CS 15-859: Algorithms for Big Data Fall 2019

Lecture 7 — Oct.17 (Part 2)

Prof. David Woodruff *Scribe: Vaidehi Srinivas*

Estimating Norms in the Streaming Model

Problem Setup

We define the p-norm of a vector $x \in \mathbb{R}^n$ to be

$$|x|_p^p = \sum_{i=1}^{n} |x_i|^p.$$

In this problem, we want output a Z such that

$$(1-\varepsilon)|x|_p^p \leq Z \leq (1+\varepsilon)|x|_p^p$$

with probability at least $\frac{9}{10}$.

Motivation

- $p = 1$ corresponds to the total variation distance between distributions.

  Say we have two distributions p and q. (By definition, they satisfy $\sum_i p_i = 1$, $p_i > 0$, $\sum_i q_i = 1$, $q_i > 0$.)

  Then, we have that:

  $$|p - q|_1 = 2 \cdot D_{\text{TV}}(pq) = 2 \cdot \max_{\text{events } E} |p(E) - q(E)|$$

- $p = 2$ is useful for geometric and linear algebraic problems, as we have seen.
- $p = \infty$ is the value of the maximum entry, which is useful for anomaly detection. In this case we will be trying to estimate $|x|_\infty$ not $|x|_\infty^\infty$.

1 Euclidean Norm (2-Norm)

We basically did this problem in the first part of the lecture with the $x = 0^n$ problem.

So we want a Z for which

$$(1-\varepsilon)|x|_2^2 \leq Z \leq (1+\varepsilon)|x|_2^2.$$

Algorithm

We do this by sampling a random CountSketch matrix S with $s \in \frac{1}{\varepsilon^2}$ rows.

Now, we will keep track of S and Sx. If we get an update Δ_j, we know that x updates according to

$$x \leftarrow x + \Delta_j \cdot e_j$$

so we can update according to

$$Sx \leftarrow Sx + S \cdot \Delta_j \cdot e_j.$$

Since we know that CountSketch preserves norms, we know that at the end of the stream

$$|Sx|_2^2 \in (1 \pm \varepsilon)|x|_2^2$$

with probability at least $\frac{9}{10}$.

Space Complexity

We can store S efficiently using limited independence. This takes $\mathcal{O}(\log n)$ bits of space.

We also need to store Sx. Sx has $s = \frac{1}{\varepsilon^2}$ entries, that each take $\mathcal{O}(\log n)$ bits of space to store.

All in all, this algorithm takes

$$\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log n\right)$$

bits of space.

2 1-Norm

Again we want to output Z such that

$$(1 - \varepsilon)|x|_1 \leq Z \leq (1 + \varepsilon)|x|_1.$$

Algorithm

We sample a random dense Cauchy matrix S with $\frac{1}{\varepsilon^2}$ rows. That is, we draw each element independently from a Cauchy distribution C.

As always, we keep track of Sx and update according to

$$Sx \leftarrow Sx + \Delta_i S_{*,i}.$$

Ok, so what do we want to output at the end of the algorithm? Just pattern matching, we might guess that we want to output $|Sx|_1$. But what is Sx?

We know that

$$(Sx)_i = \langle S_i, x \rangle$$

where S_i is a vector of i.i.d. variables drawn from C. Last lecture, we saw that Cauchy random variables are 1-stable. This means that

$$\langle S_i, x \rangle \sim |x|_1 \cdot C.$$

So what would happen if we took $|Sx|_1$? We would be summing up copies of $|x|_1 \cdot |C|$. Since Cauchy random variables have no concentration, we would get results all over the place.

Instead, we'll output the **median** of the absolute value of the elements of Sx. The probability density function (p.d.f.) $f(x)$ of $|C|$ for a Cauchy random variable C is

$$f(x) = \frac{2}{\pi(1+x^2)}.$$

This means that the cumulative distribution function (c.d.f.) of $|C|$ is

$$F(z) = \int_0^z f(x)dx = \frac{2}{\pi} \arctan(z).$$

We know that $\arctan\left(\frac{\pi}{4}\right) = 1$ so $F(1) = \frac{1}{2}$, so the median value of $|C|$ is 1.

So, if we take $r = \frac{\log\left(\frac{1}{\delta}\right)}{\varepsilon^2}$ independent samples $X_1, \ldots, X_r$ from F, and $X = \text{median}_i X_i$ then we know that

$$F(X) \in \left[\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon\right]$$

with probability $1 - \delta$.

This means that

$$F^{-1}(X) = \tan\left(\frac{X\pi}{2}\right) \in [1 - 4\varepsilon, 1 + 4\varepsilon].$$

This shows that outputting the median of the entries of Sx will result in an $(1 \pm O(\varepsilon))$ approximation of $|x|_1$, as we want.

Space Complexity

We can store S with $\frac{1}{\varepsilon}$ words of space [Kane, Nelson, W].

Keeping track of Sx takes $\frac{1}{\varepsilon^2}$ words of $\mathcal{O}(\log n)$ bits each.

In all, we get $\mathcal{O}\left(\frac{\log n}{\varepsilon^2}\right)$ bits of space complexity.

3 p-Norm for $0 < p < 2$

This is similar to 1-Norm estimation, and will rely on p-stable distributions, which only exist for $0 < p < 2$.

p-stable distributions

There is no closed form expression for the probability distribution function for the p-stable distribution, but we can efficiently sample them.

If we sample

$$\Theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$
$$r \in [0, 1]$$

uniformly randomly, then

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

is a sample from a p-stable distribution.

Algorithm

So the algorithm here will be basically the same as the one for the 1-Norm estimator. We can draw a random sketching matrix S from the p-stable distribution. Then we can discretize them and store a sketching matrix of samples using limited independence.

Then the rest of the proof carries through. This means that for each $0 < p < 2$ there is an algorithm that works, and uses $\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log n\right)$ bits of space.

4 p-Norm for $p > 2$

For $p > 2$, p-stable distributions do not exist, so unfortunately, we can't do the same thing we did for the $0 < p < 2$ case.

Lower Bound

Later in the course, we will show that we need $\Omega\left(n^{1-\frac{2}{p}}\right)$ bits of space to approximate p-norms, $p > 2$, up to a constant factor with constant probability.

Algorithm

We are going to give an algorithm that uses $\tilde{O}\left(n^{1-\frac{2}{p}}\right)$ bits of space. $\tilde{O}$ means that the notation is swallowing log factors.

We are going to use the sketch $P \cdot D$:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & -1 & \dots & 0 \\ & & & & \dots & & & & \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{E_1^{\frac{1}{p}}} & & & & \\ & \frac{1}{E_2^{\frac{1}{p}}} & & & \\ & & & \dots & \\ & & & & \frac{1}{E_n^{\frac{1}{p}}} \end{bmatrix}$$

Here the Es are standard exponential random variables. Note that this does not depend on x, which is important if we want to use it in a streaming setting, since we don't know what x is ahead of time.

Our analysis is going to assume true randomness, but in real life we would want limited independence to save on space.

We are also going to ignore discretization in our analysis.

Exponential Random Variables

An exponential random variable is defined with respect to a parameter λ. We'll call it a standard exponential if $\lambda = 1$.

The probability density function (p.d.f.), f, is

$$f(x) = \lambda e^{-\lambda x}$$

for $x \geq 0$, and 0 otherwise.

We can integrate this to get that the cumulative density function (c.d.f.), F, is

$$F(x) = 1 - e^{-\lambda x}$$

for $x \geq 0$ and 0 otherwise.

Also, for a scalar $t \geq 0$, $t \cdot E$ has c.d.f.

$$F(x) = 1 - e^{-\frac{\lambda}{t}x}.$$

This also means that an exponential random variable with parameter λ is the same as $\frac{E}{\lambda}$ for a standard exponential E.

Stability of Exponential Random Variables

We can think of exponential random variables are being "min. stable." That is, consider independent standard exponential random variables $E_1, \dots, E_n$ and fixed scalars $|y_1|, \dots, |y_n|$. Now let

$$q = \min\left(\frac{E_1}{|y_1|^p}, \dots, \frac{E_n}{|y_n|^p}\right).$$

Now, we can see that

$$\begin{aligned}
\mathrm{P}[q \geq x] &= \mathrm{P}\left[\forall i, \frac{E_i}{|y_i|^p} \geq x\right] \\
&= \sum_i e^{-x|y_i|^p} && \text{since the } E_i\text{s are indep.} \\
&= e^{-x|y|_p^p}
\end{aligned}$$

which means that q is an exponential random variable with $\lambda = |y|_p^p$. That is, $q \sim \frac{1}{|y|_p^p} \cdot E$ for a standard exponential variable E.

Looking at $|Dy|_\infty$

We'll show that we can get a good estimate for $|y|_p$ by looking at $|Dy|_\infty$.

We can see that

$$|Dy|_\infty^p = \max_i \left(\frac{|y_i|^p}{E_i}\right) = \frac{1}{\min_i \frac{E_i}{|y_i|^p}} = \frac{1}{E \cdot \frac{1}{|y|_p^p}} = \frac{|y|_p^p}{E}$$

$$|Dy|_\infty = \frac{|y|_p}{E^{\frac{1}{p}}}$$

Now we have this in terms of a standard exponential. Now, we can bound the probability that E is in a certain range:

$$\begin{aligned}
\mathrm{P}\left[E \in \left[\frac{1}{10}, 10\right]\right] &= F(10) - F\left(\frac{1}{10}\right) \\
&= \left(1 - e^{-10}\right) - \left(1 - e^{-\frac{1}{10}}\right) \\
&> \frac{4}{5}
\end{aligned}$$

So this means that

$$|Dy|_\infty \in \left[\frac{|y|_p}{10^{\frac{1}{p}}}, 10^{\frac{1}{p}}|y|_p\right]$$

with probability at least $\frac{4}{5}$.

So this is good, because we got a good estimate of $|y|_p$, but keeping track of Dy doesn't save us anything, since Dy is an n-dimensional vector. This is why we want to sketch Dy with P.

Understanding $|PDy|_\infty$

Setup

Intuitively, we want to say that P is hashing coordinates of Dy into random buckets, and taking a signed sum of the entries. We hope that everything will cancel and $|PDy|_\infty \approx |Dy|_\infty$. The truth, sadly, is slightly messier.

Let s be the number of rows of P. Think of these as hash-buckets. P is a CountSketch matrix that is defined by

$$h : [n] \to [s]$$

$$\sigma : [n] \to \{-1, 1\}$$

For the analysis, we'll assume that these hash functions are truly random. (They can be derandomized, but we won't analyze that here.)

To achieve $|PDy|_\infty \approx |Dy|_\infty$ with good probability, we want to satisfy two properties:

1. For each bucket i that does not contain the coordinate j for which $|(Dy)_j| = |Dy|_\infty$ we have that
$$|(PDy)_i| \leq \frac{|y|_p}{100}.$$
That is, we want that all of the buckets that don't have the maximum value to be small.

2. For the bucket containing i containing the coordinate j for which $|(Dy)_i| = |Dy|_\infty$, we have that
$$\left||(PDy)_i| - |Dy|_\infty\right| \leq \frac{|y|_p}{100}.$$
This just means that we don't want the bucket with the largest element to have too much noise, since our estimation depends on this value.

Expectation and Variance of Elements of PDy

Ok, so let $\delta(E)$ be an indicator if event E happens.

We know that

$$(PDy)_i = \sum_j \delta(h(j) = i) \cdot \sigma_j \cdot (Dy)_j.$$

That is, $(PDy)_i$ is the signed sum of $(Dy)_j$ for the js that are hashed to the ith bucket.

This is nice, it tells us that

$$\mathbb{E}\left[(PDy)_i\right] = 0.$$

since $\mathbb{E}[\sigma_j] = 0$.

Now what about the variance?

$$\mathop{\mathbb{E}}_{P}\left[(PDy)_i^2\right] = \sum_{j,j'} \mathbb{E}\left[\delta(h(j) = i) \cdot \delta(h(j') = i) \cdot \sigma_j \cdot \sigma_{j'}\right] (Dy)_j (Dy)_{j'}$$

So we know that when $j \neq j'$, $\mathbb{E}\left[\sigma_j \sigma_{j'}\right] = 0$. This means that the above is the same as

$$\sum_j \mathbb{E}\left[\delta(h(j) = i)^2 \cdot \sigma_j^2\right] (Dy)_j^2 = \sum_j \mathbb{E}\left[\delta(h(j) = i)\right] (Dy)_j^2 = \sum_j \frac{1}{s} \cdot (Dy)_j^2 = \frac{1}{s}|Dy|_2^2$$

Ok, now we can see that

$$\mathop{\mathbb{E}}_{D}\left[|Dy|_2^2\right] = \sum_i y_i^2 \cdot \mathbb{E}\left[D_{i,i}^2\right].$$

But what is $\mathbb{E}\left[D_{i,i}^2\right]$? Since $D_{i,i} \sim \frac{1}{E^{\frac{1}{p}}}$, we can get

$$\begin{aligned}
\mathbb{E}\left[D_{i,i}^2\right] &= \int_{t\geq 0} \left(t^{-\frac{1}{p}}\right)^2 \cdot f(t) \\
&= \int_{t\geq 0} t^{-\frac{2}{p}} e^{-t} dt \\
&= \left[\int_{t\in[0,1]} t^{-\frac{2}{p}} e^{-t} dt\right] + \left[\int_{t>1} t^{-\frac{2}{p}} e^{-t} dt\right] \\
&\leq \left[\int_{t\in[0,1]} t^{-\frac{2}{p}} dt\right] + \left[\int_{t>1} e^{-t} dt\right] \\
&= \left(\frac{1}{1-\frac{2}{p}}\right) t^{1-\frac{2}{p}}\Big|_0^1 - e^{-t}\Big|_1^\infty \\
&\in \mathcal{O}(1)
\end{aligned}$$

So this means that

$$\mathbb{E}\left[(PDy)_i^2\right] \in \mathcal{O}\left(\frac{1}{s}\right) \cdot \mathcal{O}(1) \cdot |y|_2^2.$$

But what we really want is an expression in terms of the p-norm of y, not the 2-norm. To get to this, we can use Holder's Inequality.

Holder's Inequality: For p and q such that $\frac{1}{p} + \frac{1}{q} = 1$, and vectors a and b, we have that

$$\langle a, b\rangle \leq |a|_p \cdot |b|_q.$$

(Cauchy-Schwarz is just a special case of this.)

So we want to convert the 2-norm of y into the p-norm of y. So, we can write it as

$$|y|_2^2 = \sum_i y_i^2 \cdot 1.$$

Now, we can set $a_i = y_i$, $b_i = 1$, and choose q such that $\frac{2}{p} + \frac{1}{q} = 1$. This allows us to say

$$\begin{aligned}
\sum_i y_i^2 \cdot 1 &\leq \left(\sum_i \left(y_i^2\right)^{\frac{p}{2}}\right)^{\frac{2}{p}} \cdot \left(\sum_i 1^q\right)^{\frac{1}{q}} \\
&= \left(\sum_i y_i^p\right)^{\frac{1}{p}\cdot 2} \cdot n^{\frac{1}{q}} \\
&= |y|_p^2 \cdot n^{1-\frac{2}{p}}
\end{aligned}$$

Plugging this back into our original expression gives us that

$$\mathbb{E}\left[(PDy)_i^2\right] \in \mathcal{O}\left(\frac{1}{s}\right) \cdot |y|_p^2 \cdot n^{1-\frac{2}{p}}.$$

Since $\mathbb{E}\left[(PDy)_i\right] = 0$, this above expression directly gives us a bound on the variance of $(PDy)_i$.

Bounding the Elements of PDy

To bound this we are going to use Bernstein's bound. This is a tail bound, like some of the other bounds we've seen in class (Matrix Chernoff, Azuma-Hoeffding).

Bernstein's Bound: Suppose we have $R_1, \ldots, R_n$ independent, where for all j, $|R_j| \leq K$, and $\text{Var}\left[\sum_j R_j\right] \leq \sigma^2$.
Then, there exist constants C and c such that for all $t > 0$

$$\text{P}\left[\left|\sum_j R_j - \mathbb{E}\left[\sum_j R_j\right]\right| > t\right] \leq C\left(e^{-\frac{ct^2}{\sigma^2}} + e^{-\frac{ct}{K}}\right).$$

Note that $R_1, \ldots, R_n$ do not have to be drawn from the same distribution.

Recall that

$$(PDy)_i = \sum_i \delta(h(j) = i) \cdot \sigma_j \cdot (Dy)_j.$$

Accordingly, we will set

$$R_j = \delta(h(j) = i) \cdot \sigma_j \cdot (Dy)_j.$$

Note that this has expectation 0, since $\mathbb{E}[\sigma_j] = 0$. This means that

$$\mathbb{E}\left[\sum_j R_j\right] = 0.$$

Also, $\sum_j R_j = (PDy)_i$, so we can set $\sigma^2 = \mathcal{O}\left(\frac{1}{s}\right) n^{1-\frac{2}{p}} \cdot |y|_p^2$.

Again, our goal is to get $|PDy|_\infty \approx |Dy|_\infty$, where we have conditioned on

$$|Dy|_\infty \in \left[\frac{|y|_p}{10^{\frac{1}{p}}}, 10^{\frac{1}{p}}|y|_p\right].$$

Ok, so we want to get a bound on the size of $(PDy)_i$ with error probability $\frac{1}{n^2}$. If we set

$$t = \frac{|y|_p}{100}$$
$$s \in \Theta\left(n^{1-\frac{2}{p}} \log n\right)$$

We get that

$$\text{P}\left[|(PDy)_i - 0| > \frac{|y|_p}{100}\right] \leq C\left(e^{-c\frac{|y|_p^2/100^2}{|y|_p/\log n}} + e^{-c\frac{|y|_p/100}{K}}\right)$$

If we had $K = \frac{|y|_p}{c \log n}$, then this would be great, we'd get a probability bound like

$$Ce^{-\Theta(\log n)} \approx \frac{1}{n^2}.$$

Unfortunately, we explicitly do not have this. $K = \max_j |R_j|$ could be arbitrarily large, which is not good. Bernstein's Bound is still promising if we can somehow finagle our way into being able to use this K. We'll do this by considering the large elements of of Dy separately.

Understanding the Large $(Dy)_i$s

Let's set a sufficiently small constant α. Now we'll call j large if

$$|R_j| > \frac{\alpha |y|_p}{\log n}.$$

Otherwise it is small. Note that if $|R_j| > \frac{\alpha |y|_p}{\log n}$, then necessarily $(Dy)_i > \frac{\alpha |y|_p}{\log n}$ (since $R_j = \delta(h(i) = j) \cdot \sigma_j (Dy)_j$).

Ok, so how many elements can we expect will be large?

Recall that

$$|(Dy)_j| = \frac{|y_j|}{E_j^{\frac{1}{p}}}$$

so we get that

$$\begin{aligned}
\mathrm{P}_D \left[|(Dy)j| \text{ is large} \right] &= \mathrm{P} \left[\frac{|y_j|}{E_j^{\frac{1}{p}}} \geq \frac{\alpha |y|_p}{\log n} \right] \\
&= \mathrm{P} \left[\frac{|y_j|^p}{\alpha^p |y|_p^p} \left(\log^p n \right) \geq E_j \right] \\
&= 1 - e^{-\frac{|y_j|^p \left(\log^p n \right)}{\alpha^p |y|_p^p}} \\
&\leq \frac{|y_j|^p \left(\log^p n \right)}{\alpha^p |y|_p^p}
\end{aligned}$$

The last line follows from the fact that $1 - e^{-x} \leq x$. This tells us that the expected number of large entries is $\mathcal{O}\left(\log^p n \right)$.

By Markov's, we can say that the number of large j is $\mathcal{O}\left(\log^p n \right)$ with constant probability, and we'll condition on D satisfying this. We'll also condition on

$$|Dy|_\infty \in \left[\frac{|y|_p}{10^{\frac{1}{p}}}, 10^{\frac{1}{p}} |y|_p \right]$$

which held with probability at least $\frac{4}{5}$.

We will also assume that all of the large j get perfectly hashed into separate hash buckets. This is fine, since we are throwing $\mathcal{O}(\log^p n)$ balls into $s \geq n^{1 - \frac{2}{p}}$ bins. This means that our collision probability is at most

$$\frac{\log^{2p} n}{n^{1 - \frac{2}{p}}}$$

which is known in the business as "super tiny."

Back to Bernstein's

Ok, so now we're going to throw out all the large elements and apply Bernstein's. Since we only have small elements, we know that

$$|R_j| \leq \frac{\alpha |y|_p}{\log n} = K.$$

Note that the randomness of the R_js that we are considering here is over the randomness of the CountSketch matrix, so filtering them based on D does not make them dependent.

This let's us plug into Bernstein's to get

$$\mathrm{P}\left[\left|\sum_{\text{small } j} \delta(h(j)=i)\sigma_j(Dy)_j\right| > \frac{|y|_p}{100}\right] \leq C\left(e^{-\Theta(\log n)+e^{-c\frac{\log n}{100\alpha}}}\right) \leq \frac{1}{n^2}$$

(as elaborated on page 9).

This is just for one bucket, but we have small enough failure probability here that we can union bound over all of the s buckets, and get that the "signed sum" of small j in every bucket will be at most $\frac{|y|_p}{100}$.

Wrapping Up

Ok, so if all the things that we conditioned on hold, then we have that

- $|(PDy)_i| \leq \frac{|y|_p}{100}$ if there are no large indices in the ith bucket.
- $|(PDy)_i| = |(Dy)_j| \pm \frac{|y|_p}{100}$ if exactly one large index j is in the ith bucket.
- No bucket has more than one large index.

We also conditioned on

$$|Dy|_\infty \in \left[\frac{|y|_p}{10^{\frac{1}{p}}}, 10^{\frac{1}{p}}|y|_p\right]$$

So, finally, this means that

$$\frac{|y|_p}{10^{\frac{1}{p}}} - \frac{|y|_p}{100} \leq |PDy|_\infty \leq 10^{\frac{1}{p}}|y|_p + \frac{|y|_p}{100}$$

so we can just output $|PDy|_\infty$ as our estimate of $|y|_p$.

This gives us a constant factor approximation with constant probability, which is what we wanted.

Space Complexity

Our sketching matrix has $s \in \mathcal{O}\left(n^{1-\frac{2}{p}}\log n\right)$ rows, so we need to keep track of that many words to keep our sketched vector at each time step. This is

$$\mathcal{O}\left(n^{1-\frac{2}{p}}\log^2 n\right)$$

space.

We didn't discuss it, but we assume that we can use limited independence to store our sketching matrices within this bound.