COMS W6998-5: Algorithms through Geometric Lens (Fall'17) 9 September, 2017

Lecture 2 – Estimating Frequency Moments and Heavy Hitters

Instructors: *Alex Andoni, Ilya Razenshteyn* Scribe: *Nishanth Mohan*

1 Introduction

Last time, we introduced the TugOfWar algorithm of Alon *et al.* [**?**] to estimate the second frequency moment of a stream. This time, we finish its analysis, describe and analyze an improved version of it (called TugOfWar++), and introduce the heavy hitters problem.

2 Estimating Second Frequency Moments

As in the previous lecture, assume we have a stream $e = (e_1, e_2 \dots e_m)$ of length m each element of which is from the universe $[n] = \{1, 2, \dots n\}$. Given the frequency vector $\boldsymbol{X} = (X_1, X_2 \dots X_n)^T$ where X_i is the frequency of $i \in [n]$, we want to estimate its second moment F_2 defined as

$$F_2 = \sum_{i=1}^{n} X_i^2$$

To do this, we introduced the TugOfWar algorithm which is as follows:

Algorithm 1: TugOfWar

Input: Stream $e = (e_1, e_2 \dots e_m)$ where each $e_j \in [n]$
Output: Estimate $\hat{a}$ of $F_2 = \sum_{i=1}^{n} X_i^2$

1: Generate n independent Rademacher random samples $r_1, r_2 \dots r_n \in \{-1, +1\}$
2: $Z \leftarrow 0$
3: **upon seeing** $e_j \in [n]$ **do**
4: $\quad Z \leftarrow Z + r_{e_j}$
5: **output** $\hat{a} = Z^2$

Note that

$$Z = \sum_{i=1}^{n} X_i \cdot r_i$$

Analysis

Claim 1. $\mathbb{E}[Z] = 0$

Proof. We have

$$\mathbb{E}[Z] = \mathbb{E}\left[\sum_{i=1}^{n} X_i \cdot r_i\right]$$

$$
\begin{aligned}
\Rightarrow \mathbb{E}[Z] &= \sum_{i=1}^{n} \mathbb{E}[X_i \cdot r_i] && \dots \text{(by the linearity of expectation)} \\
&= \sum_{i=1}^{n} X_i \cdot \mathbb{E}[r_i] && \dots (X_i \text{ is constant with respect to } r_i) \\
&= 0
\end{aligned}
$$

□

Claim 2. $\mathbb{E}[Z^2] = F_2$

Proof. We have

$$
\begin{aligned}
\mathbb{E}[Z^2] &= \mathbb{E}\left[\left(\sum_{i=1}^{n} X_i \cdot r_i\right)^2\right] \\
&= \left(\sum_{i=1}^{n} X_i^2 \cdot \underbrace{\mathbb{E}[r_i^2]}_{=1}\right) + \left(\sum_{i \neq j} X_i X_j \cdot \underbrace{\mathbb{E}[r_i r_j]}_{=0}\right) \\
&= \sum_{i=1}^{n} X_i^2 \\
&= F_2
\end{aligned}
$$

so $\hat{a} = Z^2$ is an unbiased estimator of F_2
.

□

Claim 3. $\mathrm{Var}[\hat{a}] \leq 4F_2^2$

Proof. We have

$$
\begin{aligned}
\mathrm{Var}[\hat{a}] &= \mathbb{E}[Z^4] - \mathbb{E}[Z^2]^2 \\
&\leq \mathbb{E}[Z^4] && (1) \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{n} X_i \cdot r_i\right)^4\right] \\
&= \mathbb{E}\left(\sum_{i,j,k,\ell} X_i X_j X_k X_\ell \cdot \mathbb{E}[r_i r_j r_k r_\ell]\right) \\
&= \left(\sum_{i} X_i^4 \cdot \underbrace{\mathbb{E}[r_i^4]}_{=1}\right) + 3\left(\sum_{i \neq j} X_i^2 X_j^2 \cdot \underbrace{\mathbb{E}[r_i^2 r_j^2]}_{=1}\right) && (2) \\
&= \left(\sum_{i} X_i^4\right) + 3\left(\sum_{i \neq j} X_i^2 X_j^2\right)
\end{aligned}
$$

$$\Rightarrow \text{Var}[\hat{a}] \leq 4\left(\sum_{i=1}^{n} X_i^2\right)^2 \tag{3}$$
$$= 4F_2^2 \qquad \ldots \text{(from Claim 2)}$$

(1) follows since $\mathbb{E}[Z^4] \geq \mathbb{E}[Z^2]^2$ by Jensen's inequality and $\mathbb{E}[Z^2]^2 \geq 0$.

(2) follows since:

- Expectation terms with odd powers of r_i evaluate to 0.
- There are $\frac{1}{2}\binom{4}{2} = 3$ terms of the form $\mathbb{E}[r_i^2 r_j^2]$

(3) holds since

$$\left(\sum_{i=1}^{n} X_i\right)^2 = \left(\sum_{i=1}^{n} X_i^2\right) + \left(\sum_{i \neq j} X_i X_j\right)$$
$$\geq \max\left\{\sum_{i=1}^{n} X_i^2 \;,\; \sum_{i \neq j} X_i X_j\right\}$$

□

Finally, combining Claim 3 and Chebyshev's inequality yields

$$\Pr\left[|\hat{a} - F_2| \geq 7F_2\right] \leq \frac{\text{Var}(Z^2)}{(7F_2)^2} = \frac{4}{49} \leq 0.1$$

so we have $0 \leq \hat{a} \leq 8F_2$ with probability at least 0.9.

Space Requirement: Storing our sketch requires $O(\log n)$ space. Moreover, for the r_i's, it turns out we only need a 4-wise independent hash function $h : [n] \to \{-1, +1\}$ (4-wise independent since we assume the independence of at most every distinct quartet (r_i, r_j, r_k, r_ℓ) to arrive at (2)) which can be implemented with $O(\log n)$ bits, so we need $O(\log n)$ bits in total.

3 Improved Second Frequency Moment Estimator

While the TugOfWar algorithm only takes $O(\log n)$ space, its variance is too high, so we use a standard averaging trick to reduce its variance while keeping the space requirements relatively low. This improved algorithm, TugOfWar++, is as follows:

Algorithm 2: TugOfWar++

Input: Stream $e = (e_1, e_2 \ldots e_m)$ where each $e_j \in [n]$, $k \in \mathbb{Z}_+$
Output: Estimate $\tilde{a}$ of $F_2 = \sum_{i=1}^{n} X_i^2$

1: **for** $i = 1, 2 \ldots k$ **do**
2: $\quad Z_k \leftarrow$ TugOfWar(e)
3: **output** $\tilde{a} = \frac{1}{k}\sum_{i=1}^{k} Z_k$

Analysis

Claim 4. $\mathbb{E}[\tilde{a}] = F_2$

Proof. We have

$$\begin{aligned}
\mathbb{E}[\tilde{a}] &= \mathbb{E}\left[\frac{1}{k}\left(\sum_{i=1}^{k} Z_k\right)\right] \\
&= \frac{1}{k}\left(\sum_{i=1}^{k} \mathbb{E}[Z_k]\right) && \ldots \text{(by the linearity of expectation)} \\
&= F_2 && \ldots \text{(from Claim 2)}
\end{aligned}$$

so $\tilde{a}$ is an unbiased estimator of F_2.

□

Claim 5. $\mathrm{Var}[\tilde{a}] \leq 4F_2^2/k$

Proof. We have

$$\begin{aligned}
\mathrm{Var}[\tilde{a}] &= \mathrm{Var}\left[\frac{1}{k}\left(\sum_{i=1}^{k} Z_k\right)\right] \\
&= \frac{1}{k^2}\left(\sum_{i=1}^{k} \mathrm{Var}[Z_k]\right) && \ldots \text{(by the linearity of variance for uncorrelated variables)} \\
&= \frac{k \cdot \mathrm{Var}[Z_k]}{k^2} \\
&\leq \frac{4F_2^2}{k} && \ldots \text{(from Claim 3)}
\end{aligned}$$

□

Finally, combining Claim 5 and Chebyshev's inequality, for $k = 49/\epsilon^2$, yields

$$\Pr\left[|\tilde{a} - F_2| \geq \epsilon F_2\right] \leq 0.1$$

so we have a $(1+2\epsilon)$-estimator for F_2 for $\epsilon < 1/2$ since

$$\frac{1+\epsilon}{1-\epsilon} \leq 1 + 2\epsilon$$

for $0 \leq \epsilon \leq 1/2$.

Space Requirement: We have to maintain $k = O(1/\epsilon^2)$ estimates of F_2, so we need $O(\log n/\epsilon^2)$ space for the sketch in total.

Note: When viewed in matrix notation, TugOfWar++ estimates the ℓ_2-norm squared $\|\boldsymbol{Z}\|_2^2$ of the vector

$$\boldsymbol{Z} = \boldsymbol{R} \cdot \boldsymbol{X}$$

where $\boldsymbol{X}$ is the vector of frequencies as defined before and $\boldsymbol{R}$ is the matrix

$$\boldsymbol{R} = \frac{1}{\sqrt{k}} \begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{i1} & \cdots & r_{ij} & \cdots & r_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{k1} & \cdots & r_{kj} & \cdots & r_{kn} \end{bmatrix}$$

wherein r_{ij} denotes the j^{th} independent Rademacher random sample in the i^{th} run of the TugOfWar algorithm. Note that we only use $k = O(1/\epsilon^2)$ instead of n rows for our estimator and $\boldsymbol{Z} \in \mathbb{R}^k$, so this a form of dimension reduction which we will explore more formally in a later lecture.

4 Application - Difference Traffic

Consider a situation similar to before where we have the universe $[n]$ and a network (like a transportation or traffic network for instance) with distinct entry and exit points. The entry point sees a stream with frequency vector $\boldsymbol{Y} = (Y_1, Y_2 \dots Y_n)^T$ while the exit point sees a stream with frequency vector $\boldsymbol{X} = (X_1, X_2 \dots X_n)^T$. We want to estimate the difference traffic $\mathrm{dt}(\boldsymbol{X}, \boldsymbol{Y})$ between the two streams defined as

$$\mathrm{dt}(\boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{X} - \boldsymbol{Y}\|_2^2 = \sum_{i=1}^{n} (X_i - Y_i)^2$$

Note that this is just the second moment of the vector $\boldsymbol{C} = \boldsymbol{X} - \boldsymbol{Y}$.

We can use the framework developed in the previous section for this as follows: for $\boldsymbol{Y}$, we compute the sketch $f(\boldsymbol{Y}) = \boldsymbol{R} \cdot \boldsymbol{Y}$. Similarly, for $\boldsymbol{X}$, we compute the sketch $f(\boldsymbol{X}) = \boldsymbol{R} \cdot \boldsymbol{X}$ with the *same* random matrix $\boldsymbol{R}$. We finally compute our estimate $g(\boldsymbol{X}, \boldsymbol{Y})$ of the difference traffic as

$$g(\boldsymbol{X}, \boldsymbol{Y}) = \|f(\boldsymbol{X}) - f(\boldsymbol{Y})\|_2^2 = \|\boldsymbol{R} \cdot (\boldsymbol{X} - \boldsymbol{Y})\|_2^2 = \|f(\boldsymbol{C})\|_2^2$$

so $g(\boldsymbol{X}, \boldsymbol{Y})$ is indeed an estimator of the second moment of $\boldsymbol{C}$.

This general property of $f(\alpha\boldsymbol{X} + \beta\boldsymbol{Y}) = \alpha f(\boldsymbol{X}) + \beta f(\boldsymbol{Y})$ wherein $f(\cdot)$ is a sketch of its argument and $\alpha, \beta \in \mathbb{R}$ are real numbers is know as *linearity* and is applicable to a wide variety of situations (essentially those wherein $f(\cdot)$ is a linear sketch).

5 Heavy Hitters

In the previous sections, we developed algorithms to estimate the second frequency moment of a stream. This, however, is a global property of the stream and doesn't tell us much about the individual frequencies of the elements in it. A more interesting problem towards this end is to find the element with the highest frequency. It turns out, however, that this is a hard problem: in general, we need $\Omega(n)$ space to find the most frequent element (more specifically, a result in [**?**] shows that any randomized algorithm estimating $\|\boldsymbol{X}\|_\infty$ for $m \geq 2n$ to within a multiplicative factor of $1/3$ with probability $\epsilon < 1/2$ requires $\Omega(n)$ space).

We thus try to solve a more modest problem wherein we try to find elements which appear "sufficiently" frequently in the stream. Such elements are known as a heavy hitters.

Definition 1. (ϕ-Heavy Hitter) *An element $i \in [n]$ is a ϕ-heavy hitter if $X_i \geq \phi \left(\sum_{j \leq n} X_j \right) = m\phi$ for $\phi \in (0, 1)$.*

5.1 Identifying Heavy Hitters

At a first pass towards an algorithm to identify heavy hitters, consider a universal hash function $h : [n] \to [w]$ where $w = O(1/\phi)$. Each element $i \in [n]$ is mapped to the bucket $h(i) \in [w]$. As before, we associate an independent Rademacher random sample $r_i \in \{-1, +1\}$ with each i. Now, define

$$S(j) = \sum_{i:h(i)=j} X_i \cdot r_i$$

To see whether $i \in [n]$ is a ϕ-heavy hitter, we look at the quantity $S(h(i))$ and output i only if $|S(h(i))| \geq m\phi$. More concretely, the algorithm is as follows:

Algorithm 3: IdentifyByHash

Input: Stream $e = (e_1, e_2 \dots e_m)$ where each $e_j \in [n]$, $\phi \in (0, 1)$
Output: ϕ-heavy hitters in e

```
Generate n independent Rademacher random samples r_1, r_2 ... r_n ∈ {−1, +1}
S[1 ... w] ← [0 ... 0]
upon seeing e_j ∈ [n] do
    S[h(e_j)] ← S[h(e_j)] + r_{e_j}
for i = 1, 2 ... n do
    if |S(h(i))| ≥ mφ then
        output i
```

Analysis

For $i \in [n]$, let $\delta(i) = S(h(i)) - X_i \cdot r_i$.

Claim 6. *$S(h(i))$ is an unbiased estimator of $X_i \cdot r_i$.*

Proof. We have

$$\begin{aligned}
\mathbb{E}[\delta(i)] &= \mathbb{E}\left[\sum_{\substack{i' \neq i \\ h(i')=h(i)}} X_{i'} \cdot r_{i'} \right] \\
&= \sum_{\substack{i' \neq i \\ h(i')=h(i)}} X_{i'} \cdot \mathbb{E}[r_{i'}] \\
&= 0
\end{aligned}$$

The claim now follows from the linearity of expectation. □

References

[AMS96] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.