CS 229r: Algorithms for Big Data Fall 2013

Lecture 4 — September 12, 2013

Prof. Jelani Nelson *Scribe: Brenton Partridge*

Contents

1 Overview

This lecture and the previous two lectures have dealt with variations on the "turnstile stream" problem, in which a vector $x \in \mathbb{R}^n$, initially initialized to zero, is given coordinate-wise updates of the form $x_i \leftarrow x_i + v$ for $v \in \mathbb{R}$, then queried for a statistic of x. One such problem is the F_p problem, in which the statistic is $F_p \triangleq \sum_{i=1}^n |x_i|^p = \|x\|_p^p$ for some $p \geq 0$. Another is the "heavy hitters" problem, where we wish to know the set of i's such that x_i is relatively larger than the other components. (One motivating example for this is Google Trends, where the search engine collects updates for frequencies of search terms, then wishes to know the most popular search terms. Heavy hitter streaming algorithms were used for this purpose in [PDGQ05].)

Linear sketches are a common way of storing data from a turnstile stream such that a specific statistic can be estimated with provable bounds on space and precision. Here, instead of maintaining $x \in \mathbb{R}^n$, we maintain $y = \Pi x \in \mathbb{R}^m$ with $m < n$ and $\Pi \in \mathbb{R}^{m \times n}$. As mentioned in previous lecture notes, updates then take the form $y \leftarrow y + v\Pi_i$ where $\Pi_i \in \mathbb{R}^m$ is the i'th column of Π; the space complexity is reduced to $O(m) + (\text{storage of } \Pi, \text{ presumably} \ll O(n))$.

In the previous lecture we discussed linear sketches for F_p estimation when $0 < p < 2$, specifically Indyk's Algorithm [Indyk06]. In this lecture, we first discuss and analyze linear sketches for F_p estimation where $p > 2$, as presented in [Andoni12]. We then turn our attention to the "heavy hitters" problem described above and analyze a linear sketch for that, as presented in [NNW12].

2 Algorithm for F_p, $p > 2$

This linear sketch, as presented in [Andoni12], uses $\Pi = PD$ such that:

$$y = \underbrace{\begin{bmatrix} & \sigma_2 & & \\ & & \sigma_3 & \cdots \\ \sigma_1 & & & \sigma_n \end{bmatrix}}_{P \in \mathbb{R}^{m \times n}} \underbrace{\begin{bmatrix} \frac{1}{u_1^{1/p}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{u_n^{1/p}} \end{bmatrix}}_{D \in \mathbb{R}^{n \times n}} \begin{bmatrix} | \\ x \\ | \end{bmatrix}$$

- Choose $m = O\left(n^{1-\frac{2}{p}} \cdot \log n\right)$, $h : [n] \to [m]$, $\sigma : [n] \to \{-1, 1\}$, and $u : [n] \to \text{Exp}(1)$
  - Note that the lower bound on any algorithm like this takes $m = O\left(n^{1-\frac{2}{p}}\right)$ space, as shown in [BJKS04] (more detail will be given later).
  - Also note that the first $O\left(\text{polylog}(n) \cdot n^{1-\frac{2}{p}}\right)$-space algorithm for this problem was presented in [IW05].
  - Recall that if $u \sim \text{Exp}(1)$ then $\mathbb{P}(u > \lambda) = \begin{cases} 1, & \lambda \leq 0 \\ e^{-\lambda}, & \text{else.} \end{cases}$
  - Let $P_{h(i),i} = \sigma_i$, and the remainder of entries 0.
  - D is a diagonal matrix with $D_{ii} = \frac{1}{u_i^{1/p}}$, and the remainder of entries 0.
- We will also introduce the notation $z = Dx$, such that $y = Pz$.[1]
- Outputs $\|y\|_\infty = \max_{1 \leq i \leq m} |y_i|$

We will prove the following bound:

Theorem 1. $\mathbb{P}\left(\|y\|_\infty \in \left[\frac{1}{4}\|x\|_p, 4\|x\|_p\right]\right) > \frac{51}{100}$ *(see footnote[2])*

In order to prove this, we will need a number of tools, including a new formulation of the Chernoff bound.

[1] Note that [Andoni12] reverses the notation for y and z if you use that for reference.

[2] The bound is chosen to be an arbitrary value bigger than $\frac{1}{2}$ so we can boost it using the standard Chernoff argument.

2.1 Alternate formulation of Chernoff bound

Theorem 2. *For $X_1, ..., X_n$ indep., $\forall i, |X_i| \leq K$, $\mathbb{E}\sum X_i = \mu$, $\sum_i \mathbb{E}\left(X_i - \mathbb{E}X_i\right)^2 = \sigma^2$,*

$$\mathbb{P}\left(\left|\sum_i X_i - \mu\right| > \lambda\right) \lesssim \max\left\{e^{-c\lambda^2/\sigma^2}, \left(\frac{\sigma^2}{\lambda K}\right)^{\frac{c\lambda}{K}}\right\}$$

Proof. Refer to Theorem 2 and Exercise 3 of [Tao10] (a hyperlink is in the references at the end of this document). □

2.2 Returning to proof of Theorem 1

Because of properties of the exponential distribution, the vector $z = Dx$ has an L_∞ norm similar to its L_p norm. Unfortunately, with just this bound, z is still n-dimensional. We can map it down using the hash function h of random size.

Now, we can sketch part of the outline of the proof. Intuitively, we are saying that the largest element in z is roughly the L_p norm of x. However, since there may be many large elements in z, we will choose m large enough that these large elements are all sent to their own buckets, all hashed to different places. Then, because our random signs σ have expectation 0, the smaller elements will not change the content of those buckets by much.

Claim 3. $\|z\|_\infty \in \left[\frac{1}{2}\|x\|_p, 2\|x\|_p\right]$ with probability $> \frac{3}{4}$.

Proof. Let $q = \min\left\{\frac{u_1}{|x_1|^p}, ..., \frac{u_n}{|x_n|^p}\right\}$. Throughout this, we will assume p is constant.

Since the u_i's are independent,

$$\begin{aligned}
\mathbb{P}(q > \lambda) &= \mathbb{P}\left(\forall i, \frac{u_i}{|x_i|^p} > \lambda\right) \\
&= \prod_i \mathbb{P}\left(\frac{u_i}{|x_i|^p} > \lambda\right) \\
&= \prod_i e^{-\lambda|x_i|^p} = e^{-\lambda\|x\|_p^p}
\end{aligned}$$

All this implies that $q \sim \frac{\text{Exp}(1)}{\|x\|_p^p}$. Therefore:

$$\begin{aligned}
\mathbb{P}\left(\|z\|_\infty \in \left[\frac{1}{2}\|x\|_p, 2\|x\|_p\right]\right) &= \mathbb{P}\left(q \in \left[\frac{1}{2^p}\|x\|_p^p, 2^p\|x\|_p^p\right]\right) \\
&= e^{-\frac{1}{2^p}} - e^{-2^p} \\
&> \frac{3}{4} \text{ for } p > 2
\end{aligned}$$

□

Now we will move on to showing that y has an L_p norm similar to the L_p norm of x. We wish to understand how many elements are on the same order as the L_p norm. Therefore, we will ask:

What's the expected number of y_i such that $|z_i| > \frac{\|x\|_p}{c \log n}$, i.e. z_i is "heavy"?

We will show that there will not be many. Let $\|x\|_p \triangleq T$ and $l \triangleq c \log n$.

Claim 4. The expected number of i such that $|z_i| > \frac{T}{l}$ is at most l^p.

Proof. Let E_i be an indicator r.v. for event $|z_i| > \frac{T}{l}$. Recall that $z_i = \frac{x_i}{u_i^{1/p}}$. We want to bound

$$\begin{aligned}
\mathbb{E} \sum_i E_i &= \sum_i \mathbb{P}\left(|z_i| > \frac{T}{l}\right) \\
&= \sum_i \mathbb{P}\left(\frac{|x_i|^p}{u_i} > \frac{T^p}{l^p}\right) \\
&= \sum_i \mathbb{P}\left(u_i < \frac{l^p |x_i|^p}{T^p}\right) \\
&= \sum_i \left(1 - e^{-\frac{l^p |x_i|^p}{T^p}}\right) \\
&\leq \sum_i \frac{l^p |x_i|^p}{T^p} \text{ since } 1 - x \leq e^{-x} \\
&= \frac{T^p}{T^p} l^p = l^p
\end{aligned}$$

□

(Note: By Markov, we know that the number of i such that $|z_i| > \frac{T}{l}$ is at most $O\left(l^p\right)$ with probability $\frac{99}{100}$.)

This can be used to interpret the expected number of "heavy" coordinates in z, which will be something like $\log n^p$. Meanwhile, the space we're using in our algorithm, m, is polynomial in n. If you're hashing poly-$\log n$ coordinates to poly-n coordinates you won't get any collisions, meaning that all the "heavy" coordinates in z will not get mapped to the same bucket. However, this requires perfect hashing.

2.3 Digression on Perfect Hashing

Consider (for this subsection) a universe of size n, and m buckets that we're hashing into. We have a hash function $h : [n] \to [m]$. If $m < n$ they will collide.

There is some subset S of the n nodes which we care about, and we say that S is **perfectly hashed** by h if $\forall i \neq j \in S$, $h(i) \neq h(j)$.

Claim 5. As long as $m \gg |S|^2$ (where $|S|$ is the size of S), and h is 2-wise independent, S is perfectly hashed with good probability, i.e. the success probability grows as c grows in $m = c|S|^2$.

Proof. Define $\binom{n}{2}$ indicator r.v.'s. The i, j'th random variable is 1 if $h(i)$ and $h(j)$ collided and 0 otherwise. Consider the expected sum of these random variables; as long as $m > \binom{n}{2}$ it is less than 1, and as m grows it becomes much less than 1. The rest is proven by Markov. □

2.4 Finishing proof of Theorem 1

Recall: $i \in [n]$ is considered "heavy" if $|z_i| > \frac{T}{c \log n}$.

Intuitively, we're hashing poly-log into poly so it's perfectly hashed with good probability.

So this implies that "heavy" coordinates are all perfectly hashed with probability $\geq \frac{99}{100}$.

> What's the "obsession" with $\frac{99}{100}$? By union bound, we can sum up the failure probability of every event in the algorithm, and if it's less than $\frac{1}{2}$ (which it will be much less than) we'll be fine.

Now consider the m buckets. Some of these buckets receive a heavy i. If we had no non-heavy i, none of these heavy-resident buckets would have non-heavy residents. But we can limit the "damage" the non-heavy residents do by utilizing the random signs.

Let L be the set of light coordinates ($L = [n] \backslash \text{heavy}$).

So the number of light residents in a heavy-resident bucket is $\sum_{i \in L, h(i)=j} \sigma_i \cdot z_i$.

- Consider $j \in [m]$. Then $\mathbb{E} \sum_{i \in L, h(i)=j} \sigma_i \cdot z_i = 0$. What about the variance?

$$\mathbb{E}_{h,\sigma} \sum_{i \in L, h(i)=j} \sigma_i^2 z_i^2 = \mathbb{E}_{h,\sigma} \sum_{i \in L} \delta_{ij} z_i^2 \text{ because } \sigma_i^2 = 1 \text{and where } \delta_{ij} = \begin{cases} 1, & h(i) = j \\ 0, & \text{else} \end{cases}$$

$$= \sum_{i \in L} (\mathbb{E}\delta_{ij})\, z_i^2 = \sum_{i \in L} \frac{z_i^2}{m} \leq \frac{\|z\|_2^2}{m}$$

- $\mathbb{E}\, \|z\|_2^2 = \sum_{i=1}^n \underbrace{x_i^2 \cdot \mathbb{E} \frac{1}{u_i^{2/p}}}_{z_i^2} = \sum_{i=1}^n x_i^2 \cdot O(1)$

- because $\mathbb{E} \frac{1}{u_i^{2/p}} = \int_0^\infty e^{-\lambda} \frac{1}{\lambda^{2/p}} d\lambda = \underbrace{\int_0^1 e^{-\lambda} \frac{1}{\lambda^{2/p}} d\lambda}_{\leq \frac{1}{\lambda^{2/p}} = \frac{1}{\lambda^{1-j}} \text{ for } p > 2} + \underbrace{\int_1^\infty \cdots}_{\leq e^{-\lambda}} = O\left(\|x\|_2^2\right)$

Theorem 6. *Holder's inequality:* $\sum_i f_i g_i \leq \left(\sum_i |f_i|^p\right)^{1/p} \cdot \left(\sum_i |g_i|^q\right)^{1/q}$ *as long as* $\frac{1}{p} + \frac{1}{q} = 1$ *and* $p, q > 0$.

Therefore:

$$\|x\|_2^2 = \sum_i x_i^2 \cdot 1$$

$$\leq \underbrace{\left(\sum_i \left(x_i^2\right)^{p/2}\right)}_{= \|x\|_p^2} \cdot \left(\sum_i 1\right)^{1 - \frac{2}{p}}$$

$$= \|x\|_p^2 \cdot n^{1 - \frac{2}{p}}$$

Putting everything together:

- Noise in a bucket has expectation 0 and has variance $\leq \frac{n^{1-\frac{2}{p}} \cdot T^2}{m}$
- Choose $m \gg n^{1-\frac{2}{p}} \cdot \log n$. Then the variance is $\leq \frac{c' \cdot T^2}{\log n}$
- Then $\mathbb{P}\left(|\text{noise}| > \frac{T}{c \log n} \cdot \frac{1}{10}\right) \leq \frac{10^2 c^2 \log^2 n}{T^2} \cdot \frac{c' \cdot T^2}{\log n}$, canceling the T^2.
- This won't interfere with the heavy components, but since we already have bounds on $\|z\|_\infty$, we really just need to think about collisions within buckets.
  - So really make that $\mathbb{P}\left(|\text{noise}| > \frac{T}{c} \cdot \frac{1}{10}\right) \leq \frac{10^2 c^2}{T^2} \cdot \frac{c' \cdot T^2}{\log n}$
- The L_∞ norm of z is relatively correct. We want to make sure the buckets that a heavy hashes into has little enough noise, and that the non-heavy-resident buckets don't have enough noise to masquerade as heavy-resident buckets.
- $\text{Var}\,(\text{noise}) \leq \frac{n^{1-2/p}}{m} \leq \frac{T^2}{c \log n}$
- $$\mathbb{P}\left(\left|\underbrace{\text{noise}}_{\sum_{i=1}^n \delta_{ij}\sigma_i z_i}\right| > \frac{T}{10}\right) \leq \max\left\{ e^{-\underbrace{\frac{T^2}{10^2}}_{\lambda^2} \cdot \underbrace{\frac{c \log n}{T^2}}_{o^2}}, \frac{\frac{T^2}{c \log n}}{\frac{T^2}{10}} \cdot \frac{c \log n}{T} \right\} \leq \left(\tfrac{1}{2}\right) c \frac{T}{10} \cdot \frac{c \log n}{T}$$

By union bound on noise (failure probabilities), Q.E.D. for Theorem 1.

3 Heavy Hitters

The heavy hitter statistics of a turnstile stream are:

- l_1 point query[3]: given i, output $x_i \pm \varepsilon \cdot \|x\|_1$.
- l_1 heavy hitters: output $L \subseteq [n]$ such that:
  - if $|x_i| \geq \varepsilon \cdot \|x\|_1$, then we must include $i \in L$, and
  - if $i \in L$, then $|x_i| \geq \frac{\varepsilon}{2} \cdot \|x\|_1$

3.1 Linear sketch for deterministic point query

- $y = \Pi x$, $\Pi \in \mathbb{R}^{m \times n}$
- $\tilde{x} = \Pi^T y = \Pi^T \Pi x$
- Estimate x_i as $\langle \Pi^i, y \rangle$

Claim 7. $\tilde{x} = x_i \pm \varepsilon \cdot \|x\|_1$ if Π is ε-incoherent.

[3]These statistics are called l_1 because the error depends on the L_1-norm.

Definition 8. Π is **ε-incoherent** if $\forall i$, $\|\Pi_i\|_2 = 1$ and $\forall i \neq j$, $\left|\left\langle \Pi^i, \Pi^j \right\rangle\right| \leq \varepsilon$.

How do we get ε-incoherent Π? One way: error-correcting codes.

Definition 9. A **code** is a set of vectors $\mathcal{C} = \{C_1, ..., C_n\}$.

- $C_i \in [q]^t$ where q is the "alphabet size" and t is the "block length"
- $d \triangleq \min_{i \neq j} \Delta(C_i, C_j)$ where $\Delta(C_i, C_j)$ are the "coordinates of disagreement", a Hamming distance.

Let the height of Π be $m = q \cdot t$, and get the i'th column of Π from C_i via:

- Break it into t blocks of length q. Each block will be a vector consisting of a single 1 and the rest 0's.
  - If we consider $C_i = (\alpha_1, ..., \alpha_t)$, and $\alpha_j \in [q]$, place the 1 at coordinate α_j within this q-long block, and 0 everywhere else.
- Multiply the entire thing by $\frac{1}{\sqrt{t}}$ to satisfy the unit-norm property of ε-incoherence.

We can observe that this Π is ε-incoherent with $\varepsilon = 1 - \frac{d}{t}$.

Claim 10. Π is ε-incoherent $\Rightarrow |\tilde{x}_i - x_i| \leq \varepsilon \cdot \|x\|_1$

Proof.

$$\begin{aligned}
\tilde{x}_i &= \left(\Pi^i\right)^T \Pi x \\
&= \left\langle \Pi^i, \sum_{j=1}^{n} x_j \Pi_j \right\rangle \\
&= \left\langle \Pi^i, \Pi^i \right\rangle x_i + \sum_{j \neq i} \underbrace{\left\langle \Pi^i, \Pi^j \right\rangle}_{\leq \varepsilon \text{ in magnitude}} x_j \\
&= x_i + \sum_j (\pm\varepsilon) \cdot x_j \\
&= x_i \pm \varepsilon \|x\|_1
\end{aligned}$$

□

References

[Andoni12] Alexandr Andoni. High frequency moments via max-stability. Manuscript, 2012. http://www.mit.edu/∼andoni/papers/fkStable.pdf

[BJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.* 68(4): 702-732, 2004. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.1160&rep=rep1&type=pdf

[IW05] Piotr Indyk, David P. Woodruff. Optimal approximations of the frequency moments of data streams. STOC 2005: 202-208. http://dl.acm.org/citation.cfm?id=1060621

[Indyk06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM* 53(3): 307–323, 2006. http://dl.acm.org/citation.cfm?id=1147955

[NNW12] Jelani Nelson, Huy L. Nguyen, David P. Woodruff. On Deterministic Sketching and Streaming for Sparse Recovery and Norm Estimation. *Proceedings of the 16th International Workshop on Randomization and Computation (RANDOM 2012)*, pgs. 627-628, 2012. http://people.seas.harvard.edu/∼minilek/papers/sparse_recov.pdf

[PDGQ05] Rob Pike, Sean Dorward, Robert Griesemer, Sean Quinlan: Interpreting the data: Parallel analysis with Sawzall. Scientific Programming 13(4): 277-298 (2005). http://iospress.metapress.com/content/99vjkgkae3jkvu9t/

[Tao10] Terence Tao. 254A, Notes 1: Concentration of measure. Website. 3 January 2010. Last accessed 12 September 2013. http://terrytao.wordpress.com/2010/01/03/254a-notes-1-concentration-of-measure/