CS369G: Algorithmic Techniques for Big Data Spring 2015-2016

Lecture 4: Estimating F_k moments for k $\in [0, 2)$.

Prof. Moses Charikar *Scribes: Lei Lei, Jacek Skryzalin*

1 Overview

This lecture starts with a recap of F_k sketch in [AMS96]. We'll review how to estimate the kth moment F_k of a data stream for $k > 2$, and we'll discuss how to estimate the kth moment F_k of a data stream for $k \in [0, 2)$.

For $k \geq 2$, our algorithm will use $O\left(n^{1-\frac{1}{k}}\right)$ space. We might see later that estimating such F_k requires $\Omega\left(n^{1-\frac{2}{k}}\right)$ space, so our algorithm isn't quite optimal, but it's rather competitive nonetheless.

2 Estimating F_k

(Continuing from last lecture) To estimate $F_k = \sum_i f_i^k$ (define $F_\infty = \max_i f_i$), we define a random variable Y such that its expectation is F_k, its variance and space requirement is small. We then apply median-of-average technique to obtain desired results.

We fix the function $g(x) = x^k$ throughout this lecture, but we note that these techniques also apply for a larger class g (in particular, we'll need the convexity of g and the fact that $g(0) = 0$).

Defining Y

Consider data stream S of length m.

$$S = a_1, a_2, \ldots, a_m \in [n]$$

and for $i \in [n]$, let f_i be $\{x \in S \mid x = i\}$, the number of time it appears in the stream.

Choose a random element a_p of the stream S, where p is choosen uniformly at random from $[m]$. Suppose that $a_p = l \in [n]$, and define

$$R = |\{q \mid q \geq p, a_q = l\}$$

be the number of occurrences of l in the stream S following x_p. We then define X as:

$$Y = m(g(R) - g(R-1)).$$

Implementation

If the length of the stream, m, is not known in advance, a_p can be sampled by reservoir sampling and we maintain a conuter for R as follows

$(x, R) = (s_0, 1)$;
while $t = 0, 1, \ldots$ **do**
  $t = t + 1$;
  $$(x, R) = \begin{cases} (s_t, 1) & \text{with probability } 1/t \\ (x, R) & \text{if } s_t \neq s \\ (x, R+1) & \text{if } s_t = s \end{cases};$$
end

The following lemma proves the correctness of reservoir sampling.

Lemma 1. *For all T, the probability that the last time the counter is reset is at time T is $\frac{1}{m}$.*

Proof. The probability that the last time the counter is reset is at time T is the product of

- $\frac{1}{T}$ (because this is the probability that we'll reset the counter at time T), and
- $1 - \frac{1}{r} = \frac{r-1}{r}$ for all $r \in \{T+1, T+2, ..., m\}$ (these are the probabilities that we won't reset the counter after time T).

So the probability that the counter is reset at time T is

$$\left(\frac{1}{T}\right)\left(\frac{T}{T+1}\right)\left(\frac{T+1}{T+2}\right)\cdots\left(\frac{m-1}{m}\right) = \frac{1}{m}.$$

□

Calculating $E[Y]$ and $Var[Y]$

Lemma 2. *For $k > 0$, we have $F_\infty \leq {F_k}^{1/k}$.*

Proof.

$$F_\infty = \max_i f_i = \left(\max_i {f_i}^k\right)^{1/k} \leq \left(\sum_i {f_i}^k\right)^{1/k} = {F_k}^{1/k}.$$

□

Lemma 3. *For $k \geq 1$, we have $\frac{F_1}{n} = \frac{\sum_i f_i}{n} \leq \left(\frac{\sum_i f_i^k}{n}\right)^{1/k} = \left(\frac{F_k}{n}\right)^{1/k}$. Hence, $F_1 \leq n^{1-\frac{1}{k}} {F_k}^{\frac{1}{k}}$.*

Proof. Note that $g(x) = x^k$ is convex. By convexity, we get

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

Evaluating g yields

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

Taking kth roots yields the desired result. □

Claim 4 (Thm 2.1, [AMS96])**.** *For $g(x) = x^k$,*

$$E[Y] = \sum_i g(f_i), \quad Var[Y] \leq kF_1F_{2k-1} \leq kn^{1-\frac{1}{k}}(F_k)^2$$

Proof. Refer to last lecture for calculation of expectation. For variance, we dutifully commence the long and arduous computation:

$$\begin{aligned}
Var(Y) &= E[Y^2] - E[Y]^2 \\
&\leq E[Y]^2 \\
&= \sum_{i=1}^{n} \sum_{j=1}^{f_i} \left(\frac{1}{m}\right) (m(g(j) - g(j-1)))^2 \\
&= m \sum_{i=1}^{n} \sum_{j=1}^{f_i} \left[j^k - (j-1)^k\right] \left[j^k - (j-1)^k\right] \\
&\leq m \sum_{i=1}^{n} \sum_{j=1}^{f_i} kj^{k-1} \left[j^k - (j-1)^k\right] \\
&= km \sum_{i=1}^{n} \sum_{j=1}^{f_i} j^{2k-1} - j^{k-1}(j-1)^k \\
&\leq km \sum_{i=1}^{n} f_i^{2k-1} \\
&= kF_1F_{2k-1}.
\end{aligned}$$

In the 5th line, we use the fact that $j^k - (j-1)^k \leq kj^{k-1}$, which can be proven using the convexity of $g(x) = x^k$ for $k \geq 2$ (very important!). To get from the 6th line to the 7th line, we telescope our sums and use the fact that $-(j+1)^{k-1}j^k + j^{2k-1} < 0$, which allows us to derive the desired upper bound on the inner summation. This proves our first inequality!

But the proof isn't done yet. There's still another inequality to prove. After taking deep breaths,

we compute:

$$\begin{aligned} F_1 F_{2k-1} &= F_1 \left(\sum_{i=1}^{n} f_i^{2k-1} \right) \\ &\leq F_1 \left(F_\infty^{k-1} \sum_{i=1}^{n} f_i^k \right) \\ &\leq F_1 {F_k}^{\frac{k-1}{k}} F_k \\ &\leq n^{1-\frac{1}{n}} F_k^2. \end{aligned}$$

We use the definition of F_∞ in the 2nd line (on $k-1$ copies of f_i in each of the summands), Lemma 2 in the 3rd line, and Lemma 3 in the 4th line. And that officially concludes the proof of this claim. □

To obtain an approximation of F_k, we may take the mean of $O\left(\frac{kn^{1-\frac{1}{k}}}{\epsilon^2}\right)$ independent random estimates. This average will have expected value F_k and variance at most $\epsilon^2 F_k^2$. By taking the median of $O(-\log\delta)$ independent copies of this average, we will attain an approximate value for F_k with high probability (proof using Chebyshev's inequality as in previous lectures).

3 Estimating F_k for $k \leq 2$

In this section we discuss techniques for estimating F_k for $k < 2$. Note that estimating F_1 is rather easy (a counter over the whole stream will work quite nicely). But we want a *linear* sketch of the data. For example, given two streams (with counts given as f_i and g_i), we might want to estimate $\sum_{i=1}^{n} |f_i - g_i|$. This is much more difficult.

For F_2, the key property that enables a $O(\log n)$ space constraint sketch is the following:

Observation 5. *Given i.i.d $X_1, X_2 \sim \mathcal{N}(0,1)$, $a_1X_1 + a_2X_2 \sim \sqrt{a^2+b^2}X$ where $a, b \in \mathbb{R}$ and $X \sim \mathcal{N}(0,1)$. In general, given i.i.d $X_1, X_2, \ldots, X_n \sim \mathcal{N}(0,1)$ and $a = [a_1, a_2, \ldots, a_n]^T \in \mathbb{R}^n$,*

$$\sum_{i=1}^{n} a_i X_i \sim ||a||_2 X, \quad X \sim \mathcal{N}(0,1)$$

This motivates the following definition:

Definition 6 (p-stable distribution)**.** *A distribution $\mathcal{D}$ is called p-stable if given two independent $X_1, X_2 \sim \mathcal{D}$, and any $a, b \in \mathbb{R}$,*

$$aX_1 + bX_2 \sim (a^p + b^p)^{1/p} X, \quad X \sim \mathcal{D}$$

Claim 7. *The following facts are known for p-stable distribution and mentioned without proof:*

1. *[Zol86] p-stable distributions exist for $p \in (0, 2]$.*

2. *A Cauchy distribution, defined by the density function* $p(x) = \frac{1}{\pi}\frac{1}{x^2+1}$, *is 1-stable.*

3. *The Gaussian (normal) distribution, defined by the density function* $p(x) = \frac{1}{\sqrt{2\pi}}\exp(-x^2/2)$, *is 2-stable.*

4. *[Lév25] for* $p \in (1,2)$, $\mathcal{D}_p$ *has finite mean and infinite variance; for* $p \in (0,1]$, $\mathcal{D}_p$ *has infinite mean and infinite variance.*

5. *[CMS76] To generate a sample from p-stable distribution for* $p \in (0,2]$, *first generate* $\theta \in Uniform[-\pi/2, \pi/2]$ *and* $r \in Uniform[0,1]$, *then our sample is given as*

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

3.1 Estimating $F_k, k \leq 1$

Claim 8 (Sec 3.1, Thm 3 in [Ind00])**.** *For* $(i,j) \in [n] \times [t]$, *let* $\{X_{i,j}\}$ *be nt i.i.d random variables drawn from a p-stable distribution* $\mathcal{D}_p$, $p \leq 1$. *Suppose further that they are scaled such that* $P(|Y| \leq 1) = 1/2$, $Y \sim \mathcal{D}_p$. *For* $j \in [t]$,

$$Y_j = \sum_i X_{i,j} f_i = (\sum_i f_i^p)^{1/p} X_j, X_j \sim \mathcal{D}_p$$

the median of $|Y_1|, |Y_2|, \ldots, |Y_t|$ *correctly estimates* F_p *up to a factor of* $1 \pm \epsilon$ *with probability at least* $1 - \delta$ *if* $t = \frac{c}{\epsilon^2}\log(\frac{1}{\delta})$ *for a suitable choice of* $c > 1$.

4 Pseudo-randomness

All sketch discussed so far are examples of linear sketch. For a data stream

$$S = s_1, s_2, \ldots, s_m \in [n]$$

at each time step, we have a single entry update to the vector $f = (f_1, f_2, \ldots, f_n)^t \in \mathbb{R}^n$ and updates Πf accordingly, where $\Pi_{i,j} \in \mathbb{R}^{t \times n}$ is a matrix of i.i.d random variables drawn from $\mathcal{D}_p$. Suppose that all real numbers are represented by b bits, at each time step, we maintains $O(bt)$ bits in memory and make t random access to Π, which takes $O(ntb)$ to specify. Question: can we specify Π with less bits?

[KNW10] has shown that if $X_{i,j}$ are drawn from k-wise independent p-stable distribution instead of i.i.d p-stable distribution, similar probabilistic guarantee holds if $k = O(1/\epsilon^p)$ and the space requirement is $\Omega\left(n^{1-\frac{2}{k}}\right)$. This result matches theoretical lower bound on space requirement [AMS96] for estimating F_p with $p \in (1,2]$. [Ind00] used psuedo-randomness to achieve a similiar result.

4.1 Nisan Pseudo-randomness Generator

Let f be a small program with bounded memory, say S bits. we can think of it as discrete finite automaton with 2^S possible state. At each step, it is fed with a string of s bits and makes a state transition. After R steps, the state it is at is the output. The entire input string can be thought of as a stream of data:

$$S = x_1, x_2, \ldots, x_R$$

which takes RS truly random bits to specify. Nisan's result [Nis92] is that:

Claim 9. *Suppose a seed of $S\log(R)$ bits is used to generate a peusdo-random string of RS bits, say S'. Then f can be fed with S' instead of S and its behavior remains 'unchanged'.*

Indyk first analyzed the stream algorithm with truly random Π. Then he show that it used a small memory and could be fooled with Nisan's PRG to generate Π. It will be discussed next class.

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.

[CMS76] John M Chambers, Colin L Mallows, and BW Stuck. A method for simulating stable random variables. *Journal of the american statistical association*, 71(354):340–344, 1976.

[Ind00] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 189–197. IEEE, 2000.

[KNW10] Daniel M Kane, Jelani Nelson, and David P Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 41–52. ACM, 2010.

[Lév25] Paul Lévy. *Calcul des probabilités*, volume 9. Gauthier-Villars Paris, 1925.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[Zol86] Vladimir M Zolotarev. *One-dimensional stable distributions*, volume 65. American Mathematical Soc., 1986.