(CS602) Applied Algorithms 20 Jan, 2014

Lecture 5: Second frequency moment, F_2

Lecturer: Nutan Limaye *Scribe: Nutan Limaye*

In the last class we gave a randomized (ε, δ) algorithm for approximating the number of distinct elements using space $O(\frac{1}{\varepsilon} \cdot \log\left(\frac{1}{\delta}\right) \cdot \log^2 m)$.

Today we will define the notion of frequency moments and give (ε, δ) algorithm for approximating the second frequency moment using space $O(\frac{1}{\varepsilon^2} \cdot \log\left(\frac{1}{\delta}\right) \cdot \log m)$.

5.1 Frequency Moments

Let $x_1, x_2, \ldots, x_n$ be input stream and for each $i \in [n]$ let $x_i \in [m]$. Let f_j denote the number of times the element $j \in [m]$ appears in the stream. The kth frequency moment is defined as follows:

$$F_k = \sum_{j \in [m]} f_j^k$$

As per this definition, F_0 is the number of distinct elements in the stream and F_1 is the length of the stream. We gave space efficient algorithms to approaximate these quantities over the last few lectures. Today we will give an algorithm to approximate F_2.

Pick h uniformly randomly from 4-wise independent family of functions
$\mathcal{F} = \{h : [m] \to \{\pm 1\}\}$;
Sum $\leftarrow 0$;
while *there exists x, an input element* **do**
  Sum $\leftarrow$ Sum $+ h(x)$;
end
Output $Z \leftarrow (\text{Sum})^2$;

We will first analyse the expected value of the output of the algorithm.

Lemma 5.1.1. $\mathbb{E}(Z) = F_2$

Proof.

$$\begin{aligned}
\mathbb{E}(Z) &= \mathbb{E}\left(\text{Sum}^2\right) \\
&= \mathbb{E}\left(\left(\sum_{x \in \text{stream}} h(x)\right)^2\right) && \text{(From the definition of Sum)} \\
&= \mathbb{E}\left(\left(\sum_{j \in [m]} f_j h(j)\right)^2\right) && \text{(From the definition of } h(x)\text{)}
\end{aligned}$$

From here we see that,

$$\begin{aligned}
\mathbb{E}(Z) &= \mathbb{E}\left(\sum_{j\in[m]} f_j^2 h(j)^2 + \sum_{j\neq\ell} f_j f_l h(j)h(\ell)\right) \\
&= \sum_{j\in[m]} f_j^2 \mathbb{E}(h(j)^2) + \sum_{j\neq\ell} f_j f_l \mathbb{E}(h(j)h(\ell)) && \text{(By linearity of expectation)} \\
&= \sum_{j\in[m]} f_j^2 \cdot 1 + \sum_{j\neq\ell} f_j f_l \cdot 0 && \text{(As } h(j)^2 = 1\ \forall j \text{ and Pairwise independence of } \mathcal{F}\text{)} \\
&= F_2 && \text{(By the definition of } F_2\text{)}
\end{aligned}$$

□

Lemma 5.1.2. $\mathbb{V}ar(Z) \leq 2F_2^2$.

To reduce the variance even further, we use the *averaging trick*. If we run t copies of the same algorithm and let the output, say Z', be the average of the outputs of all the t algorithms then we will get the following:

Lemma 5.1.3. *$\mathbb{E}(Z') = F_2$ and $\mathbb{V}ar(Z') \leq 2F_2^2/t$.*

Now using Chebyshev's inequality we know that

$$\Pr\left[|Z' - \mathbb{E}(Z')| \geq \varepsilon F_2\right] \leq \frac{2F_2^2}{t\varepsilon^2 F_2^2} \leq 1/3 \text{ (for appropriate choice of } t\text{)}$$

We can further reduce the probability of error to be bounded above by δ by using the *median trick*.

We now argue the space bound. To compute Z, we need to keep track of the variable Sum, which can be stored in $O(\log n)$ space. The number of bits required to pick a random function from the family of 4-wise independent hash functions equals $\log(|\mathcal{F})$. It is known that for any family of functions $\mathcal{H} = \{h : [m] \to [k]\}$, there exists a subfamily $\mathcal{F} \subset \mathcal{H}$ of 4-wise independent hash functions of size $k^{\log m}$. Therefore, he number of bits required to pick a random function from the family of 4-wise independent hash functions equals $\log(|\mathcal{F}) = O(\log k \log m)$. As $k = 2$ here, we can choose a random function using $O(\log m)$ bits. As we saw in Lecture 1, the use of the averaging trick and the median trick along with this space bound we get that the randomized (ε, δ) approximation algorithm for F_2 uses space $O(\frac{1}{\varepsilon^2} \cdot \log\left(\frac{1}{\delta}\right) \cdot \log m)$.

This algorithm presented here is from a seminal paper by Alon, Matias and Szegedy.

5.2 Exercises

Exercise 1. *Prove Lemmas 5.1.2, 5.1.3.*