Lecture 21: April 11

Lecturer: Naveen Garg *Scribe: Sachin Kr Chauhan*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

21.1 Background

In last class, we did the following -

$$\text{Compute } F_k = \sum_{i=1}^{n} f_i^k \text{ of a stream } \sigma \tag{21.0}$$

Using an algorithm with space requirement $\tilde{O}(n^{1-1/k})$
If r is number of occurrences of picked token after the point l
Output for $F_k = m(r^k - (r-1)^k) \; \forall \; k \geq \; 2$
The final Estimator was evaluated as Median-of-Means of the above output.

Shortcomings of above estimator -
The above algorithm works in sub-linear space for estimating k^{th} frequency moment.
Even for $k = 2$, the second frequency moment, it fails to be poly-logarithmic.

Now, we will study an algorithm which allows to estimate F_2 in logarithmic space.

21.2 Tug-of-War Sketch

21.2.1 Basic Algorithm

1. **Initialize:**
   Pick a random hash function from a 4-Universal Family $\mathcal{H}$
   h : [n] → {-1,+1}
   $x = 0$
2. **Process (j, c):**
   $x = x + c.h(j)$
3. **Output:** x^2

The random variable x is pulled in the +ve direction by tokens with +ve h(j) and -ve direction by other tokens, thus the name Tug-of-War Sketch.

21.2.2 Analysis

X is a random variable which denotes the value of x after algorithm has processed σ.

Lets define $Y_j \;=\; h(j) \quad \forall\; j\; \epsilon\; [n]$

Hence, $\qquad X = \sum_{j=1}^{n} f_j Y_j$

4-Universal Hash Family

For hash function h from a 4-Universal Family $\mathcal{H}$

$$Pr_{h \epsilon \mathcal{H}}[Y = 1] = Pr_{h \epsilon \mathcal{H}}[Y = -1] = \frac{1}{2}$$

$\mathcal{H}$ is 4-universal (and hence also 2-universal). This means that for i $\neq$ j,

$$\mathbb{E}[Y_i Y_j] = \mathbb{E}[Y_i]\mathbb{E}[Y_j] \tag{21.1}$$

$$\mathbb{E}[Y_j] = 0 \quad \forall\; j\; \epsilon\; [n] \tag{21.2}$$

$$\mathbb{E}[Y_j^2] = 1 \quad \forall\; j\; \epsilon\; [n] \tag{21.3}$$

For all distinct i,j,k,l,

$$\mathbb{E}[Y_i Y_j Y_k Y_l] = \mathbb{E}[Y_i]\mathbb{E}[Y_j]\mathbb{E}[Y_k]\mathbb{E}[Y_l] = 0 \tag{21.4}$$

When all 4 terms are same,

$$\mathbb{E}[Y_i Y_j Y_k Y_l] = \mathbb{E}[Y_i Y_i Y_i Y_i] \;= \mathbb{E}[Y_i^4] = 1 \tag{21.5}$$

When two terms are appearing twice, i.e. i = k and j = l,

$$\mathbb{E}[Y_i Y_j Y_k Y_l] = \mathbb{E}[Y_i^2]\; \mathbb{E}[Y_j^2] \;= 1 \tag{21.6}$$

Such terms can form 3 combinations - (a,a,b,b), (a,b,a,b) or (a,b,b,a)

Expectation

$$\begin{aligned}
\mathbb{E}[X^2] &= \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} f_i f_j Y_i Y_j\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{n} f_j^2 Y_j^2 \;+\; \sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n} f_i f_j Y_i Y_j\right] \\
&= \sum_{j=1}^{n} f_j^2 \mathbb{E}[Y_j^2] \;+\; \sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n} f_i f_j \mathbb{E}[Y_i]\mathbb{E}[Y_j] && \text{(From 21.1)} \\
&= F_2 \quad \text{(From 21.0, 21.2 and 21.3)} && (21.7)
\end{aligned}$$

Variance

$$\mathbb{E}[X^4] = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_i f_j f_k f_l \mathbb{E}[Y_i Y_j Y_k Y_l]$$

Any index term appearing in quadruple (21.5) or pair (21.6) will survive, others will be 0 (by 21.2, 21.4)

$$\begin{aligned}
\mathbb{E}[X^4] &= 1 * (QuadrupleTerms) + 3 * (PairTerms) + C * (SingleTerms) \\
&= \sum_{i=1}^{n} f_i^4 \ \mathbb{E}[Y_i^4] + 3\sum_{i=1}^{n}\sum_{\substack{j=1 \\ i\neq j}}^{n} f_i^2 f_j^2 \ \mathbb{E}[Y_i^2]\mathbb{E}[Y_j^2] + C\sum_{i=1}^{n}\sum_{\substack{j=1 \\ j\neq i}}^{n}\sum_{\substack{k=1 \\ k\neq ij}}^{n}\sum_{\substack{l=1 \\ l\neq ijk}}^{n} f_i f_j f_k f_l \mathbb{E}[Y_i]\mathbb{E}[Y_j Y_k Y_l] \\
&= \sum_{i=1}^{n} f_i^4 + 3\sum_{i=1}^{n}\sum_{\substack{j=1 \\ i\neq j}}^{n} f_i^2 f_j^2 + 0 && \text{(From 21.2 to 21.6)} \\
&= \sum_{i=1}^{n} f_i^4 + 3\left(\left(\sum_{i=1}^{n} f_j^2\right)^2 \ - \ \sum_{j=1}^{n} f_j^4\right) \\
&= F_4 + 3(F_2^2 - F_4) && \text{(From 21.0)} \\
&= 3F_2^2 - 2F_4 \\
Var[X^2] &= \mathbb{E}[X^4] - \Big(\mathbb{E}[X^2]\Big)^2 \\
&= 3F_2^2 - 2F_4 - F_2^2 && \text{(From 21.7)} \\
&= 2F_2^2 - 2F_4
\end{aligned}$$

$$Var[X^2] \leq 2F_2^2 \tag{21.8}$$

Bounds

By Chebychev's Inequality: For any fixed positive k,

$$Pr[|X^2 - \mathbb{E}[X^2]| > k] \leq \frac{Var[X^2]}{k^2}$$

For k = $\epsilon\mathbb{E}[X^2]$

$$\begin{aligned}
Pr[|X^2 - \mathbb{E}[X^2]| > \epsilon\mathbb{E}[X^2]] &\leq \frac{Var[X^2]}{\epsilon^2\big(\mathbb{E}[X^2]\big)^2} \\
Pr[|X^2 - F_2| > \epsilon F_2] &\leq \frac{2F_2^2}{\epsilon^2 F_2^2} && \text{(From 21.7 and 21.8)} \\
&= \frac{2}{\epsilon^2}
\end{aligned}$$

Pr[Estimate is between $(1-\epsilon)F_2$ and $(1+\epsilon)F_2$] $\geq 1 - \frac{2}{\epsilon^2}$

Mean Trick

We will first apply the mean trick to reduce the Variance.
We will evaluate t process with independent hash functions and consider their mean as the output.
This effectively reduces the Variance to $2F_2^2/t$

$$
\begin{aligned}
Mean_i &= \frac{1}{t}\sum_{j=1}^{t} X_{ij}^2 \\
Pr[|Mean_i - F_2| > \epsilon F_2] &\leq \frac{Var[Mean_i]}{\epsilon^2 F_2^2} && \text{(By Chebychev's Inequality)} \\
&= \frac{Var[X^2]}{t\epsilon^2 F_2^2} \\
&= \frac{2F_2^2}{t\epsilon^2 F_2^2} \\
&= \frac{2}{t\epsilon^2}
\end{aligned}
$$

$$
For\ t = \frac{6}{\epsilon^2},\quad Pr[|Mean_i - F_2| > \epsilon F_2] \ \leq \ \frac{1}{3} \tag{21.9}
$$

Pr[Mean is between $(1-\epsilon)F_2$ and $(1+\epsilon)F_2$] $\geq 1 - \frac{1}{3} = \frac{2}{3}$

Median Trick

Then, we apply the median trick to improve the confidence in the output.
We take k $Mean_i$'s and take the median of those values.

$$
\begin{aligned}
\text{The } Estimate &= \underset{1\leq i\leq k}{median}\ Mean_i \\
\text{Let } Z &= Z_1 \ + \ Z_2 \ + \ Z_3 \ + \ ... \ + \ Z_k \\
\text{where } Z_i &= \begin{cases} 1 & \text{with probability } |Mean_i - F_2| > \epsilon F_2 \\ 0 & \text{otherwise} \end{cases} \\
Pr[Z_i = 1] &\leq \frac{1}{3} && \text{(From 21.9)} \\
\mathbb{E}[Z] &= k\ Pr[Z_i = 1] \leq \frac{k}{3}
\end{aligned}
$$

By Chernoff Bounds:

$$
Pr[\ Z \geq (1+\eta)\mathbb{E}[Z]\ ] \leq e^{-\frac{1}{3}\eta^2\mathbb{E}[Z]} \tag{21.10}
$$

$$
\begin{aligned}
Pr[|Estimate - F_2| > \epsilon F_2] &\leq Pr\Big[Z \geq \frac{k}{2}\Big] \\
&= Pr\Big[Z \geq \frac{3}{2}\frac{k}{3}\Big] \\
&= Pr\Big[Z \geq \Big(1+\frac{1}{2}\Big)\frac{k}{3}\Big] \\
&\leq e^{-\frac{1}{3}(\frac{1}{2})^2\frac{k}{3}} && \text{(By Chernoff Bound 21.10)} \\
&= \ \delta && \text{(Assumed)}
\end{aligned}
$$

$$
\text{Hence, } log\ \delta = -\frac{1}{3}\Big(\frac{1}{2}\Big)^2\frac{k}{3} \ = \ -\frac{k}{36}
$$

$$
k = \ 36\ log\ \frac{1}{\delta} \tag{21.11}
$$

Pr[Estimate is between $(1-\epsilon)F_2$ and $(1+\epsilon)F_2$] $\geq 1 - \delta$

Matrix Visualization

We can visualize the final algorithm as running t*k independent copies of the original sketch, represented as a t*k sized random matrix. h_{ij}, the hash for X_{ij}, is selected randomly from the 4-Universal Family $\mathcal{H}$. Each row is averaged to get the $Mean_i$ and finally the median is calculated over the Means of the k rows.

$\longleftarrow$ t $\longrightarrow$

k

X_{00}^2	X_{01}^2	...	X_{0j}^2	...	X_{0t}^2	**Mean$_0$**
X_{10}^2	X_{11}^2	...	X_{1j}^2	...	X_{1t}^2	**Mean$_1$**
...	...	...	...	...	...	...
X_{i0}^2	X_{i1}^2	...	X_{ij}^2	...	X_{it}^2	**Mean$_i$**
...	...	...	...	...	...	...
X_{k0}^2	X_{k1}^2	...	X_{kj}^2	...	X_{kt}^2	**Mean$_k$**

Final Algorithm

Result: Output: $\underset{1\le i\le k}{median}\Big(\frac{1}{t}\sum_{j=1}^{t} X_{ij}^2\Big)$

Initialize Matrix X[t,k] $\leftarrow$ 0;
for *each s in stream* σ **do**
  for *each i in k* **do**
    for *each j in t* **do**
      $X_{ij} = X_{ij} + h(s)$
    end
  end
end

Space Requirements

The absolute value of x never exceeds the sum of all token frequencies i.e m, so the algorithm takes O(log m) bits to store this sketch and O(log n) bits to store the individual hash function h.

$$\begin{aligned}
\text{Space requirements} &= \text{O(t) * O(k) * O(Sketch)} \\
&= O\Big(\frac{4}{\epsilon^2}\Big) * O\Big(36log\frac{1}{\delta}\Big) * O(log\ m + log\ n) && \text{(From 21.9 and 21.11)} \\
&= O\Big(\frac{1}{\epsilon^2}log\frac{1}{\delta}(log\ m + log\ n)\Big)
\end{aligned}$$

21.3 Online Algorithms

In the remaining few minutes, Online Algorithms were introduced. They can process the input in the order it is fed to the algorithm, without having the entire input available at start. Because whole input is not known, an online algorithm is forced to make decisions that may later turn out not to be optimal. The study of online algorithms has focused on the quality of decision-making that is possible in this setting. The competitive ratio of an algorithm, is defined as the worst-case ratio of its cost divided by the optimal cost, over all possible inputs. Lets understand Online Algorithms with a comparison with Streaming Algorithms.

Scenario	Online Algorithms	Streaming Algorithms
Data Stream Length	Unknown	Known and typically big
Evaluations	Every time step	At the end
Multipass	No	Maybe possible