E0 206: Theorist's Toolkit November 17, 2020

Lecture 14-15: Streaming Algorithms

Instructor: Arindam Khan *Scribe: Aditya V. Singh, Bhargav Thankey*

In these lectures, we will first look at the AMS estimator for estimating F_k (Section 1). Along the way, we will describe reservoir sampling, which is a useful technique for sampling elements from a data stream. For the special case of F_2 estimation, we will look at the Tug-of-War sketch, and present its geometric interpretation. In Section 2, we will define p-stable distributions and look at Indyk's algorithm for estimating F_p, $p \in (0, 2]$. Finally, in Section 3, we will give a brief account of the notion of communication complexity and its application in proving lower bounds for streaming algorithms.

1 AMS Estimator for F_k

Recall from the previous lecture that the k-th frequency moment F_k of a data stream $x_1, \dots, x_m$ coming from a universe $U = \{e_1, \dots, e_n\}$ is defined as

$$F_k = \sum_{i=1}^{n} f_i^k,$$

where $f_i = |\{j : x_j = e_i\}|$ is the number of occurrences of the element e_j in the stream. The goal of this section is to prove the following theorem:

Theorem 1. *We can estimate F_k with $(1 \pm \varepsilon)$ multiplicative error with probability at least $(1 - \delta)$ using $O\left(\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right) n^{1-\frac{1}{k}}\right) \cdot (\log m + \log n)$ space.*

We will look at an explicit estimator – the AMS estimator [AMS99] – for F_k which achieves the guarantees in Theorem 1.

Intuition. Suppose we could sample an element X uniformly at random from the stream. Say $X = e_i \in U$. Then X is equally likely to be any of the f_i occurrences of e_i. If r is the number of occurrences of e_i 'from now on', then r is equally likely to be in $\{1, \dots, f_i\}$. Therefore,

$$\mathbb{E}\left[r^k | X = e_i\right] = \frac{1}{f_i} \sum_{j=1}^{f_i} j^k$$

$$\implies \mathbb{E}\left[r^k - (r-1)^k | X = e_i\right] = \frac{1}{f_i} \left(\sum_{j=1}^{f_i} j^k - \sum_{j=0}^{f_i - 1} j^k \right) = \frac{1}{f_i} \cdot f_i^k.$$

Hence, after removing conditioning,

$$\begin{aligned} &\mathbb{E}\left[r^k - (r-1)^k\right] \\ =&\sum_{i=1}^{n} \mathbb{E}\left[r^k - (r-1)^k | X = e_i\right] \Pr\left[X = e_i\right] \\ =&\sum_{i=1}^{n} \frac{1}{f_i} \cdot f_i^k \cdot \frac{f_i}{m} \\ =&\frac{1}{m} \cdot F_k. \end{aligned} \tag{1}$$

This implies that $m \cdot \left(r^k - (r-1)^k\right)$ is an unbiased estimator of F_k. To turn this intuition into a streaming algorithm, we need two things:

1. We need an algorithm that can sample an element from a stream (of length m) with probability $1/m$ *without knowing m apriori*. We will see an algorithm called *Reservoir Sampling* that achieves this.

2. We need to show concentration results because the value returned by the estimator is equal to F_k only in expectation. We will show this by computing the variance of the estimator and using Chebyshev's inequality.

1.1 Reservoir Sampling

Reservoir Sampling (Algorithm 1) is an answer to the following question:

How can we sample k elements uniformly at random (without replacement) from a stream of length m ($m \gg k$), where m is unknown?

```
Algorithm 1: Reservoir Sampling
  Input: $k \in \mathbb{Z}$, data stream: $x_1, x_2, \ldots$
1 while not end-of-stream do
2     Put the first $k$ elements of the stream into a 'reservoir' $R = \{x_1, \ldots, x_k\}$.
3     for $i \geq k+1$ do
4         With probability $k/i$, replace a random entry of $R$ with $x_i$.
5     end
6 end
7 Return $R$.
```

Claim 2. *For every $t \geq k+1$ and $i \leq t$, $\Pr[x_i \in R_t] = k/t$, where R_t denotes the 'reservoir' after observing $x_1, \ldots, x_t$.*

Proof. We prove the claim using induction on t.

Base case: $t = k+1$. For $i \leq k$, x_i will not be in R_t iff the $(k+1)$-th element is selected for inclusion in R_t *and* the i-th element is chosen to be replaced. That is

$$\Pr[x_i \in R_t] = 1 - \underbrace{\left(\frac{k}{k+1} \cdot \frac{1}{k}\right)}_{\Pr[x_i \notin R_t]} = \frac{k}{k+1} \quad \text{for } i \leq k.$$

Moreover, x_{k+1} gets included in R_t (where $t = k+1$) iff the $(k+1)$-th element is selected for inclusion in R_t. That is

$$\Pr[x_{k+1} \in R_t] = \frac{k}{k+1}.$$

Induction: We assume that for a given $t \geq k+1$, $\Pr[x_i \in R_t] = k/t$ for every $i \leq t$. We want to show that $\Pr[x_i \in R_{t+1}] = k/(t+1)$ for every $i \leq t+1$. For $i \leq t$,

$$\begin{aligned}\Pr[x_i \in R_{t+1}] &= \Pr[x_i \in R_t] \cdot \Pr[x_i \text{ not replaced at time } t+1 | x_i \in R_t] \\ &= \frac{k}{t} \cdot \left(1 - \left(\frac{k}{t+1} \cdot \frac{1}{k}\right)\right) = \frac{k}{t+1}.\end{aligned}$$

For $(t+1)$-th element,

$$\Pr[x_{t+1} \in R_{t+1}] = \frac{k}{t+1}.$$

□

Remark A similar inductive argument shows that for every $t \geq k+1$ and distinct $i_1, \ldots, i_k \leq t$, $\Pr[x_{i_1}, \ldots, x_{i_k} \in R_t] = 1/\binom{t}{k}$. Even though we have presented Algorithm 1 for general k, we only need it for the case when $k = 1$ for the purposes of this lecture.

1.2 Estimating F_k

Now that we have an algorithm that can sample an element from the stream uniformly at random without knowing the length of the stream apriori, we are ready to describe the algorithm for F_k estimation.

Algorithm 2: Estimating F_k

```
Input: Data stream: $x_1, x_2, \ldots$
$X \leftarrow x_1$, $r \leftarrow 1$, $i \leftarrow 1$, select $\leftarrow 0$;
while not end-of-stream do
    $i \leftarrow i + 1$, select $\leftarrow 0$;
    With probability $1/i$, set select $\leftarrow 1$ (Reservoir Sampling).
    if select $= 1$ then
        $X \leftarrow x_i$, $r \leftarrow 1$;
    else
        if $X = x_i$ then
            $r \leftarrow r + 1$;
        end
    end
end
Return $m(r^k - (r-1)^k)$.
```

1.2.1 Analysis

Let the random variable $Y := m(r^k - (r-1)^k)$ denote the output of Algorithm 2. We have already shown (see (1)) that $\mathbb{E}[Y] = F_k$. We now show that Y is concentrated around its mean, i.e., $Y \in [(1-\varepsilon)F_k, (1+\varepsilon)F_k]$ with high probability. For this, we compute the variance of Y and use Chebyshev's inequality.

$$
\begin{aligned}
\operatorname{var}(Y) &\leq \mathbb{E}[Y^2] \\
&= \sum_{i=1}^{n} \mathbb{E}[Y^2 | X = e_i] \cdot \Pr[X = e_i] \\
&= \sum_{i=1}^{n} m^2 \underbrace{\mathbb{E}\left[\left(r^k - (r-1)^k\right)^2 | X = e_i\right]} \cdot \frac{f_i}{m} \\
&= m \sum_{i=1}^{n} f_i \cdot \overbrace{\frac{1}{f_i} \sum_{j=1}^{f_i} \left(j^k - (j-1)^k\right)^2} \\
&\leq m \sum_{i=1}^{n} \sum_{j=1}^{f_i} k \cdot j^{k-1} \cdot \left(j^k - (j-1)^k\right) && \left(\text{since } j^k - (j-1)^k \leq k \cdot j^{k-1}, \text{ see Section 1.2.2 for a proof}\right) \\
&\leq m \cdot k \sum_{i=1}^{n} f_i^{k-1} \sum_{j=1}^{f_i} \left(j^k - (j-1)^k\right)
\end{aligned}
$$

$$\begin{aligned}
&\leq m \cdot k \sum_{i=1}^{n} f_*^{k-1} \cdot f_i^k && \text{where } f_* = \max_{i\in[n]} f_i, \text{ which implies } f_*^{k-1} = \max_{i\in[n]} f_i^{k-1} \\
&\leq m \cdot k \cdot f_*^{k-1} \cdot F_k && \left(\text{since } \sum_{i=1}^{n} f_i^k = F_k\right) \\
&\leq m \cdot k \cdot F_k^{\frac{k-1}{k}} \cdot F_k && \left(\text{since } f_*^{k-1} = \left(f_*^k\right)^{\frac{k-1}{k}} \leq \left(\sum_{i=1}^{n} f_i^k\right)^{\frac{k-1}{k}} = F_k^{\frac{k-1}{k}}\right) \\
&\leq m \cdot k \cdot F_k^{\frac{2k-1}{k}} \\
&\leq n^{1-\frac{1}{k}} \cdot F_k^{\frac{1}{k}} \cdot k \cdot F_k^{\frac{2k-1}{k}} && \left(\text{since } m = F_1 \leq n^{1-\frac{1}{k}} F_k^{\frac{1}{k}}, \text{ see Section 1.2.2 for a proof}\right) \\
&\leq k \cdot n^{1-\frac{1}{k}} \cdot F_k^2 \\
&= k \cdot n^{1-\frac{1}{k}} \cdot \left(\mathbb{E}[Y]\right)^2. && (\text{ since } F_k = \mathbb{E}[Y]) \qquad (2)
\end{aligned}$$

Now, we use Chebyshev's inequality to get

$$\Pr\left[|Y - \mathbb{E}[Y]| \geq \varepsilon \mathbb{E}[Y]\right] \leq \frac{\operatorname{var}(Y)}{\varepsilon^2 \left(\mathbb{E}[Y]\right)^2}$$
$$\implies \Pr\left[|Y - \mathbb{E}[Y]| \geq \varepsilon F_k\right] \leq \frac{k}{\varepsilon^2} \cdot n^{1-\frac{1}{k}}.$$

Note that for a fixed $k > 1$, $\left(\frac{k}{\varepsilon^2} \cdot n^{1-\frac{1}{k}}\right) \gg 1$. To improve this result, we use the **median-of-means trick**. We start with an observation.

Observation 3. *The variance of the estimator can be improved by running t independent copies of Algorithm 2 in parallel to get i.i.d. random variables $Y_1, \ldots, Y_t$, and computing their average Z. Note that*

$$\mathbb{E}[Z] = \mathbb{E}\left[\frac{1}{t}\sum_{i=1}^{t} Y_i\right] = \mathbb{E}[Y_1] = F_k$$

by linearity of expectation and

$$\operatorname{var}(Z) = \frac{1}{t}\operatorname{var}(Y_1).$$

Hence, from Chebyshev's inequality,

$$\Pr\left[|Z - F_k| \geq \varepsilon F_k\right] \leq \frac{k}{t\varepsilon^2} \cdot n^{1-\frac{1}{k}}.$$

To make this probability smaller than δ, we set

$$t = \frac{k}{\delta\varepsilon^2} \cdot n^{1-\frac{1}{k}}.$$

The median-of-means trick allows us to improve the dependence of t on δ; i.e. we can achieve the same concentration with $t = O\left(\frac{k}{\varepsilon^2} \cdot \log\frac{1}{\delta} \cdot n^{1-\frac{1}{k}}\right)$.

The median-of-means trick. For $\ell = 4\log\frac{1}{\delta}$, output

$$Z = \operatorname{median}(Z_1, \ldots, Z_\ell),$$

where $Z_1, \ldots, Z_\ell$ are i.i.d. random variables generated as follows: For $t = \frac{4k}{\varepsilon^2} \cdot n^{1-\frac{1}{k}}$,

$$Z_i = \frac{1}{t}\sum_{j=1}^{t} Y_{ij},$$

where, for each i, $Y_{i1}, \ldots, Y_{it}$ are i.i.d. random variables generated by running t copies of Algorithm 2. Thus, overall, we run $\left(\frac{16k}{\varepsilon^2} \cdot \log \frac{1}{\delta}\right) n^{1-\frac{1}{k}}$ copies of Algorithm 2. Moreover, we get the desired concentration, as shown by the following claim.

Claim 4. $\Pr\left[|Z - F_k| \geq \varepsilon F_k\right] = o(\delta)$.

Proof. For each $i \in [\ell]$, using Chebyshev's inequality (set $\delta = 1/4$ in Observation 3), we get

$$\Pr\left[|Z_i - F_k| \geq \varepsilon F_k\right] \leq \frac{1}{4}.$$

Let E_i be the event $\{|Z_i - F_k| \geq \varepsilon F_k\}$ and let S_i be the corresponding indicator random variable. Then

$$\Pr\left[S_i = 1\right] \leq \frac{1}{4}.$$

Let $S = \sum_{i=1}^{\ell} S_i$. By linearity of expectation,

$$\mathbb{E}[S] \leq \frac{1}{4} \cdot 4 \log \frac{1}{\delta} = \log \frac{1}{\delta}.$$

Now

$$\begin{aligned}
&\Pr\left[S \geq \frac{\ell}{2}\right] \\
=&\Pr\left[S \geq 2 \cdot \frac{\ell}{4}\right] \\
\leq&\Pr\left[S \geq 2\mathbb{E}[S]\right] \quad \left(\text{since } \mathbb{E}[S] \leq \log \frac{1}{\delta} = \frac{\ell}{4}\right) \\
\leq&\left(\frac{e}{2^2}\right)^{\log \frac{1}{\delta}} \quad \left(\text{using Chernoff bound: } \Pr\left[S \geq (1+\alpha)\,\mathbb{E}[S]\right] \leq \left(\frac{e^{\alpha}}{(1+\alpha)^{1+\alpha}}\right)^{\mathbb{E}[S]} \text{ with } \alpha = 1\right) \\
=& \,o(\delta).
\end{aligned}$$

So, with probability at least $1 - \delta$, at least half of Z_i's lie in $(1 \pm \varepsilon)F_k$ (implying that median lies there too). □

With the proof of this claim, we have also proved Theorem 1. We end this section with proof of two inequalities that were used to bound $\mathrm{var}(Y)$.

1.2.2 Proof of two useful inequalities

Inequality 1. For $j, k \geq 1$, we have $j^k - (j-1)^k \leq k \cdot j^{k-1}$.

Proof. We apply the mean value theorem to $f(x) = x^k$. Given $x_1 < x_2$, the mean value theorem (MVT) states that there exists $\theta \in [x_1, x_2]$ such that $f'(\theta) = \left(f(x_2) - f(x_1)\right) / (x_2 - x_1)$. Substituting $f(x) = x^k$, $x_1 = j - 1$, $x_2 = j$, we get

$$f(j) - f(j-1) = f'(\theta) = k\theta^{k-1} \leq kj^{k-1}.$$

□

Remark In general, truncated Taylor series is quite useful in deriving inequalities. Suppose a real-valued function f is infinitely differentiable at a. Then Taylor series expansion of f about a is given by

$$f(x) = f(a) + f'(a)(x-a) + f''(a)\frac{(x-a)^2}{2} + \cdots.$$

Moreover,

- there exists $y \in [a, x]$ such that $f(x) = f(a) + f'(y)(x - a)$ (MVT);
- there exists $z \in [a, x]$ such that $f(x) = f(a) + f'(a)(x - a) + f''(z)\frac{(x-a)^2}{2}$;

  $\cdots$ and so on for higher derivatives.

This is useful to derive inequalities. For instance, take $f(x) = \ln(1 + x)$. Then

$$f'(x) = \frac{1}{1+x},\ f''(x) = -\frac{1}{(1+x)^2}.$$

For any x, $f''(x) < 0$ and thus

$$\begin{aligned} & f(x) \le f(0) + f'(0)x \\ \Longrightarrow\ & \ln(1 + x) \le x \\ \Longrightarrow\ & 1 + x \le e^x. \end{aligned}$$

Inequality 2. $F_1 \le n^{1-\frac{1}{k}} F_k^{\frac{1}{k}}$.

Proof. Applying Jensen's inequality to convex function $f(x) = x^k$, we get

$$\left(\frac{\sum_{i=1}^n f_i}{n}\right)^k \le \frac{\sum_{i=1}^n f_i^k}{n}.$$

That is

$$\left(\frac{F_1}{n}\right)^k \le \frac{F_k}{n} \implies F_1 \le \frac{F_k^{1/k}}{n^{1/k}} \cdot n = n^{1-\frac{1}{k}} F_k^{\frac{1}{k}}.$$

□

1.3 Special case: F_2 Estimation

An important special case of F_k estimation is when $k = 2$. We present a simple streaming algorithm for F_2 estimation, the Tug-of-War Sketch [AMS99] (Algorithm 3). This algorithm improves upon the dependency on n of the space requirement of Algorithm 2 (with $k = 2$) – from $\sqrt{n}$ to $\log n$. Observe that $\log n$ is the minimum space needed to store the index of an element from a universe of cardinality n. In this section, we assume that the universe $U = [n]$.

Algorithm 3: Tug-of-War Sketch

```
  Input: Data stream: $x_1, x_2, \ldots, x_m$.
1 Choose a random hash function $h : [n] \to \{\pm 1\}$ from a 4-universal family.
2 $z \leftarrow 0$;
3 for $i = 1$ to $m$ do
4     $z \leftarrow z + h(a_i)$;
5 end
6 Return $z^2$.
```

1.3.1 Analysis

We will first show that z^2 is an unbiased estimator of F_2. We will then bound the variance of z^2, which will imply concentration results for this estimator.

Computing $\mathbb{E}[z^2]$. We can write z as

$$z = \sum_{s=1}^{n} x_s f_s \tag{3}$$

where, for each $s \in [n]$, $x_s = h(s)$ is a random variable taking values in $\{\pm 1\}$ uniformly. Thus

$$\mathbb{E}\left[\sum_{s=1}^{n} x_s f_s\right] = \sum_{s=1}^{n} f_s \mathbb{E}[x_s] = 0.$$

Now

$$\begin{aligned}
\mathbb{E}[z^2] &= \mathbb{E}\left[\left(\sum_{s=1}^{n} x_s f_s\right)^2\right] \\
&= \mathbb{E}\left[\sum_{s=1}^{n} x_s^2 f_s^2\right] + 2\mathbb{E}\left[\sum_{s\neq t} x_s x_t f_s f_t\right] \\
&= \sum_{s=1}^{n} f_s^2 \mathbb{E}\left[x_s^2\right] + 2\sum_{s\neq t} f_s f_t \mathbb{E}\left[x_s x_t\right] \\
&= \sum_{s=1}^{n} f_s^2,
\end{aligned}$$

where the last equality follows from the fact that $x_s^2 = 1$ with probability 1, and that for $s \neq t$, $\mathbb{E}[x_s x_t] = \mathbb{E}[x_s]\mathbb{E}[x_t] = 0$ (since 4-universality of h implies pairwise independence of x_s, x_t).

Computing $\mathrm{var}(z^2)$. Let $a = z^2$. We want to compute $\mathrm{var}(a)$. We first compute $\mathbb{E}[a^2]$.

$$\begin{aligned}
\mathbb{E}[a^2] &= \mathbb{E}\left[\left(\sum_{s=1}^{n} x_s f_s\right)^4\right] \\
&= \mathbb{E}\left[\sum_{1\le s,t,u,v\le n} x_s x_t x_u x_v f_s f_t f_u f_v\right] \\
&= 6\mathbb{E}\left[\sum_{s=1}^{n}\sum_{t=s+1}^{n} x_s^2 x_t^2 f_s^2 f_t^2\right] + \mathbb{E}\left[\sum_{s=1}^{n} x_s^4 f_s^4\right] && \text{(see Claim 6)} \\
&= 6\sum_{s=1}^{n}\sum_{t=s+1}^{n} f_s^2 f_t^2 \mathbb{E}\left[x_s^2 x_t^2\right] + \sum_{s=1}^{n} f_s^4 \mathbb{E}\left[x_s^4\right] \\
&= 6\sum_{s=1}^{n}\sum_{t=s+1}^{n} f_s^2 f_t^2 + \sum_{s=1}^{n} f_s^4 \\
&\le 3\left(\sum_{s=1}^{n} f_s^4 + 2\sum_{s=1}^{n}\sum_{t=s+1}^{n} f_s^2 f_t^2\right) \\
&= 3\left(\sum_{s=1}^{n} f_s^2\right)^2 && \left(\text{since } \left(\sum_i \alpha_i\right)^2 = \sum_i \alpha_i^2 + 2\sum_{i<j}\alpha_i\alpha_j\right)
\end{aligned}$$

$$= 3\left(\mathbb{E}[a]\right)^2.$$

Thus

$$\begin{aligned}\operatorname{var}(a) &= \mathbb{E}[a^2] - \left(\mathbb{E}[a]\right)^2 \\ &\leq 3\left(\mathbb{E}[a]\right)^2 - \left(\mathbb{E}[a]\right)^2 \\ &= 2\left(\mathbb{E}[a]\right)^2 \\ &= 2F_2^2.\end{aligned} \tag{4}$$

Remark The variance of the output of the F_k estimator with $k = 2$ (call the output a) is $2\sqrt{n}\left(\mathbb{E}[a]\right)^2$ (see equation (2)). Thus, for F_2 estimation, the variance of the output of the Tug-of-War estimator (see equation (4)) is smaller by a factor of $\sqrt{n}$.

Remark Since we have bounded the variance of the Tug-of-War estimator, the remaining calculations to get the desired concentration proceeds similarly as in the analysis of the F_k estimator, using the median-of-means trick. Moreover, since the variance here is smaller by a factor of $\sqrt{n}$, we will get a space saving of $\sqrt{n}$ in comparison to the F_k estimator (with $k = 2$). In particular, for a universal constant c, by running $t = \frac{c}{\varepsilon^2} \cdot \log \frac{1}{\delta}$ copies of Algorithm 3 (note: no dependence on n), we can estimate F_2 to within a multiplicative factor of $(1 \pm \varepsilon)$ with probability at least $(1 - \delta)$.

The following exercise shows an explicit construction of a k-universal hash family.

Exercise 5. *Prove that the family of hash functions $\mathcal{H}$ such that $h(x) = c_{k-1}x^{k-1} + \cdots + c_0$, where $c_i \overset{\text{unif}}{\sim} \{0, \ldots, p-1\}$ (for a prime p), is a k-independent universal hash family.*

Before moving on, we prove a claim that was used to bound the variance of the Tug-of-War estimator.

Claim 6. *Suppose $h : [n] \to \mathcal{X}$ is sampled from a 4-independent universal hash family (a.k.a. 4-universal hash family), such that, for every $i \in [n]$, $\mathbb{E}[h(i)] = 0$. Let $x_i = h(i)$. Then*

$$\mathbb{E}\left[\sum_{1 \leq s,t,u,v \leq n} x_s x_t x_u x_v\right] = 6\mathbb{E}\left[\sum_{s=1}^{n} \sum_{t=s+1}^{n} x_s^2 x_t^2\right] + \mathbb{E}\left[\sum_{s=1}^{n} x_s^4\right].$$

Proof. The summation on the left hand side is over all tuples (s, t, u, v) where $1 \leq s, t, u, v \leq n$. Suppose (s, t, u, v) is such that an index $i \in [n]$ appears only once. Without loss of generality, let $s = i$ and $t, u, v \neq i$. Then, by 4-independence of the hash family

$$\mathbb{E}\left[x_s x_t x_u x_v\right] = \mathbb{E}[x_i]\mathbb{E}[x_t x_u x_v] = 0.$$

Thus, $\mathbb{E}\left[x_s x_t x_u x_v\right] \neq 0$ only for tuples (s, t, u, v) where

- either there is an index $i \in [n]$ such that i appears 4 times in (s, t, u, v), i.e. $s = t = u = v = i$;
- or there are two indices $i, j \in [n], i \neq j$, such that i and j both appear twice in (s, t, u, v). For a given i, j, this can happen in $\binom{4}{2} = 6$ ways (choose 2 variables from s, t, u, v and assign them i; assign the remaining 2 variables the value j).

$\square$

1.3.2 Geometric Interpretation

To build up to the next section, where we study Indyk's algorithm to estimate F_p, $p \in (0,2]$, we give a geometric interpretation of the Tug-of-War sketch for F_2 estimation.
Let $t = 6/\varepsilon^2$, and for $i \in [t]$, let $a_i = y_i^2$ be the output of the i-th copy of Algorithm 3. Then, $\frac{1}{t}\sum_{i=1}^{t} y_i^2$ is an unbiased estimator of F_2. Moreover, by Chebyshev's inequality

$$\begin{aligned}\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t} y_i^2 - F_2\right| \geq \varepsilon F_2\right] &\leq \frac{1}{t}\cdot \underbrace{\frac{\mathrm{var}(a_1)}{\varepsilon^2\left(\mathbb{E}[a_1]\right)^2}}_{\leq 2/\varepsilon^2 \text{ from (4)}} \\ &\leq \frac{1}{3} \quad \left(\text{substituting } t = 6/\varepsilon^2\right).\end{aligned} \tag{5}$$

Let h_i be the random hash function used for the i-th copy of the Tug-of-War sketch, and let y_i^2 be its output. As in equation (3), y_i can be written as $y_i = \sum_{s=1}^{n} h_i(s) f_s$. Overall, for the t copies of the algorithm, we have

$$\underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_t \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} h_1(1) & h_1(2) & \cdots & h_1(n) \\ \vdots & \vdots & \vdots & \vdots \\ h_t(1) & h_t(2) & \cdots & h_t(n) \end{pmatrix}}_{M} \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}}_{f}.$$

Now, equation (5) tells us that, with probability $\geq 2/3$,

$$\left\|\frac{1}{\sqrt{t}} y\right\|_2 \in \left[\sqrt{1-\varepsilon}\,\|f\|_2, \sqrt{1+\varepsilon}\,\|f\|_2\right] \subset \left[(1-\varepsilon)\,\|f\|_2, (1+\varepsilon)\,\|f\|_2\right].$$

That is,

$$\left\|\frac{1}{\sqrt{t}} M f\right\|_2 \in \left[(1-\varepsilon)\,\|f\|_2, (1+\varepsilon)\,\|f\|_2\right] \quad \text{with probability} \geq 2/3.$$

Thus, in totality, running t copies of the Tug-of-War sketch can be interpreted as generating a t-dimensional sketch of the n-dimensional frequency vector f (note that $t \ll n$) using a $t \times n$ random matrix $M/\sqrt{t}$, such that the ℓ_2-norm of the t-dimensional sketch (which the algorithm generates) is within a $(1 \pm \varepsilon)$ multiplicative factor of the ℓ_2-norm of f (which we want to estimate).

The interpretation of the Tug-of-War sketch as "randomly projecting a high-dimensional vector into a low-dimensional space while approximately preserving the ℓ_2-norm with high probability" suggests that we can look for more general random projections (instead of only using hash functions, as in the Tug-of-War sketch).

Dimensionality reduction using random projections would be covered in more detail in future lectures. Here, we give an informal account of an important result in this area called the *Johnson-Lindenstrauss (J-L) lemma.* Intuitively, it says that n points in a high dimensional Euclidean space can be mapped into an $O\left(\log(n)/\varepsilon^2\right)$-dimensional Euclidean space such that the distance between any two points change by at most a multiplicative factor of $(1 \pm \varepsilon)$. More formally,

Theorem (J-L Lemma)**.** *Fix arbitrary* $0 < \varepsilon < 1$ *and* $n, k \in \mathbb{N}$ *satisfying* $k \geq 4\left(\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3}\right)^{-1} \ln n$*. Then, for any set* V *of* n *points in* $\mathbb{R}^d$*, there is a map* $f : \mathbb{R}^d \to \mathbb{R}^k$ *such that, for every* $u, v \in V$

$$(1-\varepsilon)\,\|u-v\|_2^2 \leq \|f(u) - f(v)\|_2^2 \leq (1+\varepsilon)\,\|u-v\|_2^2.$$

Furthermore, such a map can be found in polynomial time.

In fact, guarantees of the J-L lemma can be obtained (with high probability) using a random linear map given by a $k \times d$ matrix whose entries are i.i.d. random variables from standard Gaussian distribution $\mathcal{N}(0,1)$. Note that the sketch-matrix of the Tug-of-War sketch is different in two respects:

1. The entries are uniformly distributed in $\{\pm 1\}$, which is a much simpler distribution than the standard Gaussian.

2. The entries in a row need not be fully independent – 4-wise independence suffices.

However, J-L sketch will give us insights into algorithms for estimating general ℓ_p-norm, $p \in (0,2]$, of the frequency vector. To see how, let us estimate F_2 using J-L sketch.

```
Algorithm 4: Estimating $F_2$ using J-L sketch
  Input: Data stream: $x_1, x_2, \ldots, x_m$.
1 Choose $Y_1, \ldots, Y_n$ independently, each from $\mathcal{N}(0,1)$.
2 $z \leftarrow 0$;
3 for $i = 1$ to $m$ do
4     if $x_i = j$ then
5         $z \leftarrow z + Y_j$;
6     end
7 end
8 Return $z^2$.
```

Let us quickly see why this algorithm works. Note that z can be written as $z = \sum_{j=1}^{n} f_j Y_j$. Since $Y_1, \ldots, Y_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, we have that $z \sim \mathcal{N}(0, \|f\|_2^2)$ (Exercise: verify this). Hence,

$$\mathbb{E}[z^2] = \text{var}(z) + (\mathbb{E}[z])^2 = \|f\|_2^2 + 0 = F_2.$$

An important property of the J-L sketch was that the sketch z, which is a linear combination of i.i.d. Gaussian random variables, is itself a Gaussian random variable with standard deviation F_2. For estimating F_p, $p \in (0,2]$, we generalize this idea in the next Section using *p-stable distributions*, where the sketch – a linear combination of i.i.d. random variables from a p-stable distribution – would itself be a scaled version of the distribution with the scaling factor $= F_p$.

2 Indyk's Algorithm

We start by defining p-stable distributions.

Definition 7 (Stable Distribution [Ind06])**.** *Let $p \in (0,2]$. A distribution $\mathcal{D}$ is said to be p-stable if for any $a_1, \ldots, a_n \in \mathbb{R}$, and independent, identically distributed random variables $X_1, \ldots, X_n$ drawn from $\mathcal{D}$, the distribution of the random variable $\sum_{i=1}^{n} a_i X_i$ is the same as the distribution $\left(\sum_{i=1}^{n} = a_i^p\right)^{1/p} X$, where X is a random variable with distribution $\mathcal{D}$.*

Notice that the Gaussian distribution is 2-stable. In fact, it is known that for every $p \in (0,2]$, there exist p-stable distributions. For $p = 1/2$ and $p = 1$, these are the Lévy distribution and the Cauchy distribution, respectively. However, for values of p other than $1/2, 1, 2$, we do not know closed formulas for the probability density functions of p-stable distributions and they are defined using their characteristic functions: a distribution $\mathcal{D}$ is p-stable if $\mathbb{E}[e^{itX}] = e^{-|t|^p}$.

Although we do not know closed formulas for probability density functions of general p-stable distributions, we can sample from a p-stable distribution for any $p \in (0,2]$ using the Chambers, Mellows, Stuck method

[CMS76] as follows: pick θ uniformly at random from $[-\pi/2, \pi/2]$, pick p uniformly at random from $[0, 1]$ and output

$$\frac{\sin p\theta}{\cos^{1/p}\theta}\left(\frac{\cos(1-p)\theta}{-\ln r}\right)^{(1-p)/p}.$$

2.1 Indyk's algorithm for F_p estimation

Before we give an algorithm for F_p estimation, let us develop some intuition. Our goal is to obtain an (ε, δ) estimate of the F_p norm for $p \in (0, 2]$. We'll have a $k \times n$ matrix M, where $k = O(\varepsilon^{-2}\log(1/\delta))$, whose entries are draw independently from a p-stable distribution $\mathcal{D}_p$. This matrix will map the n-dimensional frequency vector f to a k-dimensional sketch $\mathbf{z}$. The sketch $\mathbf{z}$ will be a linear combination of the columns $M_1, ..., M_n$ of M i.e. $\mathbf{z} = \sum_{i=1}^{n} f_i M_i$. Since, the entries of M are drawn form a p-stable distribution, this means that every coordinate of $\mathbf{z}$ will be a random variable whose distribution is a scaled version of $\mathcal{D}_p$ with scaling factor $\|f\|_p$. We will then "extract" F_p from the sketch.

Algorithm 5: Estimating F_p norm

```
   Input: Data stream: $x_1, \ldots, x_m$.
1  Let $k = O(\varepsilon^{-2}\log(1/\delta))$. Create $k \times n$ matrix $M$ by picking its entries independently from a
     $p$-stable distribution $\mathcal{D}_p$.
2  for $i = 1$ to $k$ do
3      $z_i \leftarrow 0$;
4  end
5  for $\ell = 1$ to $m$ do
6      if $x_\ell = j$ then
7          for $i = 1$ to $k$ do
8              $z_i \leftarrow z_i + M[i, j]$
9          end
10     end
11 end
12 Return $\text{median}_{1\le i\le k}\ \{|z_i|/\text{median}(|\mathcal{D}_p|)\}$.
```

In the above algorithm $|\mathcal{D}_p|$ is the distribution of $Y = |X|$ where $X \sim \mathcal{D}_p$.

2.2 Analysis of the algorithm

Before analysing the algorithm, let us recall the definition of median of a continuous distribution and see how the distribution of the random variable $c|X|$ – for some $c \in \mathbb{R}$ – is related to the distribution of $|X|$.

Let $\mathcal{D}$ be a continuous distribution with PDF ϕ and CDF F_X. Let X be a random variable with distribution $\mathcal{D}$. Then, median of $\mathcal{D}$ is any number $\mu \in \mathbb{R}$ such that

$$\Pr[X \le \mu] = \int_{-\infty}^{\mu} \phi(x)\ dx = \frac{1}{2}.$$

In our case we will have unique medians as the PDFs of the distributions we will work with will be continuous functions.

Let $\mathcal{D}'$ be a distribution of $Y = |X|$ and let its PDF and CDF be ψ and F_Y, respectively. Then,

$$\begin{aligned}
F_Y[y] &= \Pr[Y \le y] \\
&= \Pr[|X| \le y] \\
&= \Pr[-y \le X \le y] \\
&= F_X(y) - F_X(-y).
\end{aligned}$$

Differentiating w.r.t. y on both sides, we get,

$$\psi(y) = \phi(y) - (-\phi(y)) = 2\phi(y).$$

Thus,

$$\psi(y) = \begin{cases} 2\phi(y), & y \ge 0 \\ 0, & y < 0. \end{cases}$$

Now, let $X \sim \mathcal{D}_p$ and for $c \in \mathbb{R}$, let $\phi_{p,c}$ be the PDF of $c|X|$ and $\mu_{p,c}$ its median We make the following claim.

Claim 8.

1. $\phi_{p,c}(x) = \frac{1}{c}\phi_{p,1}(\frac{x}{c})$.
2. $\mu_{p,c} = c\mu_{p,1}$.

Proof. Let $Y = |X|$ and F_Y be its CDF; its PDF is $\phi_{p,1}$. Fix $c \in \mathbb{R}$. Let $Z = c|X|$ and F_Z be its CDF; its PDF if $\phi_{p,c}$. Then,

$$\begin{aligned}
F_Z[z] &= \Pr[Z \le z] \\
&= \Pr[cY \le z] \\
&= \Pr\left[Y \le \frac{z}{c}\right] \\
&= F_Y\left(\frac{z}{c}\right)
\end{aligned}$$

Then, differentiating both sides w.r.t. z, we get

$$\phi_{p,c}(z) = \frac{1}{c}\phi_{p,1}\left(\frac{z}{c}\right),$$

which proves item 1 of the claim. Now, $\mu_{p,c}$ is any number such that

$$\begin{aligned}
\Pr[Z \le \mu_{p,c}] &= \frac{1}{2} \\
\implies \quad F_Z(\mu_{p,c}) &= \frac{1}{2} \\
\implies F_Y\left(\frac{\mu_{p,c}}{c}\right) &= \frac{1}{2}.
\end{aligned}$$

As $F_Y[\mu_{p,1}] = 1/2$, $c\mu_{p,1}$ is a median of Z. $\square$

Armed with Claim 8, let us analyse algorithm 5. Consider the final value of the variables z_i. As the entries of matrix M are independent random variables drawn from $\mathcal{D}_p$, $z_i = \|f\|_p Z$ where $Z \sim \mathcal{D}_p$. Thus the random variable $z_i/\text{median}(|\mathcal{D}_p|)$ has distribution with PDF $\phi_{p,\lambda}$, where $\lambda = \|f\|_p / \text{median}(|\mathcal{D}_p|)$ and its median is

$$\mu_{p,\lambda} = \lambda \mu_{p,1} = \frac{\|f\|_p}{\text{median}(|\mathcal{D}_p|)} \cdot \text{median}(|\mathcal{D}_p|) = \|f\|_p .$$

So, we can say the algorithm is attempting to estimate the median by sampling from the distribution $k = O(\varepsilon^{-2}\log(1/\delta))$ many times and then outputting the sample median. We will now show that the sample median is close to the actual median. As a first step in that direction, we prove the following theorem.

Theorem 9. *Let $\varepsilon > 0$ and let $\mathcal{D}$ be a distribution over $\mathbb{R}$ with PDF ϕ and a unique median $\mu > 0$. Further, suppose that ϕ is continuous on $[(1-\varepsilon)\mu, (1+\varepsilon)\mu]$ and let $\phi_* = \min\{\phi(z) : z \in [(1-\varepsilon)\mu, (1+\varepsilon)\mu]\}$. For $k \in \mathbb{N}$, let $Z_1, .., Z_k$ be independent samples drawn from $\mathcal{D}$ and let $Y = \text{median}_{1\le i\le k} Z_i$. Then,*

$$\Pr[|Y - \mu| > \varepsilon\mu] \le 2 \cdot \exp\left(-\frac{2}{3}\varepsilon^2\mu^2\phi_*^2 k\right).$$

Proof. We will show how to obtain an upper bound on $\Pr[Y < (1-\varepsilon)\mu]$; a similar argument can be used to obtain a lower bound on $\Pr[Y > (1+\varepsilon)\mu]$. Let the CDF of $\mathcal{D}$ be $F(y) = \int_{-\infty}^{y} \phi(z)\, dz$. Then,

$$\begin{aligned} \Pr[Z_i < (1-\varepsilon)\mu] &= \int_{-\infty}^{\mu} \phi(z)\, dz - \int_{-\infty}^{(1-\varepsilon)\mu} \phi(z)\, dz \\ &= \frac{1}{2} - (F(\mu) - F((1-\varepsilon)\mu)) && (\because\ \mu \text{ is the median of } Z_i) \\ &= \frac{1}{2} - \varepsilon\mu\phi(\beta) && (6) \end{aligned}$$

for some $\beta \in [(1-\varepsilon)\mu, \mu]$. To see why the last equality is true, observe that the Fundamental Theorem of Calculus implies $F'(y) = \phi(y)$. As ϕ is continuous on $[(1-\varepsilon)\mu, \mu]$, it also implies that F' is continuous on $[(1-\varepsilon)\mu, \mu]$ and differentiable on $((1-\varepsilon)\mu, \mu)$. Thus, F' satisfies the hypothesis of the Mean Value Theorem on the interval $[(1-\varepsilon)\mu, \mu]$. Applying the Mean Value Theorem, we get that there exists a $\beta \in [\mu, (1-\varepsilon)\mu]$ such that

$$\phi(\beta) = \frac{F(\mu) - F((1-\varepsilon)\mu)}{\mu - (1-\varepsilon)\mu} = \frac{F(\mu) - F((1-\varepsilon)\mu)}{\varepsilon\mu}.$$

Define α to be the number which satisfies

$$\left(\frac{1}{2} - \varepsilon\mu\phi(\beta)\right)(1+\alpha) = \frac{1}{2}. \tag{7}$$

Let $N = |\{i \in [k] : Z_i < (1-\varepsilon)\mu\}|$ and for $i \in [k]$, let $I_i = 1$, if and only if $Z_i < (1-\varepsilon)\mu$. Then,

$$\begin{aligned} \mathbb{E}[N] &= \mathbb{E}\left[\sum_{i\in[k]} I_i\right] \\ &= \sum_{i\in[k]} \mathbb{E}[I_i] && (\because\ \text{linearity of expectation}) \\ &= \sum_{i\in[k]} \Pr[I_i = 1] && (\because\ I_i \text{ are 0-1 variables}) \end{aligned}$$

$$= \left(\frac{1}{2} - \varepsilon\mu\phi(\beta)\right) k \qquad \text{(from 6)} \tag{8}$$

Now, the sample median $Y < (1-\varepsilon)\mu$ if and only if at least half of the Z_i's are less than $(1-\varepsilon)\mu$; in other words $\Pr[Y < (1-\varepsilon)\mu] = \Pr[N \geq k/2]$. Also, from (7) and (8), $(1+\alpha)\,\mathbb{E}[N] = k/2$. Thus,

$$\begin{aligned}
\Pr[Y < (1-\varepsilon)\mu] &= \Pr[N \geq k/2] \\
&= \Pr[N \geq (1+\alpha)\,\mathbb{E}[N]] \\
&\leq \exp\left(-\frac{1}{3}\alpha^2\,\mathbb{E}[N]\right) \qquad \text{(using Chernoff bound)}\,. \quad (9)
\end{aligned}$$

Now, from (7), we have

$$\begin{aligned}
k\left(\frac{1}{2} - \varepsilon\mu\phi(\beta)\right) + k\left(\frac{1}{2} - \varepsilon\mu\phi(\beta)\right)\alpha &= \frac{k}{2} \\
\implies \frac{k}{2} - \varepsilon\mu\phi(\beta)k + \alpha\,\mathbb{E}[N] &= \frac{k}{2} \qquad \text{(from (8))} \\
\implies \alpha\,\mathbb{E}[N] &= \varepsilon\mu\phi(\beta)k.
\end{aligned}$$

Also, from (7),

$$\begin{aligned}
&\left(\frac{1}{2} - \varepsilon\mu\phi(\beta)\right) + \left(\frac{1}{2} - \varepsilon\mu\phi(\beta)\right)\alpha = \frac{1}{2} \\
&\implies -\varepsilon\mu\phi(\beta)k + \frac{1}{2}\alpha \geq 0 \qquad (\because\ \varepsilon\mu\phi(\beta) \geq 0 \text{ since } \varepsilon, \mu > 0 \text{ and } \phi(\beta) \geq 0 \text{ as } \phi \text{ is continuous at } \beta.) \\
&\implies \alpha \geq 2\varepsilon\mu\phi(\beta)k.
\end{aligned}$$

Thus, from (9),

$$\begin{aligned}
\Pr[Y < (1-\varepsilon)\mu] &\leq \exp\left(-\frac{1}{3} \cdot \overbrace{\varepsilon\mu\phi(\beta)k}^{\alpha\,\mathbb{E}[N]} \cdot \overbrace{2\varepsilon\mu\phi(\beta)}^{\alpha}\right) \\
&= \exp\left(-\frac{2}{3}\varepsilon^2\mu^2(\phi(\beta))^2 k\right) \\
&\leq \exp\left(-\frac{2}{3}\varepsilon^2\mu^2\phi_*^2 k\right).
\end{aligned}$$

□

To prove the accuracy of the estimate we will use the above theorem with $\phi = \phi_{p,\lambda}$ and $\mu = \mu_{p,\lambda}$ where $\lambda = \|f\|_p / \mu_{p,1}$. Note that to use the above theorem in any meaningful way, we need an estimate of ϕ_*. Now,

$$\begin{aligned}
\mu\phi_* &= \mu_{p,\lambda} \cdot \min\left\{\phi_{p,\lambda}(z) : z \in [(1-\varepsilon)\mu_{p,\lambda}, (1+\varepsilon)\mu_{p,\lambda}]\right\} \\
&= \lambda\mu_{p,1} \cdot \min\left\{\frac{1}{\lambda}\phi_{p,1}\left(\frac{z}{\lambda}\right) : z \in [(1-\varepsilon)\lambda\mu_{p,1}, (1+\varepsilon)\lambda\mu_{p,1}]\right\} \\
&= \mu_{p,1} \cdot \min\left\{\phi_{p,1}(y) : y \in [(1-\varepsilon)\mu_{p,1}, (1+\varepsilon)\mu_{p,1}]\right\},
\end{aligned}$$

which is a constant only depending on p, say c_p. Thus, from the theorem,

$$\begin{aligned} \Pr\left[|Y-\mu| > \varepsilon\mu\right] &\leq 2\cdot\exp\left(-\frac{2}{3}\varepsilon^2\mu^2\phi_*^2 k\right) \\ \implies \Pr\left[\left|Y-\|f\|_p\right| > \varepsilon\|f\|_p\right] &\leq 2\cdot\exp\left(-\frac{2}{3}\varepsilon^2 c_p^2 k\right) \\ \implies \Pr\left[\left|Y-\|f\|_p\right| > \varepsilon\|f\|_p\right] &\leq \delta, \end{aligned}$$

by putting $k = \frac{3}{2c_p^2}\varepsilon^{-2}\log\frac{2}{\delta}$.

2.3 Some technical details about implementing the algorithm

Algorithm 5 is an idealized sketch and one needs to take care of the following details while implementing the algorithm.

1. The sketch – in particular the matrix M – uses real numbers while in practice computers can only do bounded precision arithmetic. To get around this problem, we can approximate all entries of M with sufficient precision using rational numbers. While we won't prove it, the number of bits required per entry of M is polynomial in $\log n$, $1/\varepsilon$ and $1/\delta$.

2. The matrix M used by the sketch has n columns and no small implicit representation. We can get around storing M by using a pseudorandom generator (PRG, for short) that works with space bounded algorithms (see [Nis92]). A PRG is a deterministic function that takes a "small" truly random string (called a seed) and generates a "large" string which "looks random". Note that we only need the j-th column of M when the j-th token arrives. So, we can use the PRG seeded with j plus the initial seed to generate the j-th column of M when required. While we won't prove it, this only blows up the space required by the algorithm by a $\log n$ factor and adds $1/n$ to the error probability.

3 Communication Complexity Theory and Lower Bounds

Communication complexity theory studies the following fundamental problem:

Consider two players Alice and Bob who have inputs x and y, respectively. They do not know each other's input and want to compute a function $f(x, y)$ which is known to both of them. What is the minimum amount of communication required to compute f?

Not only is communication complexity theory an important and interesting area of complexity theory, it is also very useful in proving lower bounds for many diverse areas like property testing, game theory, data structures, extension complexity, parallel and VLSI computation, and circuit lower bounds, to name just a few. Additionally, unlike some lower bounds that rely on some assumptions like $\mathsf{P} \neq \mathsf{NP}$ or the Exponential Time Hypothesis (ETH), lower bounds obtained using communication complexity are unconditional. In this section we will first introduce the one-way communication setting and then see how it can be used to prove lower bounds on streaming algorithms.

3.1 The setting

As shown in Figure 1, there are two players Alice and Bob with unlimited computational power. Alice has input $x \in \{0,1\}^a$ and Bob has input $y \in \{0,1\}^b$. Neither of them know the input of the other player and their goal is to compute a function $f : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}$, which is known to both. Moreover, they have already agreed upon a protocol to compute f before seeing their respective inputs and all that needs to be done now is to implement the protocol with inputs x and y. In this lecture, we will focus only on

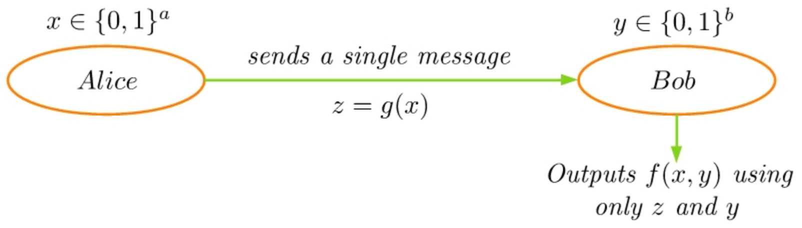

Figure 1: The one-way communication setting.

one-way communication protocols, where only one player (say, Alice) can send a message to the other player (say, Bob), who then has to compute $f(x,y)$ using the message received and their input.

Now we are ready to define the notion of one-way communication complexity (owcc, for short).

Definition 10 (One-way communication complexity or owcc)**.** *Let $f : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}$ be a Boolean function. The one-way communication complexity of f, denoted owcc(f) is the minimum number of bits communicated in the worst case by any one-way communication protocol that correctly computes f on all inputs. For randomized protocols, owcc(f) is the minimum number of bits communicated in the worst case by any one-way communication protocol that correctly computes f on all inputs with probability at least 2/3.*

Remark The probability in the above definition is on the randomness of the protocol; there is no randomness in the input. Also, as long as we are not interested in the exact value of owcc(f), but only its asymptotic behaviour, 2/3 in the above definition can be replaced by any constant in $(1/2, 1)$.

Note that, trivially, owcc$(f) \leq a$: Alice can send x to Bob and Bob can compute $f(x,y)$. On the other hand, consider the parity function defined as $PARITY_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $PARITY_n(x,y) = 1 \iff$ the sum of number of 1s in xy is odd; here xy denotes the string obtained by concatenating x and y. owcc$(PARITY_n) = 1$ as Alice can simply send the parity of x to Bob. Now let us see some "hard" functions for which the trivial protocol of sending x is optimal:

1. The equality function.

$$EQ_N : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}, \qquad EQ_N(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

2. The index function.

$$IDX_N : \{0,1\}^N \times [N] \to \{0,1\}, \qquad IDX_N(x,y) = \begin{cases} 1 & \text{if } x_y = 1, \\ 0 & \text{otherwise.} \end{cases}$$

   where x_y denotes the y-th symbol of x. For example, $IDX_3(101,2) = 0$ and $IDX_3(011,3) = 1$.

3. The disjointness function.

$$DISJ_N : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}, \qquad DISJ_N(x,y) = \begin{cases} 1 & \text{if } \forall i \in [N],\ x_i = 0 \text{ or } y_i = 0, \\ 0 & \text{otherwise} \end{cases}$$

i.e. if $X \subset [N]$ is the set of indices i such that $x_i = 1$ and $Y \subset [N]$ is the set of indices j such that $y_j = 1$, then $DISJ_N(x,y) = 1$ if and only if $X \cap Y = \phi$. For example, $DISJ_3(101,010) = 1$, $DISJ_3(101,100) = 0$ and $DISJ_3(100,010) = 1$.

3.2 Connection to streaming algorithms

Now let us see how we can obtain streaming algorithm lower bounds using communication complexity lower bounds. In what follows we show that *a small space streaming algorithm implies one-way communication protocols with low communication complexity*; we can then use the contrapositive of this statement to prove streaming algorithm lower bounds.

Small space streaming algorithm $\implies$ low communication one-way protocols: Suppose that we have a streaming algorithm $\mathcal{A}_s$ that uses s bits of memory. How can we define a one-way communication protocol using it? Well, we can have Alice and Bob treat their inputs x and y as a stream (x, y) where all of x appears before all of y. Then,

1. Alice feeds x into $\mathcal{A}_s$ without communicating it to Bob.
2. After $\mathcal{A}_s$ has processed x its state can be expressed using at most s bit.
3. Alice communicates these s bits to Bob.
4. Bob restarts $\mathcal{A}_s$ where Alice left off by initializing it with these s bits of memory.

Now, the output computed by Bob will be the same as the output of $\mathcal{A}_s$ on the stream (x, y) and so we have a one-way protocol with low communication complexity! In fact the *communication cost of the induced protocol is the same as the space used by $\mathcal{A}_s$*. This discussion suggests that to prove a lower bound on the space required by any streaming algorithm for some problem $\mathcal{P}$, we need a function f with the following two properties:

1. f can be reduced to $\mathcal{P}$. In other words, f can be computed using a streaming algorithm for $\mathcal{P}$.
2. No low-communication one-way protocol can compute f.

We now show that the disjointness function $DISJ_N$ defined on the previous page satisfies 2. In fact, in communication complexity, it is one of the canonical "hard" functions - analogous to SAT in the theory of NP-completeness.

Proposition 11. *Every deterministic one-way communication protocol that computes $DISJ_N$ uses at least N bits of communication in the worst case. (Thus, the trivial protocol for $DISJ_N$ is an optimal deterministic protocol.)*

Proof. For the sake of contradiction assume that there is a one-way communication protocol in which Alice sends at most $N - 1$ bits. This means that Alice can send at most 2^{N-1} different messages. Then, as there are 2^N distinct strings of size N, by the pigeon hole principle, there must exist $x, x' \in \{0,1\}^N$ such that the message that Alice sends to Bob on input x is the same as that sent on input x'. As $x \neq x'$, $\exists i \in [N]$ such that $x_i \neq x'_i$. Let y be the string that is all 0's except for y_i, which is 1. But, then $DISJ_N(x, y) \neq DISJ_N(x', y)$. However, as Bob cannot distinguish between x and x', he cannot distinguish between $DISJ_N(x, y)$ and $DISJ_N(x', y)$ and must thus output the wrong value for at least one of them, a contradiction. $\square$

While we will not prove it, the following statement, which is much stronger, is true.

Theorem 12. *Every randomized one-way communication protocol that correctly computes $DISJ_N(x, y)$ on all inputs $(x, y) \in \{0,1\}^n \times \{0,1\}^n$ with probability at least $2/3$ requires $\Omega(n)$ bits of communication in the worst case.*

Remark In the above theorem, the probability is over the randomness of the protocol; there is no randomness in the input. Moreover, $2/3$ can be replaced by any constant greater than $1/2$ using the "median trick".

We are now in the position to prove a lower bound for F_∞ estimation.

3.3 Space lower bound for F_∞ estimation

Recall that the F_∞ norm of a stream is the frequency of the most frequent element in it.

Theorem 13. *Any randomized streaming algorithm that computes F_∞ within a factor of $(1 \pm 1/5)$ with probability at least $2/3$ for every data stream of length m over a universe of size n uses space $\Omega(\min\{m, n\})$.*

Proof. Let $\mathcal{A}_s$ be a randomized streaming that uses s bits of space which computes F_∞ within a factor of $(1 \pm 1/5)$ with probability at least $2/3$ for every data stream of length m over a universe of size n. Consider the following one-way communication protocol for the $DISJ_n$.

Protocol 6: Protocol for $DISJ_n$

```
1 Alice feeds $\mathcal{A}_s$ the indices $i$ for which $x_i = 1$ (in any arbitrary order).
2 Alice sends the current memory state $\sigma$ of $\mathcal{A}_s$ to Bob.
3 Bob resumes $\mathcal{A}_s$ with memory state $\sigma$ and feeds it the indices $i$ for which $y_i = 1$ (in any arbitrary
    order).
4 Bob declares 'disjoint' if and only if the final output of $\mathcal{A}_s$ is at most $4/3$.
```

Let us first determine the communication cost of the protocol. Steps 1 and 3 do not incur any cost as Alice has x and Bob has y. The only communication is in Step 2. As $\mathcal{A}_s$ uses at most s bits of memory, σ can be expressed in s bits . So the communication cost of the above protocol is s bits.

Note that the universe of the stream induced by (x, y) is $[n]$ and the size of the stream is $m \leq 2n$ (the size is $2n$ when both x and y are all 1 strings). The frequency f_i of an element $i \in [n]$ is given by

$$f_i = \begin{cases} 0 & \text{if } x_i = y_i = 0, \\ 1 & \text{if exactly one of } x_i \text{ and } y_i \text{ is } 1, \\ 2 & \text{if } x_i = y_i = 1. \end{cases}$$

Hence, $F_\infty = 2$ if and only if $DISJ_n(x, y) = 0$ and $F_\infty \leq 1$ otherwise. Since, for every input (x, y), $\mathcal{A}_s$ outputs a number in $(1 \pm 1/5)F_\infty$ with probability at least $2/3$, if $DISJ_n(x, y) = 1$, the output of $\mathcal{A}_s$ is at most $(1 + 1/5) = 1.2$ and if $DISJ_n(x, y) = 0$, the output of $\mathcal{A}_s$ is at least $2(1 - 1/5) = 1.6$. As $1.2 < 4/3 < 1.6$, Protocol 6 correctly computes $DISJ_n(x, y)$ with probability at least $2/3$. Then, from Theorem 12, its communication cost must be $\Omega(n)$. Hence, $s = \Omega(n)$; furthermore, as $m \leq 2n$, $s = \Omega(m)$ and so $s = \Omega(\min\{m, n\})$. ☐

Exercise 14. *In fact, the lower bound of $s = \Omega(\min\{m, n\})$ holds for estimation of any $F_k, k \neq 1$. Extend the above proof for arbitrary $k \neq 1$.*

References

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy, *The space complexity of approximating the frequency moments*, Journal of Computer and system sciences **58** (1999), no. 1, 137–147.

[CMS76] J. M. Chambers, C. L. Mallows, and B. W. Stuck, *A method for simulating stable random variables*, Journal of the American Statistical Association **71** (1976), no. 354, 340–344.

[Ind06] Piotr Indyk, *Stable distributions, pseudorandom generators, embeddings, and data stream computation*, J. ACM **53** (2006), no. 3, 307–323.

[Nis92] Noam Nisan, *Pseudorandom generators for space-bounded computation*, Combinatorica **12** (1992), no. 4, 449–461.