F_0 = \sum_i |x_i|^0$, the number of distinct items in a stream [BYJK$^+$02,IW03]. This problem was originally considered by Flajolet and Martin [FM85] in another of the "classic" streaming papers.

In order to (ϵ, δ) approximate $F_0 = \sum_i |x_i|^0$, we first consider the following simpler problem: For given threshold $T > 0$, with probability $1 - \delta$ distinguish between the cases:

1. $F_0 > (1+\epsilon)T$
2. $F_0 < (1-\epsilon)T$

Note that if we can solve the simpler problem, can solve the original problem by testing the following $O(\epsilon^{-1}\log n)$ possible values for the threshold T in parallel:

$$T = 1, (1+\epsilon), (1+\epsilon)^2, \ldots, n$$

To solve the simpler problem we proceed as follows:

- *Projection:* Choose random sets $S_1, S_2, \ldots, S_k \subset [n]$ where $\Pr[i \in S_j] = 1/T$. This defines a projection matrix $\mathbf{A}$ where:

$$\mathbf{A}_{i,j} = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{otherwise} \end{cases}$$

  Compute the projection $\mathbf{Ax}$ and let $s_i = \sum_{j\in S_i} x_i = [\mathbf{Ax}]_i$

- *Post-Process:* If at least k/e of the s_j are zero, output $F_0 < (1-\epsilon)T$

Lemma 10. *If T is sufficiently large and $\epsilon < 1/2$:*

1. *If $F_0 > (1+\epsilon)T$, $\Pr[s_j = 0] < 1/e - \epsilon/3$*

2. If $F_0 < (1-\epsilon)T$, $\Pr[s_j = 0] > 1/e + \epsilon/3$

Proof. Note that $s_j = 0$ iff $i \notin S_j$ for all the F_0 values of i with $x_i > 0$. Hence,

$$\Pr[s_j = 0] = (1 - 1/T)^{F_0} \ .$$

If $F_0 > (1+\epsilon)T$,

$$(1-1/T)^{F_0} \leq e^{-(1+\epsilon)} < e^{-1} - \epsilon/3 \ .$$

If $F_0 < (1-\epsilon)T$,

$$(1-1/T)^{F_0} \geq (1-1/T)^{(1-\epsilon)T} > e^{-1} + \epsilon/3 \ .$$

where the second inequality follows for sufficiently large T. □

Applying the Chernoff bound with $k = O(\epsilon^{-2}\log\delta^{-1})$ ensures correctness with probability $1-\delta$.

2.2 Self-Joins

In this section we consider the problem of finding an (ϵ,δ) approximation for $F_2 = \sum_i x_i^2$, also known as a self-join.

- *Projection:* Let $\mathbf{A} \in \{-1,1\}^{k\times n}$ where entries of each row are 4-wise independent and rows are independent. Compute $\mathbf{Ax}$.
- *Post-Process:* Group entries of the sketch into $a = O(\log\delta^{-1})$ groups of $b = 12\epsilon^{-2}$. Let $Y_1, Y_2, \ldots, Y_a$ be the average of squared entries in each group. Return median$(Y_1,\ldots,Y_a)$.

Lemma 11. *For a fixed ℓ, let $\mathbf{z}$ be the ℓ-th row of $\mathbf{A}$ and let $s = \mathbf{z}\cdot\mathbf{x}$ be the ℓ-th row of $\mathbf{Ax}$. Then $\mathbb{E}\left[s^2\right] = F_2$ and $\mathbb{V}\left[s^2\right] \leq 4F_2^2$.*

Proof. Since $\mathbb{E}[z_iz_j] = 0$ unless $i = j$,

$$\mathbb{E}\left[s^2\right] = \mathbb{E}\left[\sum_{i,j\in[n]} z_iz_jx_ix_j\right] = \sum_{i,j\in[n]} x_ix_j\mathbb{E}[z_iz_j] = \sum_{i\in[n]} x_i^2$$

For the variance bound, first note that $\mathbb{E}[z_iz_jz_kz_l] = 0$ unless $(i,k) = (j,l)$, $(i,j) = (k,l)$ or $(i,j) = (l,k)$. Then

$$\mathbb{V}\left[s^2\right] = \mathbb{E}\left[s^4\right] - \mathbb{E}\left[s^2\right]^2 = \sum_i x_i^4 + 6\sum_{i<j} x_i^2x_j^2 - (\sum_{i\in[n]} x_i^2)^2 = 4\sum_{i<j} x_i^2x_j^2 \leq 4F_2^2 \ .$$

□

It follows that $\mathbb{V}[Y_i] = F_2$ and $\mathbb{V}[Y_i] = \mathbb{V}\left[s^2\right]/b = \epsilon^2F_2^2/3$. The Chebyshev bound implies that

$$\Pr[|Y_i - F_2| > \epsilon F_2] \leq \frac{\epsilon^2F_2^2/3}{\epsilon^2F_2^2} = 1/3 \ .$$

By an application of the Chernoff bound, median$(Y_1,\ldots,Y_a)$ is an (ϵ,δ) approximation of F_2.

2.2.1 Extension: Johnson-Lindenstrauss and p-stable Distributions

An interesting class of such sketches were defined by Indyk [Ind06], where each entry $\mathbf{A}$ was i.i.d. samples from a *p-stable distribution*. In particular, we consider $A_{ij} \sim \mathcal{D}_p$ where the distribution $\mathcal{D}_p$ has the property that for any constants $a, b \in \mathbb{R}$ and $X, Y \sim \mathcal{D}_p$,

$$aX + bY \sim (|a|^p + |b|^p)^{1/p} Z \quad \text{where} \quad Z \sim \mathcal{D}_p \ .$$

Such a distribution $\mathcal{D}_p$ exists for $p \in (0, 2]$.

Consider the problem of estimating p-frequency moments F_p of $\mathbf{x}$ using these projections, where $F_p = \sum_i |x_i|^p$. For $p = 2$, a normal distribution is 2-stable and using the arithmetic mean as estimator, we can get $1 \pm \epsilon$ approximation to F_2 within streaming resource bounds. For $p = 1$, Cauchy random variables are 1-stable. Then, using median as an estimator, [Ind06] obtained $1 \pm \epsilon$ streaming approximation for F_1. Since this pivotal work, other estimators such as sample quantiles, geometric mean and other estimators have been used and analyzed (e.g., [Li08, Li09]), and these have also found other applications such as in estimating Hamming norms [CDIM03] or in privacy-preserving functional estimation of F_p's [MM09], or pan-private streaming [DMW10].

2.2.2 Extension: Measuring Independence

Consider a stream $\langle a_1, \ldots, a_m \rangle$ where $a_k \in [n]^2$ and define random variables X and Y on $[n]$ by

$$\begin{aligned} \Pr[X = i, Y = j] &= |\{k : a_k = (i, j)\}|/m \\ \Pr[X = i] &= |\{k : a_k = (i, \cdot)\}|/m \\ \Pr[Y = j] &= |\{k : a_k = (\cdot, j)\}|/m. \end{aligned}$$

We say X and Y are *empirically independent* if $\Pr[X = i, Y = j] = \Pr[X = i] \Pr[Y = j]$ for all $i, j \in [n]$. Various authors [IM08, BO10, BCL$^+$10] have considered the problem of checking this condition, and more generally estimating how close the condition is to being true. There are numerous ways of quantifying this notion of closeness. For example, one could consider the ℓ_1, ℓ_2, or KL difference between the joint distribution and the product distribution. If any of these quantities are 0 then X and Y are empirically independent. Note that KL divergence between the joint distribution and product distribution is commonly referred to as the *mutual information* between X and Y:

$$I(X; Y) = \sum_{i,j} \Pr[X = i, Y = j] \lg \frac{\Pr[X = i, Y = j]}{\Pr[X = i] \Pr[Y = j]}$$

and this can also be expressed as $H(X) + H(Y) - H(X, Y)$. Hence, an additive approximation is possible using the entropy estimation algorithms from the previous section.

In this section we present the algorithm for estimating the ℓ_2 difference between the joint and product distributions. The algorithm is based on the earlier self-join algorithm of Alon, Matias, and Szegedy [AMS99]. Using the same analysis it can be shown that numerous 4-wise independent vectors $z \in \{-1, 1\}^{n^2}$ could be used to estimate the ℓ_2 difference between two distributions on $[n]^2$. However, for this the elements of z will be the elements of the outer product of two vectors

$x, y \in \{-1, 1\}^n$ which are 4-wise independent. As such, they can be shown to 3-wise independent but not 4-wise independent, e.g.,

$$z_{1,1}z_{2,2} = (x_1^1 x_1^2)(x_2^1 x_2^2) = (x_1^1 x_2^2)(x_2^1 x_1^2) = z_{1,2}z_{2,1} \ .$$

However, by exploiting the geometry of the dependencies, the next lemma establishes that the elements of z are still sufficiently independent.

Exercise 12. *Consider $x^1, x^2 \in \{-1, 1\}^n$ where each vector is 4-wise independent. Let $v \in \mathbb{R}^{n^2}$ and $z_{\mathbf{i}} = x_{i_1}^1 x_{i_2}^2$. Define $\Upsilon = (\sum_{\mathbf{i} \in [n]^2} z_{\mathbf{i}} v_{\mathbf{i}})^2$. Then $\mathbb{E}[\Upsilon] = \sum_{\mathbf{i} \in [n]^2} v_{\mathbf{i}}^2$ and $\mathbb{V}[\Upsilon] \le 9 (\mathbb{E}[\Upsilon])^2$.*

Constructing $\sum_{i,j \in [n]} x_i y_j r_{i,j}$ is simple since the pairs (i, j) arrive together. It turns out the constructing $\sum_{i,j \in [n]} x_i y_j p_i q_j$ is also simple because a sketch of a product of distribution is the product of sketches of the distributions: $\sum_{i,j \in [n]} x_i y_j p_i q_j = (x.p)(y.q)$.

The proof of correctness is given in the next theorem.

Theorem 13. *There exists a single-pass, $\tilde{O}(\epsilon^{-2} \log \delta^{-1})$-space (ϵ, δ) approximation for $\|r - s\|_2$.*

Proof. By appealing to Lemma 12, $\mathbb{E}[\Upsilon] = \sum_{i,j \in [n]} (r_{i,j} - p_i q_j)^2$. By Lemma 12 and the Chebyshev bound, averaging $O(\epsilon^{-2})$ independent Υ returns a $(\epsilon, 1/4)$-approximation. Taking the median of $O(\log \delta^{-1})$ averages returns an (ϵ, δ)-approximation as desired. It remains to be argued that the space requirement is as stated. This follows because there are only $O(\epsilon^{-2} \log \delta^{-1})$ independent estimators and each only requires $O(\log m + \log n)$ space. □

3 Count-Min and Count-Sketch

In this section we present Count-Min and Count-Sketch. The basic functionality of these sketches to support point-queries, e.g., returning an estimate $\tilde{x}_i$ for x_i when queries with $i \in [n]$. But as we shall see, it is possible to build upon this basic functionality and solve a much larger range of problems.

3.1 Count-Min

Pick $d = \log(\delta^{-1})$ hash functions $h_j : [n] \to [w]$ where $w = e/\epsilon$ chosen uniformly at random from a family of pair-wise independent hash functions. We think of $h_j(i)$ as a bucket for i corresponding to the jth hash function. We keep a counter for each bucket, $c_{j,i}$. Initially all buckets are empty, or equivalently, all counters are set to 0. When there is an update (i, Δ), we update $c_{j,i}$ by Δ for all j.

In terms of projection matrices, this is equivalent to $\mathbf{A} \in \{0, 1\}^{wd \times n}$ where for $i \in [w], j \in [d]$:

$$A_{i+w(j-1),k} = \begin{cases} 1 & \text{if } h_j(k) = i \\ 0 & \text{otherwise} \end{cases}$$

This data structure can be used to estimate x_i for any point query i. The result is an estimate for x_i, denoted by $\tilde{x}_i$, where

$$\tilde{x}_i = \min_j c_{j,h_j(i)}.$$

Claim 14. *For simplicity, assume $x_i \ge 0$ for all $i \in [n]$.*

1. $\tilde{x}_i \geq x_i$, *always.*

2. $\tilde{x}_i \leq x_i + \epsilon(F_1 - x_i)$ *with probability at least* $1 - \delta$.

Proof. Let $E = (F_1 - x_i)$. The first part is clear since all $x_i \geq 0$. For the second part, denote by X_{ji} the contribution of items other than i to the $(j, h_j(i))$th bucket. Clearly,

$$\mathbb{E}\left[X_{ji}\right] = \frac{\epsilon}{e} E.$$

Then by Markov's inequality,

$$\Pr\left[\tilde{x}_i > x_i + \epsilon E\right] = \Pr\left[\forall j \; x_i + X_{ji} > x_i + \epsilon E\right] = \Pr\left[\forall j \; X_{ji} > e\mathbb{E}\left[X_{ji}\right]\right] \leq 2^{-\log 1/\delta} = \delta \; .$$

$\square$

Thus, we conclude that we can estimate x_i within an error of $\epsilon(F_1 - x_i)$ with probability at least $1 - \delta$ using $O(\epsilon^{-1} \log \delta^{-1})$ space.

3.2 Count-Sketch

Count-Sketch is similar to Count-Min but in addition to $h_j : [n] \to [w]$, we also use the hash functions $r_j : [n] \to \{-1, 1\}$. As before, we compute the following counts

$$c_{j,k} = \sum_{i : h_j(i) = k} r_j(i) x_i$$

for $j \in [d], k \in [w]$. To estimate x_i we return:

$$\hat{x}_i = \text{median}(r_1(i) c_{1,h_1(i)}, \ldots, r_d(i) c_{d,h_1(i)})$$

Lemma 15. *For any* j, $\mathbb{E}\left[r_j(i) c_{j,h_j(i)}\right] = x_i$ *and* $\mathbb{V}\left[r_j(x) c_{j,h_j(i)}\right] \leq F_2/w$

Proof. Pick an arbitrary $i \in [n]$ and $j \in [d]$. Let $X_k = I[h_j(i) = h_j(k)]$ and so

$$r_j(i) c_{j,h_j(i)} = \sum_k r_j(i) r_j(k) x_k X_k$$

Using the fact that $\mathbb{E}\left[r_j(i) r_j(k)\right] = 0$ for $i \neq k$, we can bound the expectation as:

$$\mathbb{E}\left[r_j(i) c_{j,h_j(i)}\right] = \mathbb{E}\left[x_i + \sum_{k \neq i} r_j(i) r_j(k) x_k X_k\right] = x_i$$

$$\begin{aligned}
\mathbb{V}\left[r_j(i) c_{j,h_j(i)}\right] &\leq \mathbb{E}\left[(\sum_k r_j(i) r_j(k) x_k X_k)^2\right] \\
&= \mathbb{E}\left[\sum_k x_k^2 X_k^2 + \sum_{k \neq \ell} x_k x_\ell r_j(k) r_k(\ell) X_k X_\ell\right] \\
&= F_2/w
\end{aligned}$$

$\square$

By an application of the Chebyshev bound, for $w = 3/\epsilon^2$:

$$\Pr\left[|x_i - r_j(i)c_{j,h_j(i)}| \geq \epsilon\sqrt{F_2}\right] \leq \frac{F_2}{\epsilon^2 w F_2} = 1/3 \ .$$

Therefore by an application of the Chernoff bound, with $d = O(\log \delta^{-1})$ hash functions,

$$\Pr\left[|x_i - \hat{x}_i| \geq \epsilon\sqrt{F_2}\right] \leq 1 - \delta \ .$$

3.3 A Deterministic Variant: CR-Precis

The sketches we have considered so far are randomized. However, we can also consider deterministic sketches. Using a deterministic collection of primes [Mut06, GM07a] devised a data structure called CR-Precis which we now describe. Again, assume $x_i \geq 0$.

For t that will be picked later, let $q_1, \ldots, q_t$ be the first t primes. Hence, $q_t \approx t \ln t$. The algorithm is almost identical to Count-Min except that instead of a random hash function we define:

$$h_j(i) = (i \mod q_j) + 1 \ .$$

As before, we compute $c_{j,k} = \sum_{i:h_j(i)=k} x_k$ and to estimate x_k we use

$$\tilde{x}_i = \min_{j \in [t]} c_{j,h_j(i)} \ .$$

Theorem 16. *For any $i \in [n]$,*

$$x_i \leq \tilde{x}_i \leq x_i + \frac{\log_2 n}{t}(F_1 - x_i)$$

Proof. The first inequality is trivial. For the second one note that for any $k \in [n]$, $k \neq i$, $k \mod q_j = i \mod q_j$ for at most $\log_2 n$ different j's. This is implied by Chinese Remainder Theorem. Hence, at most $\log_2 n$ counters corresponding to i may get incremented as a result of an arrival of k. Since this is true for all $k \neq i$, the counters corresponding to i may get over-counted by at most $\log_2 n \cdot \sum_{k \in [n]:k \neq i} x_k$ in total. On average they get over-counted by at most $\frac{\log_2 n}{t} \sum_{k \in [n]:k \neq i} x_k$, so there must be at least one of the counters corresponding to x that gets over-counted by no more than this number. □

We choose $t = \epsilon^{-1} \log_2 n$. This implies that we will use space $O(t^2 \log t) = O(\frac{\log^2 n}{\epsilon^2} \log \log n)$, where we measure the space in counters of size $O(\log(\sum_i x_i))$.

Open Problem 17. *Design $o(1/\epsilon^2)$ space deterministic streaming algorithm for point queries or show a matching lower bound of $\Omega(1/\epsilon^2)$.*

There are interesting recent approaches to streaming via expanders [Gan08] or codes [PIR10], which while they do not immediately address the problem above, might provide insights.

3.4 Applications: Quantiles, Heavy Hitters, Range Queries

One particularly useful property of linear sketches is the ability to combine them with other linear maps. For example, we can combine a projection matrix $\mathbf{A}$ with another matrix $\mathbf{B}$ and compute $\mathbf{ABx}$. Now, when we see update (i, Δ) we update

$$\mathbf{ABx} \leftarrow \mathbf{ABx} + \Delta \mathbf{ABe}_i \ .$$

In this section, we show how to chose $\mathbf{B}$ such that, given a sketch matrix $\mathbf{A}$ for point-queries, we can support the following queries:

- **Range Queries:** Range queries are a generalization of point-queries. Given query $i, j \in [n]$ we want to estimate:
$$x_{[i,j]} = x_i + x_{i+1} + \ldots + x_j \ .$$
- **Quantiles:** Given $\phi, \epsilon \in (0, 1)$, the problem of determining the *quantiles* is finding $1/\phi$ items $i_0 = 0 \leq i_1 \leq \ldots \leq i_{1/\phi} = n$ such that
$$x_{[1,i_j-1]} < (j\phi + \epsilon)\|\mathbf{x}\|_1 \text{ and } x_{[i_j+1,n]} < (1 - j\phi + \epsilon)\|\mathbf{x}\|_1 \ .$$
Note that when $\epsilon = 0$ and each $x_i \in \{0, 1\}$ this condition implies $x_{[1,i_j]} = j\phi\|\mathbf{x}\|_1$.
- **Heavy Hitters:** Define $S_\tau = \{i \in [n] : x_i \geq \tau\}$. Then given $\phi, \epsilon \in (0, 1)$, the (ϕ, ϵ) Heavy Hitter problem is to find a set S of indices such that:
$$S_\phi \subseteq S \subseteq S_{\phi-\epsilon} \ .$$

In Section 3.6, we will consider $\mathbf{B}$ to be a change of basis matrix such that we can perform point-queries in an alternative basis, e.g., estimating Fourier coefficients or wavelet coefficients.

The above three problems are closely related. Firstly, given the ability to estimate $x_{[i,j]}$, because $x_{[1,\cdot]}$ is monotonic we can perform a binary search on find t such that for a given $j \in [1/\phi]$

$$x_{[1,t-1]} < (j\phi + \epsilon)\|\mathbf{x}\|_1 \text{ and } x_{[t+1,n]} < (1 - j\phi + \epsilon)\|\mathbf{x}\|_1 \ .$$

Secondly, as described in [Mut06], the problems of quantiles and heavy hitters are also closely related. The set of items with relative frequency at least ϵ is a subset of the set of ϵ-quantiles. But more precisely, there is a reduction both ways between the two problems up to $\log n$ factors in space and time [Mut06, Page 22].

Therefore, we will focus on presenting a solution to support range queries. The main idea is to consider dyadic ranges.

Definition 18. *We say a range* $\{i+1, i+2, i+3, \ldots, i+j\}$ *is a* dyadic range *if for some* $k \in [\log_2 n]$, $j = 2^{k-1}$ *and* $2^{k-1} \mid i$.

For example, if $n = 4$ the dyadic ranges are

$$\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{3, 4\}, \text{ and } \{1, 2, 3, 4\} \ .$$

An important property of dyadic ranges is that an arbitrary range can be decomposed into a small number of dyadic ranges.

$$
\begin{pmatrix} x_{[1,8]} \\ x_{[1,4]} \\ x_{[5,8]} \\ x_{[1,2]} \\ x_{[3,4]} \\ x_{[5,6]} \\ x_{[7,8]} \\ x_{[1,1]} \\ x_{[2,2]} \\ x_{[3,3]} \\ x_{[4,4]} \\ x_{[5,5]} \\ x_{[6,6]} \\ x_{[7,7]} \\ x_{[8,8]} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix}
$$

Figure 1: Example of Dyadic-Range Mapping that maps a length-n signal to a length-$(2n-1)$ signal.

Exercise 19. *Show that every range $\{i, i+1, i+2, \ldots, j\}$ can be exactly partitioned into $2\log_2 n$ dyadic ranges.*

Since each dyadic range is a linear combination of some x_i, it is straightforward to define a vector, $\mathbf{x}^D \in \mathbb{R}^{2n-1}$, whose entries correspond to all dyadic ranges as a linear map of $\mathbf{x}$:

$$
\mathbf{x}^D = \mathbf{B}\mathbf{x} .
$$

See Figure 1 for an example when $n = 8$.

Combining $\mathbf{B}$ with a sketch-matrix $\mathbf{A} \in \mathbb{R}^{k\times(2n-1)}$ for point queries allows us to estimate each dyadic range x_i^D. For example, with $\mathbf{A}$ being a Count-Min sketch and $k = O(\epsilon^{-1}\log\delta^{-1})$ we can find an estimate $\tilde{x}_{[i,j]}$ such that with probability $1-\delta$,

$$
x_i^D \leq \tilde{x}_i^D \leq x_i^D + \epsilon\|\mathbf{x}^D\|_1 .
$$

Note that $\|\mathbf{x}^D\|_1 = (\log_2 n)\cdot\|\mathbf{x}\|_1$. Therefore, by decomposing an arbitrary interval $[i,j]$ into dyadic intervals, and estimating the corresponding entry of $\mathbf{x}^D$ we get that with probability $1-\delta(2\log_2 n)$,

$$
x_{[i,j]} \leq \tilde{x}_{[i,j]} \leq x_{[i,j]} + \epsilon\cdot(\log_2 n)\cdot\|\mathbf{x}\|_1 .
$$

Rescaling ϵ and δ gives the following:

Theorem 20. *There is an $O(\epsilon^{-1}\operatorname{polylog} n \log\delta^{-1})$ dimensional sketch that for any $i \leq j \in [n]$ will return an approximation $\tilde{x}_{[i,j]}$ of $x_{[i,j]}$ such that with probability $1-\delta$,*

$$
x_{[i,j]} \leq \tilde{x}_{[i,j]} \leq x_{[i,j]} + \epsilon\cdot(\log_2 n)\cdot\|\mathbf{x}\|_1 .
$$

The sketch also solves (ϕ,ϵ) Heavy Hitters and (ϕ,ϵ) Quantiles.

3.5 Application: Sparse Recovery

The goal of sparse recovery is to find $\mathbf{z}$ such that $\|\mathbf{z}\|_0 \leq k$ and $\|\mathbf{x}-\mathbf{z}\|_p$ is as small as possible. Define $\mathrm{err}_p^k(\mathbf{x}) = \min_{\mathbf{z}:\|\mathbf{z}\|_0\leq k} \|\mathbf{x}-\mathbf{z}\|_p$. It is simple to show that

$$\mathrm{err}_p^k(\mathbf{x}) = \left(\sum_{i\notin S} |x_i|^p\right)^{1/p}$$

where S is the set of indices with the k largest x_i.

We consider the case of $p=2$ and start by revisiting the Count-Sketch analysis. Previously we showed that with Count-Sketch of width $w = 3/\epsilon^2$ and depth $O(\log n)$, we can return estimates $\hat{x}_i$ for each x_i such that with high probability:

$$\forall i \in [n],\ |\hat{x}_i - x_i| \leq \epsilon\sqrt{F_2} = \epsilon\,\mathrm{err}_2^0(x)$$

We can generalize this as follows:

Lemma 21. *Count-Sketch of width $w = \frac{3k}{\epsilon}$ and depth $d = O(\log n)$ suffices to ensure:*

$$\forall i \in [n],\ |\hat{x}_i - x_i| \leq \frac{\epsilon}{\sqrt{k}}\,\mathrm{err}_2^k(\mathbf{x})$$

Proof. Fix a row j of the Count-Sketch data structure. For $i \in [n]$, let $\tilde{x}_i = c_{j,h_j(i)}$ for some row $j \in [d]$. Let $S = \{i_1, \ldots, i_k\}$ be the indices with maximum frequencies. Let A_i be the event that there exists $k \in S \setminus i$, with $h_j(i) = h_j(k)$. Then for $i \in [n]$,

$$\begin{aligned}
\Pr\left[|x_i - \tilde{x}_i| \geq \frac{\epsilon}{\sqrt{k}}\,\mathrm{err}^k(\mathbf{x})\right] &= \Pr[A_i] \times \Pr\left[|x_i - \tilde{x}_i| \geq \frac{\epsilon}{\sqrt{k}}\,\mathrm{err}^k(\mathbf{x}) | A_i\right] \\
&\quad + \Pr[\neg A_i] \times \Pr\left[|x_i - \tilde{x}_i| \geq \frac{\epsilon}{\sqrt{k}}\,\mathrm{err}^k(\mathbf{x}) | \neg A_i\right] \\
&\leq \Pr[A_i] + \Pr\left[|x_i - \tilde{x}_i| \geq \frac{\epsilon}{\sqrt{k}}\,\mathrm{err}^k(\mathbf{x}) | \neg A_i\right] \\
&\leq k/w + 1/3 < 1/2
\end{aligned}$$

Hence, by taking the median estimate over $O(\log n)$ rows we ensure error high probability, all x_i are approximated up to error $\frac{\epsilon}{\sqrt{k}}\,\mathrm{err}^k(\mathbf{x})$ with high probability. □

The sparse recovery result follows because the guarantee in the above lemma is actually stronger than

$$\|\mathbf{x}-\mathbf{z}\|_2 \leq (1+5\epsilon)\,\mathrm{err}_2^k(\mathbf{x})$$

Lemma 22. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ satisfy*

$$\|\mathbf{x}-\mathbf{y}\|_\infty \leq \frac{\epsilon}{\sqrt{k}}\,\mathrm{err}_2^k(\mathbf{x})\ .$$

Then, if T is the set of indices corresponding to the k largest indices of $\mathbf{y}$,

$$\|\mathbf{x}-\mathbf{z}\|_2 \leq (1+5\epsilon)\,\mathrm{err}_2^k(\mathbf{x})$$

where $\mathbf{z} = \mathbf{y}_T$, i.e., the vector whose elements are $\mathbf{z}_i = \mathbf{y}_i$ if $i \in T$ and $\mathbf{z}_i = 0$ otherwise.

Proof. For ease of notation, let $E = \mathrm{err}_2^k(\mathbf{x})$ and let S be the set of indices corresponding to the k largest elements of $\mathbf{x}$. Then

$$\|\mathbf{x}-\mathbf{z}\|_2^2 = \|(\mathbf{x}-\mathbf{z})_T\|_2^2 + \|\mathbf{x}_{S\setminus T}\|_2^2 + \|\mathbf{x}_{[n]\setminus(S\cup T)}\|_2^2$$

since $z_i = 0$ for $i \notin T$. To bound the first term we use the fact that $|T| = k$ and $\|\mathbf{x}-\mathbf{y}\|_\infty^2 \leq \frac{\epsilon^2}{k}E^2$ and so:

$$\|(\mathbf{x}-\mathbf{z})_T\|_2^2 \leq k\frac{\epsilon^2}{k}E^2 = \epsilon^2 E^2 \ .$$

The second term is the most challenging. First note that for $i \in S \setminus T$ and $j \in T \setminus S$ we can write

$$|x_i| - |x_j| \leq |y_i| - |y_j| + 2\sqrt{\frac{\epsilon^2}{k}}E \leq 2\sqrt{\frac{\epsilon^2}{k}}E$$

where $|y_i| - |y_j| \leq 0$ follows since $j \in T$ and $i \notin T$. Therefore, if $a = \max_{i \in S\setminus T} |x_i|$ and $b = \min_{j \in S\setminus T} |x_j|$ we have $a \leq b + 2\sqrt{\frac{\epsilon^2}{k}}E$. Consequently,

$$\|\mathbf{x}_{S\setminus T}\|_2^2 \leq a^2|S\setminus T| \leq \left(b + 2\sqrt{\frac{\epsilon^2}{k}}E\right)^2 |S\setminus T| \leq \left(\frac{\|\mathbf{x}_{T\setminus S}\|_2}{\sqrt{|S\setminus T|}} + 2\sqrt{\frac{\epsilon^2}{k}}E\right)^2 |S\setminus T| \leq (\|\mathbf{x}_{T\setminus S}\|_2 + 2\epsilon E)^2$$

where the second last inequality follows since $\|\mathbf{x}_{T\setminus S}\|_2 \geq b\sqrt{|T \setminus S|} = b\sqrt{|S \setminus T|}$. Furthermore,

$$\begin{aligned}
\|\mathbf{x}_{S\setminus T}\|_2^2 &\leq (\|\mathbf{x}_{T\setminus S}\|_2 + 2\epsilon E)^2 \\
&= \|\mathbf{x}_{T\setminus S}\|_2^2 + 4\epsilon E\|\mathbf{x}_{T\setminus S}\|_2 + 4\epsilon^2 E^2 \\
&\leq \|\mathbf{x}_{T\setminus S}\|_2^2 + 4\epsilon E^2 + 4\epsilon^2 E^2 \\
&\leq \|\mathbf{x}_{T\setminus S}\|_2^2 + 8\epsilon E^2
\end{aligned}$$

Hence,

$$\|\mathbf{x}_{S\setminus T}\|_2^2 + \|\mathbf{x}_{[n]\setminus(S\cup T)}\|_2^2 \leq \|\mathbf{x}_{T\setminus S}\|_2^2 + 8\epsilon E^2 + \|\mathbf{x}_{[n]\setminus(S\cup T)}\|_2^2 = 8\epsilon E^2 + \|\mathbf{x}_{[n]\setminus S}\|_2^2 = (1+8\epsilon)E^2 \ .$$

The result follows since $(1+9\epsilon)^{1/2} \leq 1 + 5\epsilon$.

□

We therefore deduce the following theorem.

Theorem 23. *There is a $O(k\epsilon^{-1} \operatorname{polylog} n)$ dimensional sketch that returns $\mathbf{z}$ such that $\|\mathbf{z}\|_0 \leq k$ and*

$$\|\mathbf{x}-\mathbf{z}\|_2 \leq (1+\epsilon)\,\mathrm{err}_2^k(\mathbf{x}) \ .$$

3.6 Application: Wavelet Decompositions

In the previous section the goal was to find a "simple" approximation for a vector $\mathbf{x} \in \mathbb{R}^n$ where the notion of simple corresponded to having a few non-zero entries. A more general notion of simplicity is the $\mathbf{x}$ can be expressed as the linear combination of only a few basis vectors in some basis. Different bases are relevant in different applications. In this section we consider the Haar wavelets [Haa10] basis although the general algorithmic ideas will apply to arbitrary bases.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{8} & 1/\sqrt{8} & 1/\sqrt{8} & 1/\sqrt{8} & 1/\sqrt{8} & 1/\sqrt{8} & 1/\sqrt{8} & 1/\sqrt{8} \\ 1/\sqrt{8} & 1/\sqrt{8} & 1/\sqrt{8} & 1/\sqrt{8} & -1/\sqrt{8} & -1/\sqrt{8} & -1/\sqrt{8} & -1/\sqrt{8} \\ 1/\sqrt{4} & 1/\sqrt{4} & -1/\sqrt{4} & -1/\sqrt{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\sqrt{4} & 1/\sqrt{4} & -1/\sqrt{4} & -1/\sqrt{4} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix}$$

Figure 2: Example of Change of Basis Mapping that maps a n signal to a length n signal.

Definition 24. *Let n be a power of 2. The Haar basis consists of the vector $(1/\sqrt{n}, 1/\sqrt{n}, \ldots, 1/\sqrt{n})$ and for any $k \in \{1, 2, 4, 8, \ldots, n/2\}, j \in \{1, 2, 3, \ldots, n/(2k)\}$ the vector ψ with entries:*

$$\psi_i = \begin{cases} 1/\sqrt{2k} & \text{if } 2k(j-1) < i \leq 2k(j-1) + k \\ -1/\sqrt{2k} & \text{if } 2k(j-1) + k < i \leq 2kj \\ 0 & \text{otherwise} \end{cases}$$

Denote the Haar basis vectors as $\psi_1, \psi_2, \ldots, \psi_n$.

Exercise 25. *Verify that the above definition gives rise to a set of n orthonormal basis.*

Wavelets can be used to represent signals. Any signal $\mathbf{x}$ is exactly recoverable using the wavelet basis, i.e.,

$$\mathbf{x} = \sum_i \langle \mathbf{x}, \psi_i \rangle \psi_i.$$

We call $y_i = \langle \mathbf{x}, \psi_i \rangle$ the wavelet coefficients and define $\mathbf{B}$ to be the change of basis matrix such that $\mathbf{y} = \mathbf{B}\mathbf{x}$. See Figure 2 for an example when $n = 8$.

Typically, we are not interested in recovering the signal exactly using all the n *wavelet coefficients*; instead, we want to represent the signal using no more than k wavelet coefficients for some $k \ll n$. Say Λ is a set of wavelets of size at most k. Signal $\mathbf{x}$ can be represented as $\tilde{\mathbf{x}}$ using these coefficients as follows:

$$\tilde{\mathbf{x}} = \sum_{i \in \Lambda} \langle \mathbf{x}, \psi_i \rangle \psi_i \ .$$

Clearly $\tilde{\mathbf{x}}$ can only be an approximation of $\mathbf{x}$ in general. The *best k-term representation* (aka *wavelet synopsis*) of $\mathbf{x}$ is the choice of Λ that minimizes the error the sum-squared-error $\|\mathbf{x} - \tilde{\mathbf{x}}\|_2$. Define $\mathrm{err}^k_{p,Haar}(\mathbf{x}) = \min_{\mathbf{z}:\|\mathbf{B}\mathbf{z}\|_0 \leq k} \|\mathbf{x} - \mathbf{z}\|_p$. Because $\mathbf{B}$ is a unitary transformation,

$$\min_{\mathbf{z}:\|\mathbf{B}\mathbf{z}\|_0 \leq k} \|\mathbf{x} - \mathbf{z}\|_2^2 = \min_{\mathbf{z}:\|\mathbf{B}\mathbf{z}\|_0 \leq k} \|\mathbf{B}\mathbf{x} - \mathbf{B}\mathbf{z}\|_2^2 = \min_{\mathbf{z}:\|\mathbf{z}\|_0 \leq k} \|\mathbf{y} - \mathbf{z}\|_2^2 \ .$$

and therefore,

$$\mathrm{err}^k_{2,\mathrm{Haar}}(\mathbf{x}) = \left(\sum_{i \notin S} |y_i|^2 \right)^{1/2}$$

where S is the set of indices with the k largest $y_i = \langle \mathbf{x}, \psi_i \rangle$ values. Therefore the problem can be solved via sparse-recovery.

Theorem 26. *There is a $O(k\epsilon^{-1} \operatorname{polylog} n)$ dimensional sketch that returns $\mathbf{z}$ such that $\|\mathbf{z}\|_0 \leq k$ and*

$$\|\mathbf{x} - \mathbf{z}\|_2 \leq (1+\epsilon) \operatorname{err}^k_{2,Haar}(\mathbf{x}) .$$

4 Sampling via Linear Sketches

In this section we introduce ℓ_p sampling. Here the goal is to return a random tuple $(I, R) \in [n] \times \mathbb{R}$ such that:

$$\Pr[I = i] = (1 \pm \epsilon) \frac{|x_i|^p}{F_p(\mathbf{x})}$$

and $R = (1 \pm \epsilon) x_i$.

4.1 ℓ_0 Sampling

An algorithm for ℓ_0 sampling proceeds as follows:

- Maintain $\tilde{F}_0$, an (1 ± 0.1)-approximation to F_0.
- Hash items using $h_j : [n] \to [0, 2^j]$ for $j \in [\log n]$.
- For each j, maintain:
  - $D_j = (1 \pm 0.1)|\{t | h_j(t) = 0\}|$
  - $S_j = \sum_{t, h_j(t)=0} x_t i_t$
  - $C_j = \sum_{t, h_j(t)=0} x_t$
- Let $\ell = 2 + \left\lceil \log \tilde{F}_0 \right\rceil$. If $D_\ell < 2$ then return element $i = S_\ell / C_\ell$ with frequency estimate C_ℓ.

Lemma 27. *At level ℓ there is an* unique *element in the stream that maps to 0 with constant probability.*

Proof. First observe that

$$2F_0 < 4\hat{F}_0 \leq 2^\ell \leq 8\hat{F}_0 < 12F_0$$

and that for any i, $\Pr[h_\ell(i) = 0] = 1/2^\ell$. The probability there exists a unique i such that $h_\ell(i) = 0$,

$$\begin{aligned}
\sum_{i:x_i>0} \Pr[h_\ell(i) = 0 \text{ and } \forall k \neq i,\ h_\ell(k) \neq 0] &= \sum_{i:x_i>0} \Pr[h_j(i) = 0] \Pr[\forall k \neq i,\ h_\ell(k) \neq 0 | h_\ell(i) = 0] \\
&\geq \sum_{i:x_i>0} \Pr[h_\ell(i) = 0] (1 - \sum_{k \neq i} \Pr[h_\ell(k) = 0 | h_\ell(i) = 0]) \\
&= \sum_{i:x_i>0} \Pr[h_\ell(i) = 0] (1 - \sum_{k \neq i} \Pr[h_\ell(k) = 0]) \\
&\geq \sum_{i:x_i>0} \frac{1}{2^\ell} (1 - \frac{F_0}{2^\ell}) = \frac{F_0}{2^\ell} (1 - \frac{F_0}{2^\ell}) \geq \frac{1}{24}
\end{aligned}$$

$\square$

By repeating the algorithm $O(\operatorname{polylog} n)$ times in parallel we show the following result.

Theorem 28. *There exists an $O(\operatorname{polylog} n)$-dimensional sketch for ℓ_0 sampling where $\delta = 1/\operatorname{poly}(n)$.*

4.2 ℓ_2 Sampling

The idea behind ℓ_2 sampling is as follows. We weight x_i by $\gamma_i = \sqrt{1/u_i}$ where $u_i \in_R [0,1]$ to form vector $\mathbf{y}$:

$$\begin{aligned} \mathbf{x} &= (x_1, x_2, \ldots, x_n) \\ \mathbf{y} &= (y_1, y_2, \ldots, y_n) \quad \text{where } y_i = \gamma_i x_i \end{aligned}$$

Suppose we return (i, x_i) if there is a unique i such that $y_i^2 \geq t := F_2(\mathbf{x})/\epsilon$. Then note that

$$\begin{aligned} \Pr\left[y_i^2 \geq t \text{ and } \forall j \neq i : y_j^2 < t\right] &= \Pr\left[y_i^2 \geq t\right] \prod_{j \neq i} \Pr\left[y_j^2 < t\right] \\ &= \Pr\left[u_i \leq x_i^2/t\right] \prod_{j \neq i} \Pr\left[u_j > x_j^2/t\right] \\ &= x_i^2/t \prod_{j \neq i} (1 - x_j^2/t) \end{aligned}$$

which is approximately x_i^2/t because $1 \geq \prod_{j \neq i}(1 - x_j^2/t) \geq 1 - \sum_{j \neq i} x_j^2/t \geq 1 - \epsilon$. Hence, the probability of y_i being larger than the threshold is approximately proportional to x_i^2 and furthermore, the probability that a unique y_i passes the threshold is $\Omega(\epsilon)$. Hence, repeating the process $1/\epsilon$ times ensures that we returns a sample with constant probability.

Of course, it is impossible to calculate all y_i exactly. Instead we will use a Count-Sketch of size $O(w \log n)$ to estimate each y_i such that with high probability, for all i,

$$\tilde{y}_i^2 = y_i^2 \pm F_2(\mathbf{y})/w \ .$$

While intuition is that while the guarantees of Count-Sketch are in terms of additive error, we also have multiplicative guarantees for the large coordinates that pass the threshold. We will also compute multiplicative estimate of $F_2(\mathbf{y})$ such that $F_2(\mathbf{y}) \leq \tilde{F}_2(\mathbf{y}) \leq 2F_2(\mathbf{y})$. For simplicity, we shall assume that we know the exact value of $F_2(\mathbf{x})$. Then we return $(i, \tilde{y}_i/\gamma_i)$ if

1. $\tilde{y}_i^2 \geq t$ and $\tilde{y}_j^2 < t$ for $j \neq i$
2. $\tilde{F}_2(\mathbf{y}) \leq kF_2(\mathbf{x})$ where $k = 12\epsilon^{-1} \ln n + \epsilon^{-2}$.

Note that the second condition ensures that $F_2(\mathbf{y}) \leq kF_2(\mathbf{x})$ and hence if $w = k$, we have $y_i^2 = \tilde{y}_i^2 \pm F_2(\mathbf{x})$. And therefore satisfying the first case implies $y_i^2 = (1 \pm \epsilon)\tilde{y}_i^2$.

We start with a preliminary lemma that we will use to bound the probability the $F_2(\mathbf{y})$ is not significantly larger than $F_2(\mathbf{x})$.

Lemma 29. *With probability at least $1 - \epsilon$, $F_2(\mathbf{y}) \leq 6\epsilon^{-1} \ln n F_2(\mathbf{x})$.*

Proof. For any fixed j, $\Pr\left[u_j \le 1/n^2\right] = 1/n^2$ and hence by the union bound we deduce that the event $L = \{\forall j \in [n]:\ u_j \ge 1/n^2\}$ has probability at least $1 - 1/n$. Therefore

$$\mathbb{E}\left[F_2(\mathbf{y})|L\right] = \sum_i x_i^2 \mathbb{E}\left[1/u_i|L\right] = \sum_i x_i^2 \frac{1}{1-1/n^2}\int_{1/n^2}^{1}\frac{1}{u}du = F_2(\mathbf{x})\frac{2\ln n}{1-1/n^2} \le 3\ln n F_2(\mathbf{x})\ .$$

Hence, by an application of the Markov inequality, $\Pr\left[F_2(\mathbf{y}) \le 6\epsilon^{-1}\ln n F_2(\mathbf{x})|L\right] \ge 1-\epsilon/2$, and therefore $\Pr\left[F_2(\mathbf{y}) \le 6\epsilon^{-1}\ln n F_2(\mathbf{x})\right] \ge (1-\epsilon)\cdot\Pr\left[L\right] \ge (1-\epsilon)$. □

Theorem 30. *Let U_i be the event that there exists a unique i such that $\tilde{y}_i^2 \ge t$ and that $\tilde{F}_2(\mathbf{y}) \le k/2F_2(\mathbf{x})$. Then,* $\Pr\left[U_i\right] = (1 \pm O(\epsilon))x_i^2/t$.

Proof. Define $t' = t/2$ and consider the following events:

$$\begin{aligned} A_i &= \{y_i^2 \ge t' \text{ and } y_j^2 < t' \text{ for } j \ne i\} \\ A_{i,j} &= \{y_i^2 \ge t' \text{ and } y_j^2 \ge t'\} \\ B &= \{F_2(\mathbf{y}) \le k/2\cdot F_2(\mathbf{x})\} \end{aligned}$$

Appealing to the accuracy guarantees of count-sketch, event B implies that $\tilde{F}_2(\mathbf{y}) \le kF_2(\mathbf{x})$. Furthermore, event B and $y_j^2 \le t/2$ implies $\tilde{y}_j^2 \le t$. Hence, $\Pr\left[U_i|B^C\right] = 0$, $\Pr\left[U_i|y_i^2 \le t', B\right] = 0$ and

$$\Pr\left[U_i|A_i \cap B\right] = \Pr\left[\tilde{y}_i^2 \ge t|y_i^2 \ge t'\right] = \frac{1}{2(1\pm\epsilon)}\ .$$

Therefore, $\Pr\left[U_i\right] = \frac{\Pr[A_i\cap B]}{2(1\pm\epsilon)} + \Pr\left[U_i \cap B \cap (\cup_{j\ne i}A_{i,j})\right]$. We next show $\Pr\left[A_i \cap B\right] \approx x_i^2/t'$ as follows:

$$\Pr\left[A_i \cap B\right] \le \Pr\left[y_i^2 \ge t'\right] \le x_i^2/t'$$

and

$$\begin{aligned} \Pr\left[A_i \cap B\right] &\ge \Pr\left[\frac{t'}{\epsilon} \ge y_i^2 \ge t' \text{ and } y_j^2 < t' \text{ for } j \ne i \text{ and } \sum_{j\ne i} y_j^2 < \frac{kF_2(\mathbf{x})}{2} - \frac{t'}{\epsilon}\right] \\ &\ge \Pr\left[\frac{t'}{\epsilon} \ge y_i^2 \ge t'\right]\Pr\left[y_j^2 < t' \text{ for } j \ne i \text{ and } \sum_{j\ne i} y_j^2 < \frac{kF_2(\mathbf{x})}{2} - \frac{t'}{\epsilon}\right] \\ &\ge \frac{(1-\epsilon)^2 x_i^2}{t'} \end{aligned}$$

where the last line follows because

$$\begin{aligned} \Pr\left[y_j^2 < t' \text{ for } j \ne i \text{ and } \sum_{j\ne i} y_j^2 < \frac{kF_2(\mathbf{x})}{2} - \frac{t'}{\epsilon}\right] &\ge \Pr\left[\sum_{j\ne i} y_j^2 < 6\epsilon^{-1}\ln n F_2(\mathbf{x})\right]\prod_{j\ne i}\left(1 - \Pr\left[y_j^2 > t'\right]\right) \\ &\ge (1-\epsilon)^2 \end{aligned}$$

by appealing to Lemma 29. Finally,

$$0 \le \Pr\left[U_i \cap B \cap (\cup_{j\ne i}A_{i,j})\right] \le \Pr\left[\cup_{j\ne i}A_{i,j}\right] \le \Pr\left[y_i^2 \ge t'\right]\sum_{j\ne i}\Pr\left[y_j^2 \ge t'\right] \le \frac{x_i^2}{t'}\sum_j \frac{x_j^2}{t'} = \frac{2\epsilon x_i^2}{t'}$$

where the last line follows because $\sum_j x_j^2 = \epsilon t$. Hence, we conclude that

$$\frac{(1-\epsilon)^2}{2(1+\epsilon)}\frac{x_i^2}{t'} \leq \Pr[U_i] \leq \frac{1}{2(1-\epsilon)}\frac{x_i^2}{t'} + \frac{2\epsilon x_i^2}{t'} ,$$

and therefore $\Pr[U_i] = (1 \pm O(\epsilon))\frac{x_i^2}{t}$ as claimed. $\square$

Probability some value is returned is $\Omega(\sum_i x_i^2/t) = \Omega(\epsilon)$ so repeating $O(\epsilon^{-1}\log\delta^{-1})$ ensures a value is returned with probability $1-\delta$. The total space used by the algorithm is $O(\epsilon^{-3}\log\delta^{-1})$ but this can be improved using a more careful analysis.

4.2.1 Example: Frequency Moments

Earlier we used $\tilde{O}(n^{1-1/k})$ space to (ϵ, δ) approximate $F_k = \sum_i |x_i|^k$ via AMS sampling. However, ℓ_2-sampling gives a simple way to achieve a near-optimal space use.

Algorithm: Let (I, R) be an ℓ_2 sample. Return

$$T = \hat{F}_2 R^{k-2} \qquad \text{where } \hat{F}_2 \text{ is an } e^{\pm\epsilon} \text{ estimate of } F_2$$

Lemma 31. $\mathbb{E}[T] = e^{\pm\epsilon k}F_k$ *and* $0 \leq T \leq F_k n^{1-2/k}$.

Proof.

$$\mathbb{E}[T] = \hat{F}_2 \sum \Pr[I=i]\,(e^{\pm\epsilon}x_i)^{k-2} = e^{\pm\epsilon k}F_2 \sum_{i\in[n]} \frac{x_i^2}{F_2} x_i^{k-2} = e^{\pm\epsilon k}F_k$$

For the second part note that $T \leq F_2 F_\infty^{k-2}$. It remains to prove that $F_2 F_\infty^{k-2}/F_k \leq n^{1-2/k}$ for $k \geq 2$. Without loss of generality we may assume $F_\infty = 1$ since $F_2 F_\infty^{k-2}/F_k$ is invariant to scaling. By an application of Holder's inequality $F_2 \leq F_k^{2/k} n^{1-2/k}$ and hence

$$F_2 F_\infty^{k-2}/F_k \leq F_k^{2/k-1} n^{1-2/k} \leq n^{1-2/k}$$

where the last line follows because $F_k \geq F_\infty^k = 1$. $\square$

Therefore, by an application of the Chernoff bound it suffices to average the results of $O(n^{1-2/k}\epsilon^{-2}\log\delta^{-1})$ copies of the basic estimator.

Theorem 32. *There is a* $\tilde{O}(n^{1-2/k}\epsilon^{-4})$*-dimensional sketch for estimating* F_k *where* $k \geq 2$.

5 Historical Notes and Further Topics

5.1 Historical Notes

Cormode et al. [CGHJ12] provide a good overview of sketches for signals. Gilbert and Indyk [GI10] cover topics in sparse recovery.

Quantiles. The problem of estimating the median of these values, or more generally, the quantiles has enjoyed significant attention particularly in the database community [MRL98, MRL99, GK01, GKMS02, GM06]. Estimating biased quantiles, e.g., the 99-th or 99.9-th percentile, has also been considered [GZ03, CKMS06]. Appropriately enough, sorting and selection were the subject of one of the first streaming papers [MP80].

Counter-Based Algorithms. Numerous counter-based algorithms exist other than Misra-Gries [MG82, FS82]. Examples are Lossy Counting [MM02] and Space Saving [MAA05]. Various extensions of Misra-Gries exist [DLOM02,KSP03,MAA05]. See [CH08] for an overview and a comparison.

Count-Min and Count-Sketch. Count-Min sketch gives similar accuracy guarantees and small space usage for a number of other problems including estimating ℓ_2 norms (in this case, it is similar to Count Sketch [CCFC04] and more efficient than AMS sketch [AMS99]), inner products, heavy hitters, quantiles, histograms, compressed sensing, matrix approximation, and so on. See [CM10] for a wiki of its many extensions and applications. Also, see [CM05b] for an improved analysis for skewed data. The Count-Min sketch is closely related to Bloom filters and a similar sketching technique was proposed by [EV03].

Frequency Moments, Entropy, and ℓ_p Norms. The problem of estimating ℓ_p norms and frequency moments has been extensively studied [AMS99, Woo04, IW05, BGKS06] and was one of the canonical data stream problems that motivated the development of many important techniques. ℓ_∞ is the frequency of the most frequent item and is discussed above. ℓ_0 is the Hamming norm. Estimation of F_1, the length of the stream, using sub-logarithmic space was considered by Morris [Mor78]. There has also been work done on estimating the ℓ_p distance between two streams [Ind06, FS01, FKSV02]. Given the importance of estimating distances between streams, other distance measures have been considered, including the Hamming distance [CDIM03].

Motivated by networking applications [GMT05, WP05, XZB05], there are also numerous results for estimating the empirical entropy of a sequence of m items in sublinear space [CDM06, GMV06, LSO$^+$06, BG06, CCM07, HNO08] including sketch-based algorithms that naturally handle deletions. For example, Harvey et al. [HNO08] reduced the problem to ℓ_p estimation. First they used the relationship between Shannon entropy and other forms of entropy

$$\text{Renyi entropy:} \quad H_\alpha = \tfrac{\log \|x\|_\alpha^\alpha}{1-\alpha} \tag{1}$$

$$\text{Tsallis entropy:} \quad T_\alpha = \tfrac{1-\|x\|_\alpha^\alpha}{\alpha-1} \tag{2}$$

and used the fact that $H = \lim_{\alpha\to 1} H_\alpha = \lim_{\alpha\to 1} T_\alpha$. The approach in [HNO08] is to evaluate T_α at a few values of α and extrapolate from it to estimate that at $\alpha = 1$.

5.2 Cascaded Aggregates

There is a rich class of difficult problems that arise from "cascading" the computation of one aggregate say P_g for the set of items in a group g, and computing a different aggregate say Q over the results P_g's for different g's.

Example 33 (Multigraph Moments)**.** *Say the stream consists of edges of a multigraph and hence, multiple edges between a pair of vertices will occur several times over the stream. Define the degree d_i of node i to be number of* distinct *neighbors i, that is, not counting the multiplicity of edges between a pair of vertices. Then, the* multigraph moment $M_2 = \sum_i d_i^2$. *M_2 estimation can be thought of as a* cascaded *computation $F_2(F_0)$ where F_0 is applied to each node i and F_2 is applied on the resulting sums.*

Q	P	Upper Bound	Lower bound
F_k	F_0	$O(\epsilon^{-4} n^{1-1/k} \log n)$ [CM05a]	$\Omega(n^{1-1/k})$ [JW09]
ℓ_p, $0 \leq p \leq 1$	ℓ_p, $0 \leq p \leq 2$	$O(1/\epsilon^2)$ [?]	
ℓ_k	ℓ_p, $k \geq p \geq 2$	$O(n^{1-2/k} d^{1-2/p})$ [JW09]	
Heavy hitters			
quantiles	F_0	$\mathrm{poly}(1/\epsilon, \log n)$ [CM05a]	
F_1	F_k	$\mathrm{poly}(1/\epsilon, \log(1/\delta))$ [CGK+09]	$\pm\epsilon$ w.p $1-\delta$
F_k, $k \geq 1$	F_p, $p \in [0, 2]$		$\Omega(n^{1-1/k})$ [MW10]

Table 1: Cascaded Aggregates

Of interest are arbitrary cascaded computations $P(Q)$ for different norms P and Q; several open problems remain in understanding the full complexity of $P(Q)$ cascaded computations. Let domain of P be of size n and domain of Q be of size d.

Study of cascaded aggregates was initiated in [CM05a], but now we know a lot about various special cases. We summarize what is known (in terms of space used, some polylog $n, 1/\epsilon$ terms omitted) and open problems via this table.

5.3 Information Divergences

Given two probability distributions $p = (p_1, p_2, ..., p_n)$ and $q = (q_1, q_2, ..., q_n)$ there are many notions of the "distance" between p and q other than the ℓ_p norm of $p-q$. In particular, in many applications the relative change of the mass at a coordinate is

1. $Hellinger(p, q) = \sum_i (\sqrt{p_i} - \sqrt{q_i})^2$
2. $\Delta(p, q) = \sum_i \frac{(p_i - q_i)^2}{p_i + q_i}$
3. $JS(p, q) = KL(p, (p+q)/2) + KL(q, (p+q)/2) = \sum_i \left(p_i \ln \frac{2p_i}{p_i + q_i} + q_i \ln \frac{2p_i}{p_i + q_i} \right)$

These all come from the f-divergence family $\sum_i p_i f(q_i/p_i)$ where f is convex and $f(1) = 0$. We assume that the precision of each p_i and q_i is polynomial in n

We consider the following models:

1. *Aggregate Model:* Alice knows p and Bob knows q.
2. *Update Model:* Alice has $2n$ non-negative values $(p_1^a, p_2^a, ..., p_n^a, q_1^a, q_2^a, ..., q_n^a)$ and Bob has $2n$ non-negative values $(p_1^b, p_2^b, ..., p_n^b, q_1^b, q_2^b, ..., q_n^b)$ such that $p_i = p_i^a + p_i^b$ and $q_i = q_i^a + q_i^b$.

Note that the aggregate model is a special case of the update model.

It is known that constant factor approximation to the Hellinger divergence, Δ, or JS requires $\Omega(n)$ communication in the (multi-round) update model Guha et al. [GIM07]. The Hellinger divergence can be $(1+\epsilon)$-approximated in the aggregate model with $\mathrm{poly}(\log n, \log \delta^{-1}, \epsilon^{-1})$ communication because of its relationship to ℓ_2. Because Δ and JS are constant factor related to the Hellinger divergence, there exists constant factor approximations for them in the aggregate model using $\mathrm{poly}(\log n, \log \delta^{-1})$ communication.

5.4 Other Representations

There are a number of variations of wavelet representations of interest. For example, one may wish to minimize not ℓ_2 but ℓ_1 and other errors. Certain approximation algorithms are shown for this problem in [GH06]. Sometimes there is a weight associated with each $i \in [1, n]$, and one wishes to minimize weighted norms. Some approximations are in [Mut05].

Open Problem 34. *Design streaming algorithms in presence of increments and decrements for approximate wavelet representation for ℓ_p or weighted ℓ_p errors.*

Other research on histograms and wavelet decompositions include [GKMS01, GGI$^+$02, GIMS02, CGL$^+$05, GKS06, GH06]. A slightly different problem is to learn the probability density function from independent samples given that the probability density function of a k-bucket histogram. This was considered in [CK06, GM07b]. Problems related to finding succinct representation of matrices have been tackled. These are mainly based on sampling rows and columns, an idea first explored in [FKV04] where the goal was to compute the best low-rank approximation of a matrix. A related multiple-pass algorithm was given by [DRVW06]. Other papers use similar ideas to compute a single value decomposition of a matrix [DFK$^+$04], approximation matrix multiplication [DKM06a], succinct representations [DKM06b] and approximate linear programming [DKM06c].

6 Problems

Question 1. *In ℓ_2-sampling the goal is to return a random value $I \in_R [n]$ such that $\Pr[I = i] = (1 \pm \epsilon) f_i^2 / F_2$. Design an simple, small-space stream algorithm for ℓ_2-sampling that takes $O(\log n)$ passes over the data stream. Hint: You can use an F_2 approximation algorithm as a subroutine.*

Question 2. *Prove that for any $1 \le i \le j \le n$, the interval $[i, i+1, \ldots, j]$ can be partitioned into at most $2 \log_2 n$ intervals of the form $[1 + k2^l, 2 + k2^l, \ldots, (k+1)2^l]$ where $k, l \in \mathbb{N}_0$. You may assume n is a power of 2.*

Question 3. *Suppose you may assume that there are at most k values of i such that $f_i > 0$. Adapt the CR-Precis sketch to find all (i, x_i) pairs where $x_i > 0$. Extension to tail.*

Question 4. *How would you adapt to the Count-Min sketch when frequencies can be negative?*

Question 5. *Show how to emulate Count-Sketch sketch with a Count-Min Sketch if you use 4-wise independent hash functions.*

Question 6. *How would you extend reservoir sampling to achieve ℓ_1 sampling on the assumption that every $\Delta > 0$.*

Question 7. *Consider a stream of $n+1$ numbers where each number is in the set $[n]$. Design a small space algorithm that returns an element that occurs twice in the stream.* **Hint:** *Use ℓ_1 sampling and consider the vector $\mathbf{y} = (x_1 - 1, x_2 - 1, \ldots, x_n - 1)$ where x_i is the number of occurrences of i.*

Question 8. *Consider a stream that consists of the m (distinct) edges of a graph on n nodes. Let T be the number of triangles in the graph. Design a small space algorithm that approximate T up to additive error ϵmn.* **Hint:** *Use ℓ_0 sampling on some vector g of length $\binom{n}{3}$.*

Question 9. *Design an algorithm for estimating $F_2(\mathbf{x})$ based on Count-Sketch.* **Hint:** *Consider summing the squares of the entries of a row of the Count-Sketch table. What's the expectation and variance?*

Question 10. *Prove that the Cauchy distribution is 1-stable. Something about sampling from a p-stable distribution.*

Question 11. *Design a sketch-based algorithm for estimating entropy by combining ℓ_1 sampling with the algorithm from Section 1.3.2.*

Question 12. *Let $\mathcal{A}$ be a stream algorithm that returns the median of a* sorted *list of m values in the range $[n]$ with probability $9/10$. If m is not known in advance, prove that $\mathcal{A}$ must use $\Omega(n)$ memory.*

Question 13. *Modify the F_0 algorithm given in class such that instead of estimating the number of non-zero entries, it estimates the number of odd frequencies.*

References

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[BCL+10] Vladimir Braverman, Kai-Min Chung, Zhenming Liu, Michael Mitzenmacher, and Rafail Ostrovsky. Ams without 4-wise independence on product domains. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 119–130, 2010.

[BG06] Lakshminath Bhuvanagiri and Sumit Ganguly. Estimating entropy over data streams. In *ESA*, pages 148–159, 2006.

[BGKS06] Lakshminath Bhuvanagiri, Sumit Ganguly, Deepanjan Kesh, and Chandan Saha. Simpler algorithm for estimating frequency moments of data streams. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 708–713, 2006.

[BO10] Vladimir Braverman and Rafail Ostrovsky. Measuring independence of datasets. In *ACM Symposium on Theory of Computing*, pages 271–280, 2010.

[BYJK+02] Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *Proc. 6th International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 1–10, 2002.

[CCFC04] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004.

[CCM07] Amit Chakrabarti, Graham Cormode, and Andrew McGregor. A near-optimal algorithm for computing the entropy of a stream. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 328–335, 2007.

[CDIM03] Graham Cormode, Mayur Datar, Piotr Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Trans. Knowl. Data Eng.*, 15(3):529–540, 2003.

[CDM06] Amit Chakrabarti, Khanh Do Ba, and S. Muthukrishnan. Estimating entropy and entropy norm on data streams. In *STACS*, pages 196–205, 2006.

[CGHJ12] Graham Cormode, Minos N. Garofalakis, Peter J. Haas, and Chris Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1-3):1–294, 2012.

[CGK+09] Graham Cormode, Lukasz Golab, Flip Korn, Andrew McGregor, Divesh Srivastava, and Xi Zhang. Estimating the confidence of conditional functional dependencies. In *ACM International Conference on Management of Data*, pages 469–482, 2009.

[CGL+05] A. Robert Calderbank, Anna C. Gilbert, Kirill Levchenko, S. Muthukrishnan, and Martin Strauss. Improved range-summable random variable construction algorithms. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 840–849, 2005.

[CH08] Graham Cormode and Marios Hadjieleftheriou. Finding frequent items in data streams. *PVLDB*, 1(2):1530–1541, 2008.

[CK06] Kevin L. Chang and Ravi Kannan. The space complexity of pass-efficient algorithms for clustering. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1157–1166, 2006.

[CKMS06] Graham Cormode, Flip Korn, S. Muthukrishnan, and Divesh Srivastava. Space- and time-efficient deterministic algorithms for biased quantiles over data streams. In *ACM Symposium on Principles of Database Systems*, pages 263–272, 2006.

[CM05a] Graham Cormode and S. Muthukrishnan. Space efficient mining of multigraph streams. In *ACM Symposium on Principles of Database Systems*, pages 271–282, 2005.

[CM05b] Graham Cormode and S. Muthukrishnan. Summarizing and mining skewed data streams. In *SDM*, 2005.

[CM10] G. Cormode and S. Muthukrishnan. Count-min sketch. *https://sites.google.com/site/countminsketch/home*, 2010.

[DFK+04] Petros Drineas, Alan M. Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004.

[DKM06a] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM J. Comput.*, 36(1):132–157, 2006.

[DKM06b] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36(1):158–183, 2006.

[DKM06c] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM J. Comput.*, 36(1):184–206, 2006.

[DLOM02] Erik D. Demaine, Alejandro López-Ortiz, and J. Ian Munro. Frequency estimation of internet packet streams with limited space. In *European Symposium on Algorithms*, pages 348–360, 2002.

[DMW10] A Nikolov D. Mir, S. Muthukrishnan and R. Wright. Pan-private algorithms: when memory does not help. Unpublished manuscript, 2010.

[DRVW06] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *ACM-SIAM Symposium on Discrete Algorithms*, pages 1117–1126, 2006.

[EV03] Cristian Estan and George Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Trans. Comput. Syst.*, 21(3):270–313, 2003.

[FKSV02] Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan. An approximate L^1 difference algorithm for massive data streams. *SIAM Journal on Computing*, 32(1):131–151, 2002.

[FKV04] Alan M. Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004.

[FM85] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209, 1985.

[FS82] M. Fischer and S. Salzberg. Finding a majority among n votes. *Journal of Algorithms*, 3(4):362–380, 1982.

[FS01] Jessica H. Fong and Martin Strauss. An approximate L^p-difference algorithm for massive data streams. *Discrete Mathematics and Theoretical Computer Science*, 4(2):301–322, 2001.

[Gan08] Sumit Ganguly. Data stream algorithms via expander graphs. In *International Symposium on Algorithms and Computation*, pages 52–63, 2008.

[GGI+02] Anna C. Gilbert, Sudipto Guha, Piotr Indyk, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *ACM Symposium on Theory of Computing*, pages 389–398, 2002.

[GH06] Sudipto Guha and Boulos Harb. Approximation algorithms for wavelet transform coding of data streams. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 698–707, 2006.

[GI10] Anna Gilbert and Piotr Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947, 2010.

[GIM07] Sudipto Guha, Piotr Indyk, and Andrew McGregor. Sketching information divergences. In *Conference on Learning Theory*, pages 424–438, 2007.

[GIMS02] Sudipto Guha, Piotr Indyk, S. Muthukrishnan, and Martin Strauss. Histogramming data streams with fast per-item processing. In *International Colloquium on Automata, Languages and Programming*, pages 681–692, 2002.

[GK01] Michael Greenwald and Sanjeev Khanna. Efficient online computation of quantile summaries. In *ACM International Conference on Management of Data*, pages 58–66, 2001.

[GKMS01] Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *International Conference on Very Large Data Bases*, pages 79–88, 2001.

[GKMS02] Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. How to summarize the universe: Dynamic maintenance of quantiles. In *International Conference on Very Large Data Bases*, pages 454–465, 2002.

[GKS06] Sudipto Guha, Nick Koudas, and Kyuseok Shim. Approximation and streaming algorithms for histogram construction problems. *ACM Trans. Database Syst.*, 31(1):396–438, 2006.

[GM06] Sudipto Guha and Andrew McGregor. Approximate quantiles and the order of the stream. In *ACM Symposium on Principles of Database Systems*, pages 273–279, 2006.

[GM07a] Sumit Ganguly and Anirban Majumder. CR-precis: A deterministic summary structure for update data streams. In *ESCAPE*, 2007.

[GM07b] Sudipto Guha and Andrew McGregor. Space-efficient sampling. In *AISTATS*, pages 169–176, 2007.

[GM09] Sudipto Guha and Andrew McGregor. Sketching information divergences in a distributed model. In *Manuscript*, 2009.

[GMT05] Y. Gu, A. McCallum, and D. Towsley. Detecting anomalies in network traffic using maximum entropy estimation. In *Proc. Internet Measurement Conference*, 2005.

[GMV06] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 733–742, 2006.

[GZ03] Anupam Gupta and Francis Zane. Counting inversions in lists. *ACM-SIAM Symposium on Discrete Algorithms*, pages 253–254, 2003.

[Haa10] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. (Erste Mitteilung.) [On the theory of orthogonal function systems (first communication)]. *Math. Ann.*, 69:331–371, 1910.

[HNO08] Nicholas J. A. Harvey, Jelani Nelson, and Krzysztof Onak. Sketching and streaming entropy via approximation theory. In *IEEE Symposium on Foundations of Computer Science*, pages 489–498, 2008.

[IM08] Piotr Indyk and Andrew McGregor. Declaring independence via the sketching of sketches. In *ACM-SIAM Symposium on Discrete Algorithms*, 2008.

[Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006.

[IW03] Piotr Indyk and David P. Woodruff. Tight lower bounds for the distinct elements problem. *IEEE Symposium on Foundations of Computer Science*, pages 283–288, 2003.

[IW05] Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In *ACM Symposium on Theory of Computing*, pages 202–208, 2005.

[JW09] T.S. Jayram and David Woodruff. Cascaded aggregates on data streams. In *Manuscript*, 2009.

[KSP03] Richard M. Karp, Scott Shenker, and Christos H. Papadimitriou. A simple algorithm for finding frequent elements in streams and bags. *ACM Trans. Database Syst.*, 28:51–55, 2003.

[Li08] Ping Li. Estimators and tail bounds for dimension reduction in α (0 < α ≤ 2) using stable random projections. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 10–19, 2008.

[Li09] Ping Li. Compressed counting. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 412–421, 2009.

[LSO+06] Ashwin Lall, Vyas Sekar, Mitsunori Ogihara, Jun Xu, and Hui Zhang. Data streaming algorithms for estimating entropy of network traffic. In *ACM SIGMETRICS*, 2006.

[MAA05] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *ICDT*, pages 398–412, 2005.

[MG82] Jayadev Misra and David Gries. Finding repeated elements. *Sci. Comput. Program.*, 2(2):143–152, 1982.

[MM02] Gurmeet Singh Manku and Rajeev Motwani. Approximate frequency counts over data streams. In *International Conference on Very Large Data Bases*, pages 346–357, 2002.

[MM09] Andre Madeira and S. Muthukrishnan. Functionally private approximations of negligibly-biased estimators. In *FSTTCS*, pages 323–334, 2009.

[Mor78] Robert Morris. Counting large numbers of events in small registers. *CACM*, 21(10):840–842, 1978.

[MP80] J. Ian Munro and Mike Paterson. Selection and sorting with limited storage. *Theor. Comput. Sci.*, 12:315–323, 1980.

[MRL98] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. Approximate medians and other quantiles in one pass and with limited memory. In *ACM International Conference on Management of Data*, pages 426–435, 1998.

[MRL99] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *ACM International Conference on Management of Data*, pages 251–262, 1999.

[Mut05] S. Muthukrishnan. Subquadratic algorithms for workload-aware haar wavelet synopses. In *Proc. FSTTCS*, pages 285–296, 2005.

[Mut06] S. Muthukrishnan. Data streams: Algorithms and applications. *Now Publishers*, 2006.

[MW10] Morteza Monemizadeh and David P. Woodruff. 1-pass relative-error l_p-sampling with applications. In *ACM-SIAM Symposium on Discrete Algorithms*, 2010.

[PIR10] Hung Q. Ngo Piotr Indyk and Atri Rudra. Efficiently decodable non-adaptive group testing. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 20–29, 2010.

[Woo04] David P. Woodruff. Optimal space lower bounds for all frequency moments. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 167–175, 2004.

[WP05] Arno Wagner and Bernhard Plattner. Entropy based worm and anomaly detection in fast IP networks. In *14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WET ICE)*, pages 172–177, 2005.

[XZB05] Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya. Profiling internet backbone traffic: behavior models and applications. In *SIGCOMM*, pages 169–180, 2005.