CS 15-851: Algorithms for Big Data Spring 2024

Lecture lecture 6 — 02/22/2024

Prof. David Woodruff *Scribe: Kunal Kapoor*

1 Proof of Sketching Theorem (Cont.)

Theorem 1. *W.h.p.,* $|Ax|_1 \leq |RAx|_1 \leq d \log d |Ax|_1$

Remember, $|RAx|_1 = |Ax|_1 * \Sigma_j |Z_j|/d \log d$ where $|Z_j|$ is a half-cauchy. We analyzed $\Sigma_j |Z_j|$ with Chernoff bounds to conclude $\Sigma_j |Z_j| = \Omega(d \log d)$ with probability $1 - e^{-d \log d}$ which implied w.h.p., $RAx|_1 \geq |Ax|_1$ as desired. Now, we want to show $|RAx|_1 \leq d \log d |Ax|_1$.

Note that the $|Z_j|$ are heavy-tailed; each $|Z_j|$ has a c.d.f. asymptotic to $1 - \theta(\frac{1}{z})$ which implies $Pr[Z_j \geq t] = \frac{1}{t}$ (ignoring constant factors). Intuitively, if we try to apply a similar analysis as we did in the $|Ax|_1 \leq |RAx|_1$ proof, we will not get a result strong enough to combine with a union bound over our $\gamma-$net.

We will approach showing the upper bound as follows:

Note that $|RA_{*i}|_1 = \ |A_{*i1}|_1 * \sum_j |Z_{i,j}|/d \log d$.

Let E_{ij} be the event that half-Cauchy is less than d^3. This has a probability around $1 - \frac{1}{z^3}$ by the earlier point about the cdf.

Define the truncated version of Z_{ij} as Z'_{ij}; this just cuts off the Cauchy at d^3. Now, note that $E[|Z_{ij}||E_{ij}] = E[|Z'_{ij}|] = O(\log d)$.

This can be shown as follows: the expectation is $\int_0^{d^3} \frac{2z}{\pi \cdot (1+z^2) \cdot \Pr[Z_{ij} \text{ in } [0,d^3]]} dz$, which is proportional to $\int_1^{d^3} \frac{1}{z} dz$, which is in $O(\log(d))$.

Now, let E be the event that for all i, j, E_{ij} occurs: $Pr[E] \geq 1 - \frac{\log d}{d}$. We get this by a union bound; $Pr(E$ doesn't hold) is the sum of probabilities of $Pr[E_{ij}]$ and there are $d^2 \log d$ i, j each with a probability of $\frac{1}{d^3}$.

Now, when solving for $E[Z'_{ij}|E_{ij}]$, we must use the law of total expectation to bound this quantity. This is because we don't know if the entries of the rows of RA are independent even though the

columns of RA are independent.

$$\begin{aligned}
E[Z'_{i,j}|E_{i,j}] &= E[Z'_{i,j}|E_{i,j}, E] \cdot Pr[E|E_{i,j}] + E[Z'_{i,j}|E_{i,j}, \neg E] \cdot Pr[E|\neg E_{i,j}] \\
&\geq E[Z'_{i,j}|E_{i,j}, E] \cdot Pr[E|E_{i,j}] \\
&= E[Z'_{i,j}|E] \cdot Pr[E|E_{i,j}] \\
&\text{Note that by Bayes' rule,} \\
Pr[E|E_{i,j}] &= \frac{Pr[E_{i,j}|E] \cdot Pr[E]}{Pr[E_{i,j}]} \\
&\geq 1 - \frac{\log d}{d}
\end{aligned}$$

Finally, we have $E[Z'_{ij}|E] = O(\log d)$ after moving the log-term to the other side. We get $E[Z'_{ij}] \geq E[Z'_{ij}|E] \cdot (1 - \frac{\log d}{d})$ (refer to slides for the steps which are easy). We can then conclude by moving the $\log d$ factor to the other side that $E[Z'_{ij}]|E = O(\log d)$.

Conditioned upon E happening, we can now compute the expectation of $|RA_{*i}|_1 = |A_{*i1}|_1 * \sum_j |Z_{i,j}|/d \log d$. We have the sum of the $|RA_{*i}|_1$ over $i = O(\log d) \cdot \sum_i |A_{*i}|_1$; we have this with constant probability by Markov's Inequality. Now, to improve from our prior bound of $poly(d)$, we need to take advantage of having a well-conditioned basis of A by using the Auerbach basis.

We know there exists a well-conditioned basis of A ($A_{*1}, A_{*2}, ..., A_{*d}$). A is well-conditioned; it is called an Auerbach basis which always exists and has the properties:

- For all x, $\|x\|_\infty \leq |Ax|_1$
- $\sum_i |A_{*i}|_1 = d$. The sum of column norms is d

For all x, we have $|RAx|_1 \leq \sum_i |RA_{*i}x_i|_1$ (triangle inequality) $\leq |x|_\infty \sum_i |RA_{*i}|_1$ (just pull the x_i out as it's a constant and it's bounded by the first property) $= |x|_\infty \cdot O(d \log d) \leq |Ax|_1 \cdot O(d \log d)$ by property 1 of the Auerbach basis. We have shown that for all x, our upper bound holds.

Finally, we are left showing that $|Ax|_1 \leq |RAx|_1$ for all x; as of now, we have that it holds with probability $1 - e^{-d \log d}$ for a fixed x. We will doing this by using properties of our $\gamma-$net; specifically, note that there exists some y close to Ax with distance at most $\gamma = \frac{1}{d^3 \log d}$. Then, we have the following chain of logic:

$|RAx|_1 \geq |Ry|_1 - |R(Ax - y)|_1$ by the triangle inequality.

$|Ry|_1 - |R(Ax - y)|_1 \geq |y_1| - O(d \log d)|Ax - y|_1$

This is greater than $|y_1| - O(d \log d)\gamma \geq |y_1| - O(\frac{1}{d^2}) \geq |y|_1/2$ where the last inequality follows from $|Ax|_1 \geq |x|_\infty$ from Auerbach $\geq \frac{1}{d}$ because $|x|_1 = 1$. Then, $|y|_1 \geq |Ax|_1 - \gamma - O(\frac{1}{d^2}) = |Ax|_1 - O(\frac{1}{d^2}) \geq |Ax|_1/2$ because $|Ax|_1 \geq \frac{1}{d}$.

2 Faster L_1 Regression

In the current algorithm for L1 Regression, the most expensive part is computing RA where R is the Cauchy matrix. It turns out this can be sped up by choosing R to be the product of a CountSketch and a matrix with Cauchy RV's on the diagonals. The bounds are different but this iss faster (overall time would become $\text{nnz}(A) + \text{poly}(\frac{d}{\epsilon})$.

3 Fun fact about Cauchy RVs

Suppose you have i.i.d copies of $R_1, R_2, ..., R_n$. Distribution of their mean is a normal RV by CLT $\sim (0, \frac{\sigma^2}{n})$. Now, take copies of a standard Cauchy RVs and take their mean in the same manner; it turns out the resulting RV is still a standard Cauchy. This illustrates a case where the central limit theorem fails as the central limit theorem requires a well-defined mean and variance.

4 Streaming Model

Sketching so far has been used to improve the time complexity of algorithms; now, we will examine ways to use it to improve memory usage.

4.1 Turnstile Streaming Model

- Start with an underlying n-dimensional vector x in $\mathbb{R}^n$ initialized to 0's.
- x undergoes a long stream of updates: $x_i \leftarrow x_i + \delta_i$ where δ is 1 or -1.
- At the end of the stream, x is still in a set from $(-M, \ldots, M)^n$ for some bound $M \leq \text{poly}(n)$.
- Output an approximation to $f(x)$ with high probability.
- Goal: use as little space (in bits) as possible. Can be used in industry when there's massive datasets, e.g., stock transactions, weather data, genomes, network traffic, etc.; so we can compute/keep track of statistics and such over the data.

4.2 Testing if $x = 0^n$

One question we may ask: Is the underlying vector x (after end of a stream) 0? We can figure this out by simply storing x as it undergoes the updates but that takes $O(n)$ space.

We can do better by sketching; if S is a CountSketch matrix with $O(\frac{1}{\epsilon^2})$ rows, then $|Sx|_2^2$ is within $(1 \pm \epsilon)|x|_2^2$ with probability at least $\frac{9}{10}$. Note that if $|x|_2 = 0$, Sx will also be 0 by our subspace embedding guarantee with high probability. Set $\epsilon = \frac{1}{2}$; we use $O(\log n)$ bits to store $O(1)$ entries of Sx because we only need to store the hash function and sign function.

One may wonder if it is possible to have a deterministic algorithm for this problem if we use less than $\Omega(n \log n)$ bits of space. It turns out this is impossible. We can use an argument using the pigeonhole principle to show this. Let a be the vector after the first half of the stream updates; there are $poly(n)^n$ possible vectors which means we need $\Omega(n \log n)$ bits to store all possibilites for a. If we have less than this many bits, we necessarily map two different values of a (say a_1, a_2) to the same representation. Then, consider the scenario in which the second half of the stream corresponds to a change of $-a_1$; a_1 should be mapped to 0 and a_2 should be mapped to $a_1 - a_2$ but since they were represented the same way, they must correspond to the same vector at the end of the data stream which means our algorithm will mess up on one of them, meaning it is not deterministic.

4.3 Recovering a k-spare vector

In this problem, we are promised that at the end of the stream, x has at-most k non-zero entries. The question is: can we recover the indicies and values of the k non-zero entries with high