CS 229r: Algorithms for Big Data Fall 2015

Lecture 5 — September 17, 2015

Prof. Jelani Nelson *Scribe: Yakir Reshef*

1 Recap and overview

Last time we discussed the problem of norm estimation for p-norms with $p > 2$. We had described an algorithm by [Andoni12] that, given $x \in \mathbb{R}^n$ updated under a turnstile model, approximates $\|x\|_p$ with constant multiplicative error. The algorithm generates two random matrices $P \in \mathbb{R}^{m\times n}$ (with $m \ll n$) and $D \in \mathbb{R}^{n\times n}$. P is sampled so that each of its columns contains all zeros except for one entry, which contains a random sign. D is a diagonal matrix whose i-th diagonal entry is $u_i^{-1/p}$ where the u_i are i.i.d. exponential random variables. The algorithm then maintains $y = PDx$, and its output is $\|y\|_\infty = \max_i |y_i|$.

In this lecture we will complete the proof of correctness of this algorithm and then move on from p-norm estimation to other problems related to linear sketching.

2 Completing the proof of correctness

From last time we have the following claim.

Claim 1. *Let $Z = DX$. Then*

$$P\left(\frac{1}{2}\|x\|_p \le \|Z\|_\infty \le 2\|x\|_p\right) \ge 3/4$$

This claim establishes that if we could maintain Z instead of y then we would have a good solution to our problem. Remember though that we can't store Z in memory because it's n-dimensional and $n \gg m$. That's why we need to analyze $PZ \in \mathbb{R}^m$.

2.1 Overview of analysis of $y = PZ$

The idea behind our analysis of $y = PZ$ is as follows: each entry in y is a sort of counter. And the matrix P takes each entry in Z, hashes it to a perfectly random counter, and adds that entry of Z times a random sign to that counter. Since $n > m$ and there are only m counters, there will be collisions, and these will cause different Z_i to potentially cancel each other out or add together in a way that one might expect to cause problems. We'll get around this by showing that there are very few large Z_i's, so few relative to m that with high probability none of them will collide with each other.

We still need to worry, because small Z_i's and big Z_i's might collide with each other. But remember that when we add the small Z_i's, we multiply them with a random sign. So the expectation of the aggregate contributions of the small Z_i's to each bucket is 0. We'll bound their variance as well, which will show that if they collide with big Z_i's then with high probability this won't substantially change the relevant counter. All of this together will show that the maximal counter value (i.e., $\|y\|_\infty$) is close to the maximal Z_i – and therefore to $\|x\|_p$ – with high probability

2.2 Analysis of $y = PZ$

We make the following definitions.

- Let $T = \|x\|_p$.
- Define the "heavy indices" as $H = \{j : |Z_j| \geq T/(v \lg(n))\}$. Think of c as big. We'll set it later.
- Define the "light indices" as $L = [n] \backslash H$.

2.2.1 Analyzing the heavy indices

We begin by showing that there will not be many heavy indices.

Claim 2. *For any $\ell > 0$, we have*

$$E\left(\left|\left\{i \in [n] : |Z_i| > \frac{T}{\ell}\right\}\right|\right) < \ell^p$$

Before we prove this claim, let's reflect: if $\ell = v \lg(n)$ then we get polylog(n) heavy indices, which is miniscule compared to the $m = O(n^{1-2/p} \ln(n))$ counters. Birthday paradox-type reasoning will then translate this bound into the idea that with high probability there will not be collisions between big Z_j.

Proof. Let

$$Q_i = \begin{cases} 1 & |z_i| > T/\ell \\ 0 & \text{else} \end{cases}$$

so that the number of indices with $|Z_i| > T/\ell$ equals $\sum Q_i$. We then have

$$\begin{aligned} E\left(\sum_i Q_i\right) &= \sum_i E(Q_i) \\ &= \sum_i P\left(|x_i/u_i^{1/p}| > T/\ell\right) \\ &= \sum_i P\left(u_i < |x_i|^p \ell^p / T^p\right) \\ &= \sum_i (1 - e^{-|x_i|^p \ell^p / T_p}) && (u_i \text{ exponentially distributed}) \\ &\leq \sum_i \ell^p |x_i|^p / T^p && (1 + x \leq e^x \text{ for } x \in \mathbb{R}) \\ &= \ell^p && \textstyle\sum_i |x_i|^p = \|x\|_p^p = T^p \end{aligned}$$

which completes the proof. □

2.2.2 Recalling Bernstein's inequality

To analyze the light indices, we'll need to recall *Bernstein's inequality.*

Theorem 1 (Bernstein's inequality)**.** *Suppose $R_1, \dots, R_n$ are independent, and for all i, $|R_i| \leq K$, and var$(\sum_i R_i) = \sigma^2$. Then for all $t > 0$*

$$P\left(\left|\sum_i R_i - E\left(\sum_i R_i\right)\right| > t\right) \lesssim e^{-ct^2/\sigma^2} + e^{-ct/K}$$

2.2.3 Analyzing the light indices

We now establish that the light indices together will not distort any of the heavy indices by too much. Before we write down our specific claim, let's parametrize P as follows. We have a function $h : [n] \to [m]$ as well as a function $\sigma : [n] \to \{-1, 1\}$ that were both chosen at random. (One can show that these can be chosen to be k-wise independent hash functions, but we won't do so in this lecture.) We then write

$$P_{ij} = \begin{cases} \sigma(j) & \text{if } h(j) = i \\ 0 & \text{else.} \end{cases}$$

So essentially, h tells us which element of the column to make non-zero, and σ tells us which sign to use for column j.

We can now write our claim about the light indices.

Claim 3. *It holds with constant probability that for all $j \in [m]$,*

$$\left|\sum_{j \in L : h(j) = i} \sigma(j) Z_j\right| < T/10.$$

Let us see how this claim completes our argument. It means that

- If y_i didn't get any heavy indices then the magnitude of y_i is much less than T, so it won't interfere with our estimate.
- If y_i got assigned the maximal Z_j, then by our previous claim that is the only heavy index assigned to y_i. In that case, this claim means that all the light indices assigned to y_i won't change it by more than $T/10$, and since Z_j is within a factor of 2 of T, y_i will still be within a constant multiplicative factor of T.
- If y_i got assigned some other heavy index, then the corresponding Z_j is by definition is less than $2T$ since it's less than the maximal Z_j. In that case, this claim again tells us that y_i will be at most $2.1T$.

To put this more formally:

$$\begin{aligned} y_i &= \sum_{j : h(j) = i} \sigma(j) Z_j \\ &= \sum_{j \in L : h(j) = i} \sigma_(j) Z_j + \sigma(j_{heavy}) Z_{j_{heavy}} \end{aligned}$$

where the second term is added only if y_i got some heavy index, in which case we can assume it received at most one. The triangle inequality then implies that

$$\begin{aligned}|y_i| &\in Z_{j_{heavy}} \pm \left| \sum_{j \in L : h(j) = i} \sigma_(j) Z_j \right| \\ &= Z_{j_{heavy}} \pm T/10\end{aligned}$$

Applying this to the bucket that got the maximal z_i then gives that that bucket of y should contain at least $0.4T$. And applying this to all other buckets gives that they should contain at most $2.1T$.

Let us now prove the claim.

Proof of Claim 3. Fix $i \in [m]$. We use Bernstein on the sum in question. For $j \in L$, define

$$\delta_j = \begin{cases} 1 & \text{if } h(j) = i \\ 0 & \text{else.} \end{cases}$$

Then the sum we seek to bound equals

$$\sum_{j \in L} \delta_j \sigma(j) Z_j$$

We will call the j-th term of the summand R_j and then use Bernstein's inequality. The brunt of the proof will be computing the relevant quantities to see what the inequality gives us. First, the easy ones:

1. We have $E(\sum R_j) = 0$, since the $\sigma(j)$ represent random signs.
2. We also have $K = T/(v \lg(n))$ since $|\delta_j| \le 1$, $|\sigma(j)| \le 1$, and we only iterate over light indices so $|Z_j| \le T/(v \lg(n))$.

It remains only to compute $\sigma^2 = \text{var}(\sum_j R_j)$. If we condition on Z, then a problem from problem set 1 implies that

$$\text{var}\left(\sum_j R_j | Z \right) \le \frac{\|Z\|_2^2}{m}$$

This isn't enough of course: we need to get something that takes the randomness of Z into account. However, instead of computing the unconditional variance of our sum, we will prove that σ^2 is small with high probability over the choice of Z. We'll do this by computing the unconditional expectation of σ^2 and then using Markov. We write

$$E\left(\|Z\|_2^2\right) = \sum_j x_j^2 E\left(\frac{1}{u_j^{2/p}} \right)$$

and

$$\begin{aligned}
E\left(\frac{1}{u_j^{2/p}}\right) &= \int_0^\infty e^{-x}(x^{-2/p})dx \\
&= \int_0^1 e^{-x}(x^{-2/p})dx + \int_1^\infty e^{-x}\cdot(x^{-2/p})dx \\
&= \int_0^1 x^{-2/p}dx + \int_1^\infty e^{-x}dx. && \text{(trivial bounds on } e^{-x} \text{ and } x^{-2/p}\text{)}
\end{aligned}$$

The second integral trivially converges, and the former one converges because $p > 2$. This gives that

$$E(\|Z\|^2) = O(\|x\|_2^2)$$

which gives that with high probability we will have $\sigma^2 \le O(\|x\|_2^2)/m$.

To use Bernstein's inequality, we'll want to relate this bound on σ^2, which is currently stated in terms of $\|x\|_2$, to a bound in terms of $\|x\|_p$. We will do this using a standard argument based on Hölder's inequality, which we re-state without proof below.

Theorem 2 (Hölder's inequality)**.** *Let* $f, g \in \mathbb{R}^n$*. Then*

$$\sum_i f_i g_i \le \|f\|_a \|g\|_b$$

for any $1 \le a, b \le \infty$ *satisfying* $1/a + 1/b = 1$*.*

Setting $f_i = x_i^2$, $g_i = 1$, $a = p/2$, $b = 1/(1-a)$ then gives

$$\begin{aligned}
\|x_i\|^2 &= \sum_i f_i g_i \\
&\le \left(\sum_i (x_i^2)^{p/2}\right)^{2/p} \left(\sum_i 1^{1/(1-2/p)}\right)^{1-2/p} && \text{(Hölder)} \\
&\le \|x\|_p^2 \cdot n^{1-2/p}
\end{aligned}$$

Using the fact that we chose m to $\Theta(n^{1-2/p}\lg(n))$, we can then obtain the following bound on σ^2 with high probability.

$$\begin{aligned}
\sigma^2 &\le O\left(\frac{\|x\|_2^2}{m}\right) \\
&\le O\left(\frac{T^2 n^{1-2/p}}{m}\right) && \text{(Hölder trick)} \\
&\le O\left(\frac{T^2 n^{1-2/p}}{n^{1-2/p}\lg n}\right) && \text{(choice of } m\text{)} \\
&\le O\left(\frac{T^2}{\lg(n)}\right)
\end{aligned}$$

We now need to apply Bernstein's inequality and show that it gives us the desired result. Initially, the inequality gives us the following guarantee.

$$\begin{aligned} P\left(\left|\sum R_i\right| > T/10\right) &\lesssim e^{-cT^2/100 \cdot O(\lg(n)/T^2)} + e^{-cT/10 \cdot (v \lg(n)/T)} \\ &\leq e^{-C \lg(n)} && \text{(for some new constant } C\text{)} \\ &= n^C \end{aligned}$$

So the probability that the noise at most $T/10$ can be made poly n. But there are at most n buckets, which means that a union bound gives us that with at least constant probability all of the light index contributions are are at most $T/10$. □

3 Wrap-up

Thus far we presented algorithms for p-norm estimation for $p \leq 2$, $p = 2$, and $p > 2$ separately. (Of course, the $p \leq 2$ can be used for $p = 2$ as well.) We noticed that at $p = 2$ there seems to be a critical point above which we appeared to need a different algorithm. Later in the course we'll see that there are space lower-bounds that say that once $p > 2$ we really do need as much space as the algorithm we presented for $p > 2$ required.

We conclude our current treatment of norm estimation and approximate counting by briefly noting some motivating applications for these problems. For example, distinct elements is used in SQL to efficiently count distinct entries in some column of a data table. It's also used in network anomaly detection to, say, track the rate at which a worm is spreading: you run distinct elements on a router to count how many distinct entities are sending packets with the worm signature through your router. Another example is: how many distinct people visited a website? For more general moment estimation, there are other motivating examples as well. Imagine x_i is the number of packets sent to IP address i. Estimating $\|x\|_\infty$ would give an approximation to the highest load experienced by any server. Of course, as we just mentioned, $\|x\|_\infty$ is difficult to approximate in small space, so in practice people settle for the closest possible norm to the ∞-norm, which is the 2-norm. And they do in fact use the 2-norm algorithm developed in the problem set for this task.

4 Some setup for next time

Next time we'll talk about two new, related problems that sites like Google trends solve. They are called the heavy hitters problem and the point query problem.

In Point Query, we're given some $x \in \mathbb{R}^n$ updated in a turnstile model, with n large. (You might imagine, for instance, that x has a coordinate for each string your search engine could see and x_i is the number of times you've seen string i.) We seek a function query(i) that, for $i \in [n]$, returns a value in $x_i \pm \varepsilon \cdot \|x\|_1$.

In Heavy Hitters, we have the same x but we seek to compute a set $L \subset [n]$ such that

1. $|x_i| \geq \varepsilon \|x\|_1 \Rightarrow i \in L$
2. $|x_i| < \frac{\varepsilon}{2} \|x\|_1 \Rightarrow i \notin L$

As an observation: if we can solve Point Query with bounded space then we can solve Heavy Hitters with bounded space as well (though not necessarily efficient run-time). To do this, we just run Point Query with $\varepsilon/10$ on each $i \in [n]$ and output the set of indices i for which we had large estimates of x_i.

4.1 Deterministic solution to Point Query

Let us begin a more detailed discussion of Point Query. We begin by defining an *incoherent matrix.*

Definition 1. $\Pi \in \mathbb{R}^{m \times n}$ *is* ε-incoherent *if*

1. *For all* i, $\|\Pi_i\|_2 = 1$
2. *For all* $i \neq j$, $|\langle \Pi_i, \Pi_j \rangle| \leq \varepsilon$.

We also define a related object: a *code.*

Definition 2. *An* (ε, t, q, N)*-code is a set* $\mathcal{C} = \{C_1, \ldots, C_N\} \subseteq [q]^t$ *such that for all* $i \neq j$, $\Delta(C_i, C_j) \geq (1 - \varepsilon)t$*, where* Δ *indicates Hamming distance.*

The key property of a code can be summarized verbally: any two distinct words in the code agree in at most εt entries.

There is a relationship between incoherent matrices and codes.

Claim 4. *Existence of an* (ε, t, q, n)*-code implies existence of an* ε*-incoherent* Π *with* $m = qt$.

Proof. We construct Π from $\mathcal{C}$. We have a column of Π for each $C_i \in \mathcal{C}$, and we break each column vector into t blocks, each of size q. Then, the j-th block contains binary string of length q whose a-th bit is 1 if the j-th element of C_i is a and 0 otherwise. Scaling the whole matrix by $1/\sqrt{t}$ gives the desired result. □

We'll start next time by showing the following two claims.

Claim 5 (to be shown next time)**.** *Given an* ε*-incoherent matrix, we can create a linear sketch to solve Point Query.*

Claim 6 (shown next time)**.** *A random code with* $q = O(1/\varepsilon)$ *and* $t = O(\frac{1}{\varepsilon} \log N)$ *is an* (ε, t, q, N)*-code.*

References

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