CS 395T: Sublinear Algorithms Fall 2016

Lecture 6 — September 13, 2016

Prof. Eric Price *Scribe: Shanshan Wu, Yitao Chen*

1 Overview

Recap of last lecture. We talked about *Johnson-Lindenstrauss (JL) lemma* [JL84] and how to construct a distribution of linear maps that satisfies the JL guarantee. The main result is that let $A \in \mathbb{R}^{m \times n}$ with each entry drawn i.i.d. from a zero-mean subgaussian distribution with variance $\lesssim 1/m$, if $m = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$, then we have for $0 < \epsilon, \delta < 1$,

$$\forall x \in \mathbb{R}^n, \quad \mathbb{P}(|\|Ax\|_2^2 - \|x\|_2^2| \geq \epsilon \|x\|_2^2) \leq \delta. \tag{1}$$

The key idea is to show that $\|Ax\|_2^2$ is subgamma distributed:

$$\begin{aligned}
&A_{ij} \sim \text{subgaussian}(1/m) \\
&\implies A_{ij}x_j \sim \text{subgaussian}(x_i^2/m) \\
&\implies \sum_j A_{ij}x_j \sim \text{subgaussian}(\|x\|_2^2/m) \\
&\implies (\sum_j A_{ij}x_j)^2 \sim \text{subexponential}(\|x\|_2^2/m) \\
&\implies \|Ax\|_2^2 = \sum_i (\sum_j A_{ij}x_j)^2 \sim \text{subgamma}(\|x\|_2^4/m, \|x\|_2^2/m)
\end{aligned} \tag{2}$$

Since $\mathbb{E}[\|Ax\|_2^2] = \|x\|_2^2$, we can use subgamma's tail bound to prove concentration of $\|Ax\|_2^2$:

$$\mathbb{P}(|\|Ax\|_2^2 - \|x\|_2^2| \geq \epsilon \|x\|_2^2) \leq 2e^{-\frac{1}{2}\min(\epsilon^2 m, \epsilon m)} = 2e^{-\frac{1}{2}\epsilon^2 m} \leq \delta,$$

where the last inequality follows from the assumption that $m = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$. Surprisingly, this bound on m is actually tight (see Larsen and Nelson's recent paper [LN16] and the references therein).

Overview of this lecture. We will talk about two problems:

1. Storing the linear mapping A requires $O(mn)$ space, can we do better (sublinear in n)?
2. How to find the most frequent items ("heavy hitters") in a data stream?

2 AMS-sketch

In order to reduce the space usage of storing A, we relax the original full independence assumption to limited independence, and use hash functions to generate binary values for A_{ij}. Here is a formal definition for $k-$wise independence.

Definition 1. *$\mathcal{H} = \{h : [m] \to [l]\}$ is a k-wise independent hash family if $\forall i_1 \neq i_2 \neq \cdots \neq i_k \in [n]$ and $\forall j_1, j_2, \cdots, j_k \in [l]$,*

$$\underset{h \in H}{\mathbb{P}}[h(i_1) = j_1 \wedge \cdots \wedge h(i_k) = j_k] = \frac{1}{l^k}.$$

The key idea of using 4-wise independent hash function comes from the famous AMS sketch[1], which originates in a paper by Alon, Matias and Szegedy [AMS99]. It is initially designed to estimate the second frequency moments of streaming data. The algorithm works as follows.

1. Pick m random hash function h_1, h_2, ..., h_m from a 4-wise independent hash family $\mathcal{H} = \{h : [n] \to \{-\frac{1}{\sqrt{m}}, +\frac{1}{\sqrt{m}}\}$.
2. Let $A_{ij} = h_i(j)$, and compute $y_i = \sum_j A_{ij} x_j$, for all $i = 1, 2, .., m$.
3. Output $\sum_i y_i^2$, which is essentially $\|Ax\|_2^2$.

To see how this algorithm performs, we now compute the mean and variance of $\sum_i y_i^2$. And we will make use of the property of 4-wise independence.

$$\mathbb{E}[\sum_i y_i^2] = m\,\mathbb{E}[y_1^2] = m\,\mathbb{E}[(\sum_j A_{1j} x_j)^2] = m\,\mathbb{E}[\sum_j x_j^2/m + \sum_{j \neq k} A_{1j} A_{1k} x_j x_k] = \|x\|_2^2.$$

$$\mathrm{Var}(\sum_i y_i^2) = \mathrm{Var}(\|Ax\|_2^2) = \mathbb{E}[(\sum_i (\sum_j A_{ij} x_j)^2 - \|x\|_2^2)^2] = O(\|x\|_2^4/m), \tag{3}$$

where that last equality comes from the fact that $(\sum_i (\sum_j A_{ij} x_j)^2 - \|x\|_2^2)^2$ only involves upto 4 terms of A_{ij}, and with 4-wise independence, we can directly use the result achieved by full independence. The $O(\|x\|_2^4/m)$ is implied by the previous Eq. (2).

We have shown that the AMS algorithm outputs an unbiased estimator of $\|x\|_2^2$ with variance $O(\|x\|_2^4/m)$. Using Chebyshev's inequality gives the JL guarantee in Eq. (1) with probability at least 3/4:

$$\mathbb{P}(|\|Ax\|_2^2 - \|x\|_2^2| \geq \epsilon \|x\|_2^2) \leq \frac{\mathrm{Var}(\sum_i y_i^2)}{\epsilon^2 \|x\|_2^4} = O(\frac{1}{\epsilon^2 m}) \leq 1/4,$$

where the last inequality holds for $m = O(\frac{1}{\epsilon^2})$.

2.1 High-probability bound

There are two ways to get a high-probability bound with dependence $\log \frac{1}{\delta}$. The first method is to use the "median of mean" trick: we can perform the algorithm $O(\log \frac{1}{\delta})$ times and output the median. Using Chernoff's inequality[2], we can show that the median satisfies Eq. (1) with probability at least $1 - \delta$. The total space usage is $O(\frac{1}{\epsilon^2} \log \frac{1}{\delta} \log n)$.

Another way of achieving $O(\log \frac{1}{\delta})$ dependence is to use higher moments' bound. As shown in Figure 1, the bounds achieved by higher moments are better in the tail, but are worse otherwise (due to larger constants). In general, the k-th moment bound of a subguassian (reps. subexponential) variable has the form of $k^{k/2}/t^k$ (reps. k^k/t^k).

[1] It is sometimes called the "tug-of-war" estimator.

[2] To be more specific, we only need to bound the probability that more than half of the variables are bad.

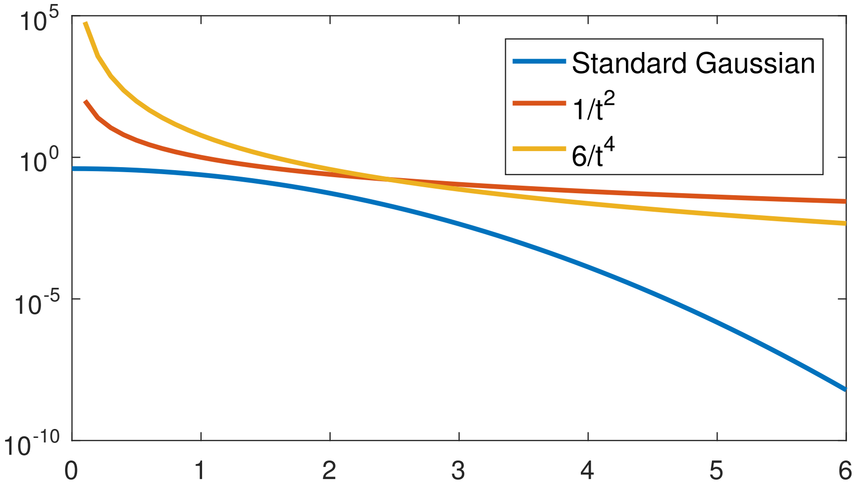


Figure 1: A comparison of different moments' bound: higher moments perform better in the tail, but worse close in (due to larger constants).

The idea of using higher moments is a simple extension of what we have done in Eq. (3): instead of using 4-wise independence to bound the second moment, more generally, if we have $2k$-wise independence, we can then compute the k-th moment of $|\|Ax\|_2^2 - \|x\|_2^2|$ by treating all the A_{ij}'s as independent. In other words, $|\|Ax\|_2^2 - \|x\|_2^2|^k$ should have the same behavior as that of an i.i.d. subgaussian matrix. More specifically, according to Eq. (2), $\|Ax\|_2^2 - \|x\|_2^2$ should behave like a subexponential[3] variable with parameter $\|x\|_2^2/\sqrt{m}$, i.e.,

$$\mathbb{E}[|\|Ax\|_2^2 - \|x\|_2^2|^k] \leq k^k(\|x\|_2^2/\sqrt{m})^k.$$

Using the Chebyshev bound, we get

$$\begin{aligned}\mathbb{P}(|\|Ax\|_2^2 - \|x\|_2^2| > \epsilon\|x\|_2^2) &= \mathbb{P}(|\|Ax\|_2^2 - \|x\|_2^2|^k > \epsilon^k\|x\|_2^{2k}) \\ &\leq \frac{\mathbb{E}[|\|Ax\|_2^2 - \|x\|_2^2|^k]}{\epsilon^k\|x\|_2^{2k}} \\ &\leq \frac{k^k}{(\epsilon\sqrt{m})^k}. \end{aligned} \tag{4}$$

The above bound holds for any k as long as we use $2k$-wise independent hash function, so we can optimize over k to get the best bound. Taking the derivate of $k\log(\frac{k}{\epsilon\sqrt{m}})$ and setting it to be zero, we get $k = \epsilon\sqrt{m}/e$ and

$$\mathbb{P}(|\|Ax\|_2^2 - \|x\|_2^2| > \epsilon\|x\|_2^2) \leq \min_k \frac{k^k}{(\epsilon\sqrt{m})^k} = e^{-\epsilon\sqrt{m}/e}.$$

Setting the failure probability to be at most δ gives us the desired $m = O(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$. Compared to the 4-wise independent hash family used in the original AMS sketch, we now need to use $O(\epsilon\sqrt{m}/e)$-wise hash family. The space complexity for this method is $O(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\log n)$.

[3] According the definition of subgamma distribution, a subgamma$(\|x\|_2^4/m, \|x\|_2^2/m)$ variable is also a subgamma$(\|x\|_2^4/m, \|x\|_2^2/\sqrt{m})$ variable, and hence is subexponetial.

3 Heavy hitters

The next problem that we are going to study is the *Heavy Hitter* problem. Given a (turnstile) stream of items, our goal is to find the "heavy hitters", i.e., the most frequent items. More formally, we are given a data stream $S = (s_1, s_2, \cdots)$, where $s_i \in [n]$. Let $x \in \mathbb{R}^n$ be the final histogram, i.e., x_i represents the number of times that item i appears. The goal is to find (approximately)

$$\{\ (i, x_i) \mid x_i \text{ is "large" } \}.$$

Furthermore, we want to get an estimate $\hat{x} \in \mathbb{R}^n$ of x, such that $\|\hat{x} - x\|_\infty \leq$ bound.

How small can the bound be? Here is a summary of results achieved by three algorithms.

Guarantee	Method	Bound	Space
l_∞/l_1	Heavy Hitters	$\epsilon\|x\|_1$	$O(\frac{\log n}{\epsilon})$
l_∞/l_1	Count-min Sketch [CM05]	$\frac{1}{K}\|x - x_K\|_1$	$O(K \log n)$
l_∞/l_2	Count Sketch [CCF02]	$\frac{1}{\sqrt{K}}\|x - x_K\|_2$	$O(K \log n)$

Remarks:

- Let $x_K \in \mathbb{R}^n$ denote the K largest entries of x. Then $\|x - x_K\|_1$ represents total numbers of of less frequent items.
- In homework, we will show that $\frac{1}{\sqrt{K}}\|x - x_{2K}\|_2 \leq \frac{1}{K}\|x - x_K\|_1$.

3.1 Preliminaries

First, we are interested in how $\{x_i\}$ usually behaves in practice.

Zipf's Law: The ith most common word in English appears approximately $1/i$ times.

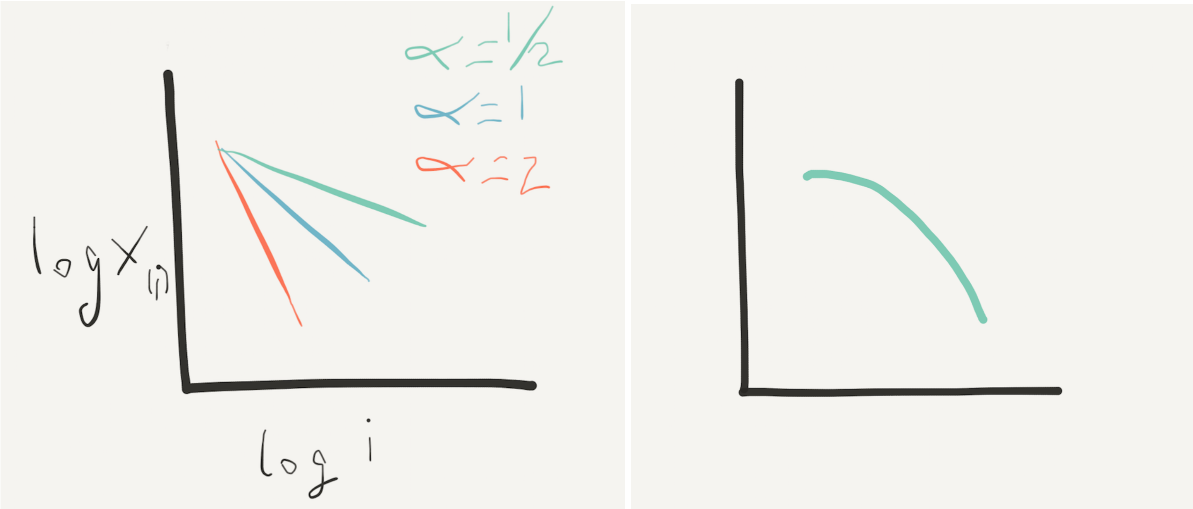
Power Law: The ith largest x_i appears approximately $1/i^\alpha$ for some $\alpha > 0$.

Power law has been widely observed in URLs in web, frequencies in music, population in cities, etc. Empirically, power law approximately holds for large entries.

Given power law, we can calculate different norms of x with parameter α,

$$\|x\|_1 = \sum_{i=1}^{n} \frac{1}{i^\alpha} = \begin{cases} 1 & \text{if } \alpha > 1 \\ \log n & \text{if } \alpha = 1 \\ n^{1-\alpha} & \text{if } \alpha < 1 \end{cases}$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} \frac{1}{i^{2\alpha}}} = \begin{cases} 1 & \text{if } \alpha > 1/2 \\ \sqrt{\log n} & \text{if } \alpha = 1/2 \\ n^{1/2-\alpha} & \text{if } \alpha < 1/2 \end{cases}$$



(a) Ideal power law

(b) Actual data, often

Figure 2: The $\log \log$ plot of power law: it empirically holds for large values.

And we can also calculate $\|x - x_K\|$,

$$\|x - x_K\|_1 = \sum_{i=K+1}^{n} \frac{1}{i^\alpha} = \begin{cases} K^{1-\alpha} & \text{if } \alpha > 1 \\ \log \frac{n}{K} & \text{if } \alpha = 1 \\ n^{1-\alpha} & \text{if } \alpha < 1 \end{cases}$$

$$\|x - x_K\|_2 = \sqrt{\sum_{i=K+1}^{n} \frac{1}{i^{2\alpha}}} = \begin{cases} K^{1/2-\alpha} & \text{if } \alpha > 1/2 \\ \sqrt{\log \frac{n}{K}} & \text{if } \alpha = 1/2 \\ n^{1/2-\alpha} & \text{if } \alpha < 1/2 \end{cases}$$

The following table summarizes how the bounds of different methods behave

Method	Bound equals	$\alpha > 1$	$1/2 < \alpha < 1$
Heavy Hitters	$\frac{1}{K}\|x\|_1 =$	$1/K$	$n^{1-\alpha}/K$
Count-min Sketch	$\frac{1}{K}\|x - x_K\|_1 =$	$1/K^\alpha$	$n^{1-\alpha}/K$
Count Sketch	$\frac{1}{\sqrt{K}}\|x - x_K\|_2 =$	$1/K^\alpha$	$1/K^\alpha$

We see count-sketch is much better than others when $1/2 < \alpha < 1$, which is the most common regime.

3.2 Heavy Hitters Algorithm

One potential way of (approximately) finding the heavy hitters is to sample a subset of the stream and find the most frequent item in the sampled stream. However, the sampling method has two issues: 1. It can not handle deletions; 2. It can not handle sparsity.

Another approach is to hash the stream to a smaller universe $B = O(K)$. According to pigeonhole principle, collision happens. However, in the following, we will argue that even count in the collided items, it is still a *good* estimate.

Here is the “Heavy Hitters” algorithm:

1. For $r = 1$ to $R = O(\log n)$, in parallel:
   (a) Pick a pairwise independent hash function $h_r : [n] \to [B]$.
   (b) Record $Y^{(r)}$, where $Y_j^{(r)} = \sum_{u:h_r(u)=j} x_u$, for $j \in [B]$.

2. Estimate $\hat{x}_u^{(r)} = Y_{h(u)}^{(r)}$, for all r and u.

3. Compute $\hat{x} = f(x^{(1)}, \ldots, x^{(R)})$, where f is some function (we will specify later).

Analysis: We first focus on any fixed hash function r and fixed item u. Without loss of generality, we let $r = 1$ and omit the superscription. Suppose it is the strict turnstile model: $x_u \geq 0, \forall u$, then the collisions only increase our estimate, $\hat{x}_u - x_u \geq 0, \forall u$.

Taking the expectation of the residual gives

$$\begin{aligned}
\mathbb{E}[\hat{x}_u - x_u] &= \mathbb{E}\left[\sum_{u':h(u')=h(u),u'\neq u} x_{u'}\right] \\
&= \mathbb{E}\left[\sum_{u'\neq u} x_u \cdot \mathbb{1}\{h(u') = h(u)\}\right] \\
&= \sum_{u'\neq u} x_{u'}\, \mathbb{P}[h(u') = h(u)] \\
&\leq \|x\|_1/B.
\end{aligned}$$

If $B = 4K$, by Markov inequality,

$$\mathbb{P}\left[|\hat{x}_u - x_u| > \frac{\|x\|_1}{K}\right] \leq \frac{\mathbb{E}[\hat{x}_u - x_u]}{\|x\|_1/K} \leq \frac{\|x\|_1/B}{\|x\|_1/K} = \frac{K}{B} = \frac{1}{4}.$$

So for any fixed u, we get $\hat{x}_u - x_u \leq \|x\|_1/K$ with probability at least $3/4$.

To bound the infinity norm $\|\hat{x}_u - x_u\|_\infty$, we require $\mathbb{P}[|\hat{x}_u - x_u| > \|x\|_1/K] < 1/4$ for all $u \in [n]$. The idea is to create R independent hash functions, and take the minimum of the outputs, i.e.,

$$\hat{x}_u = \min_r \hat{x}_u^{(r)}, \forall u \in [n].$$

Since we are considering the strict turnstile model: $\hat{x}_u^{(r)} \geq x_u, \forall r$, taking the minimum over all R estimates will get us closer to the true x_u. The minimum value is bad if and only if all R estimates are bad, so the failure probability is (recall $R = O(\log n)$)

$$\mathbb{P}\left[|\hat{x}_u - x_u| > \frac{\|x\|_1}{K}\right] \leq \frac{1}{4^R} \leq O(\frac{1}{n^2}).$$

Using the union bound, we have that $\|\hat{x} - x\|_\infty \leq \|x\|_1/K$ with probability at least $1 - 1/n$.

The total space used is $O(K \log n)$, because each hashing uses $O(K)$ space and we repeat $R = O(\log n)$ times.

3.3 Extentions

- What if x_u can be less than 0 (non-strict turnstile model) ?
  1. We still have $|\hat{x}_u^{(r)} - x_u| \leq \|x\|_1/K$ with probability 3/4.
  2. But $\min \hat{x}^{(r)}$ is bad. We can replace min with median, which gives worse constant under the same assumptions.

- How to obtain bounds using $\|x - x_K\|_1/K$?
  We split x into two parts: the largest K items x_K and the rest $x - x_K$. The key idea is to show that the probability of colliding with the top k most frequent items is small. More specifically, the probability that item u collides with any of largest K entires of x is

  $$\begin{aligned}&\mathbb{P}[h(u) = h(u') \text{ for some other } u' \text{ in top } K]\\ &\leq K \cdot \mathbb{P}[h(u) = h(u') \text{ for fixed } u' \neq u]\\ &= K/B.\end{aligned}$$

  where the last equality follows from the pairwise independence of h. Therefore, with probability at least $1 - K/B$, $h(u)$ will not collide with any of top K items. We can then bound $\mathbb{E}[\hat{x}_u - x_u]$ in terms of $\|x - x_K\|_1/K$ as before.

References

[JL84] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[LN16] Kasper Green Larsen and Jelani Nelson. Optimality of the Johnson-Lindenstrauss Lemma. *arXiv:1609.02094v1*, 2016.

[AMS99] Noga Alon, Yossi Matias, Mario Szegedy. The Space Complexity of Approximating the Frequency Moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

[CM05] Graham Cormode, S Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55.1 (2005): 58-75.

[CCF02] Moses Charikar, Kevin Chen, Martin Farach-Colton. Finding frequent items in data streams. *International Colloquium on Automata, Languages, and Programming*, Springer Berlin Heidelberg, 2002.