Sketching Algorithms for Big Data Fall 2017

Lecture 4 — September 12, 2017

Prof. Piotr Indyk *Scribe: Jialiang Wang*

1 Overview

In the last lecture we studied two methods for estimating L_2 norm of a stream. A stream of update $(i, 1)$ can be viewed as $x_i = x_i + 1$, i.e. increasing the respective bit in a count vector by 1. The algorithm maintains a linear sketch Rx, where R is a km random matrix. We then use low dimensional L_2 norm $||Rx||_2^2$ to estimate a high dimensional L_2 norm $||x||_2^2$

In this lecture we will first introduce a third algorithm to estimate the L_2 norm. The motivation behind it is that unlike the previous two algorithms we have seen in the previous lecture, this third algorithm can be easily generalized to L_p norm where $0 < p \leq 2$. In particular, we can use this algorithm to estimate the L_1 norm. Similar to the previous two algorithms, this algorithm also uses polylogarithmic space. Next, we will discuss another algorithm for estimating L_p norm where $p \geq 2$. This algorithm only works for positive updates (such as increment by 1), and it does not work for negative counting. The algorithm uses sampling instead of sketches and it uses $O(k\frac{m^{1-\frac{1}{k}}}{\epsilon^2})$ space for $(1 \pm \epsilon)$-approximation with constant probability

2 Another L_2 norm estimation

The first step of this algorithm is exactly the same as the previous two algorithms discussed in the previous lecture.

First, we use a linear sketch $R_x = [Z_1 \cdots Z_k]$, where each entry of R has distribution $N(0, 1)$, $k = O(\frac{1}{\epsilon^2})$. Therefore, each of Z_i has $N(0, 1)$ distribution with variance $\sum_i x_i^2 = ||x||_2^2$. Alternatively, we have $Z_i = ||x||_2 G_i$, where G_i is drawn from $N(0, 1)$.

Next, we need to choose an estimator to estimate $||x||_2^2$ from $Z_1, Z_2, \cdots Z_k$. In the previous two algorithms, we used $Y = [Z_1^2 + \cdots + Z_k^2]/k$ to estimate $||x||_2^2$, but there are many other estimators that we can potentially use. For example, we could instead use,

$$Y = \frac{\text{median}[|Z_1|, \cdots, |Z_k|]}{\text{median}[|G|]}$$

to estimate $||x||_2$ (G drawn from $N(0, 1)$). Here, median of an array A of numbers is the usual number in the middle of the sorted A, whereas median is the median of a random variable U if $Pr[U \leq \text{median}] = 0.5$. The intuition here is that $\text{median}[|Z_1|, \cdots, |Z_k|] = ||x||_2 \text{median}[|G_1|, \cdots |G_k|]$. For large enough k, $\text{median}[|G_1|, \cdots |G_k|]$ becomes close to $\text{median}[|G|]$. Next, we prove it formally.

Lemma 1: Let $U_1 \cdots U_k$ be i.i.d. real random variables chosen from any distribution having continuous c.d.f. F and median M, i.e., $F(t) = Pr[U_i < t]$ and $F(M) = 1/2$. Define $U =$

median$[U_1, \cdots, U_k]$. Then, for some absolute constant $C > 0$,

$$Pr[F(U) \in (\frac{1}{2} - \epsilon, \frac{1}{2} + \epsilon)] \geq 1 - e^{-C\epsilon^2 k}$$

Proof: Assume k is odd so that the median is well defined. Consider the events $E_i : F(U_i) < \frac{1}{2} - \epsilon$. We have $p = Pr[E_i] = \frac{1}{2} - \epsilon$. We can see that $F(U) < \frac{1}{2} - \epsilon$ if and only if at least $\frac{k}{2}$ of these events hold. By Chernoff bound, the probability that at least $\frac{k}{2}$ of the events hold is at most $e^{-C\epsilon^2 k}$. Therefore, $Pr[F(u) < \frac{1}{2} - \epsilon] \leq e^{-C\epsilon^2 k}$. The other side can be dealt with a similar manner. See Figure 1 for a visualization of the proof.

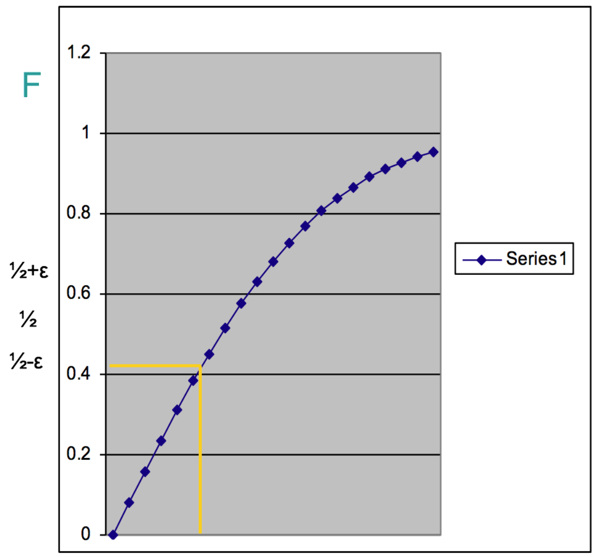

Figure 1: Visualization of Lemma 1. The blue dots are a series of samples. The median of the samples will be close to the true median (noted in the figure by the orange lines).

Lemma 2: Let F be the CDF of a random variable $|G|$, where G drawn from $N(0, 1)$. There exists a $C' > 0$ such that if for some z we have

$$F(z) \in (\frac{1}{2} - \epsilon, \frac{1}{2} + \epsilon),$$

then

$$z = \text{median}(g) \pm C'\epsilon$$

Proof: omitted. Use Calculus.

Putting everything together, we have:

Theorem: If we use median estimator

$$Y = \frac{\text{median}[|Z_1|, \cdots, |Z_k|]}{\text{median}[|g|]}$$

where $Z_j = \sum_i r_{ij} x_i$, r_{ij} is chosen i.i.d. from $N(0, 1)$, then we have

$$Y = ||x||_2 \frac{[\text{median}(g) \pm C'\epsilon]}{\text{median}[|g|]} = ||x||_2 (1 \pm C''\epsilon)$$

with probability

$$1 - e^{-C\epsilon^2 k}.$$

Note 1: the reason why we use Gaussian here is not because we need Gaussian for the proof of the two Lemmas, but rather the property that we can divide Z_i by G_i to estimate $||x||_2$.

Note 2: In general, we cannot use median to replace median of means. Consider the samples $[-1, 1, -1, 1, -1, 1 \cdots]$. If we only use median, we will never get the true expectation 0. We cannot use mean to replace median of means either. Consider the example of a heavy-tailed distribution. Simply speaking, we use mean to reduce the variance, and we use median to amplify the probability.

3 Extending the median estimator to L_p

The key property of normal distribution that we used is: if $U_1, \cdots U_k$ are independent, and U is normal distribution, then $x_1U_1 + \cdots + x_mU_m$ is distributed as

$$(|x_1|^p + \cdots + |x_m|^p)^{\frac{1}{p}} U$$

Such distributions are called "p-stable" and exist for and $p \in (0, 2]$.

For $p = 1$, we have Cauchy distribution. The density function is $f(x) = \frac{1}{\pi(1+x^2)}$, CDF is $F(z) = \frac{arctan(z)}{\pi} + \frac{1}{2}$. 1-stability: $x_1U_1 + \cdots + x_mU_m$ is distributed as $(|x_1| + \cdots + |x_m|)U$. The PDF and CDF of Cauchy distirbution can be visualized in Figures 2 and 3.

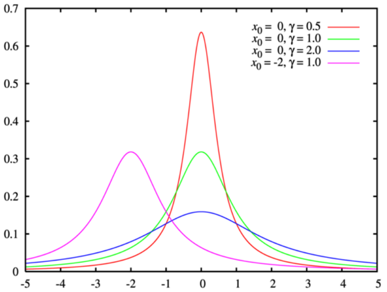

Figure 2: PDF of Cauchy distribution

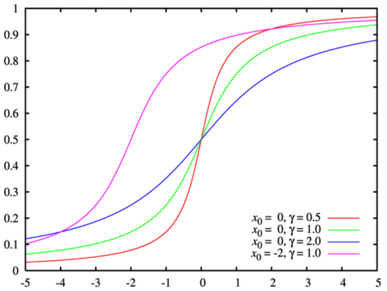

Figure 3: CDF of Cauchy distribution

Note that Cauchy distribution does not have the first or the second moment. However, the estimator arguments we used in proving the algorithm can still go through. Therefore, we can generate random Cauchy by choose a random $u \in [0, 1]$ and computing $F^{-1}(u)$ to estimate L_1 norm.

Similarly, for $L_{\frac{1}{2}}$ norm, we can use Levy distribution. See Wikipedia.

When $p \neq 1, 2, \frac{1}{2}$, there is no closed form formula for density or CDF, and we are not clear where the median is. Also, we are not clear what the derivative of CDF around the median is. Nevertheless, we can use some hack to estimate. See [1] and [2]. For more information on p-stable distributions, see this book.

So far, we have ignored randomness and discretization issues (but everything can be done using O(log (m+n)) bit numbers). See [3] on how to fix this and get the optimal bounds.

4 L_p norm for $p \geq 2$

Unfortunately, p-stable distributions do not exist with $k > 2$. We will introduce a new algorithm for estimating L_k norm in this section. The algorithm only works for **a stream**. For a stream of elements $i_1, \cdots i_n$, each i can be interpreted as $x_i = x_i + 1$. This algorithm only work for these updates, not for decrements. The space this algorithm needs is $O(\frac{m^{1-\frac{1}{k}}}{\epsilon^2})$ for $(1 \pm \epsilon)$-approximation with constant probability.

4.1 L_k norm estimation [4]

We will use notation $F_k = \sum_{i=1}^m x_i^k = ||x||_k^k$ for frequency moment of the stream $i_1 \cdots i_n$. The first algorithm we introduce has two passes:

Pass 1: Pick a stream element $i = i_j$ uniformly at random
Pass 2: Compute x_i
Return $Y = nx_i^{k-1}$

Analysis: our estimator is $Y = nx_i^{k-1}$. The expectation of our estimator is $E[Y] = \sum_i \frac{x_i}{n} n x_i^{k-1} = \sum_i x_i^k = F_k$. The second moment ($\geq$ variance) is $E[Y^2] = \sum_i \frac{x_i}{n} n^2 x_i^{2k-2} = n \sum_i x_i^{2k-1} = nF_{2k-1}$. We claim that $nF_{2k-1} \leq m^{1-\frac{1}{k}}(F_k)^2$. Therefore, by averaging over $O(\frac{m^{1-\frac{1}{k}}}{\epsilon^2})$ samples and using Chebyshev bound will do the job.

Next we prove the above-mentioned claim: $nF_{2k-1} \leq m^{1-\frac{1}{k}}(F_k)^2$
Proof:

$$\begin{aligned} & nF_{2k-1} \\ =& n||x||_{2k-1}^{2k-1} \\ \leq& n||x||_k^{2k-1} \\ =& ||x||_1 ||x||_k^{2k-1} \\ \leq& m^{1-\frac{1}{k}} ||x||_k ||x||_k^{2k-1} \\ =& m^{1-\frac{1}{k}} ||x||_k^{2k} \\ =& m^{1-\frac{1}{k}} F_k^2 \end{aligned}$$

Now, we want to reduce our algorithm to only having one pass. However, such one-pass algorithm cannot compute x_i exactly. Instead we first pick $i = i_j$ uniformly at random from the stream.

Then compute $r = \#$ occurrences of i in the suffix $i_j \cdots i_n$. We use r instead of x_i in the estimator. Clearly $r \leq x_i$, but we can see that $E[r] = \frac{x_i+1}{2}$, so intuitively things should work out up to a constant factor.

There is an even better idea, which is to use $Y' = n(r^k - (r-1)^k)$ as the estimator. The expectation is

$$\begin{aligned} E[Y'] &= nE[(r^k - (r-1)^k)] \\ &= n\frac{1}{n}\sum_i \sum_{j=1}^{x_i} \left[j^k - (j-1)^k\right] \\ &= \sum_i x_i^k \end{aligned}$$

To see the second moment, we first observe that $Y' = n(r^k - (r-1)^k) \leq nkr^{k-1} \leq kY$. Therefore, $E[Y'^2] \leq k^2 E[Y^2] \leq k^2 m^{1-\frac{1}{k}} F_k^2$ (We can improve this to $km^{1-\frac{1}{k}} F_k^2$ for integer k). Using these two, we have proved the one pass algorithm for F_k (positive updates). The space requiremnet is $O(\frac{k^2 m^{1-\frac{1}{k}}}{\epsilon^2})$ for $(1 \pm \epsilon)$-approximation.

Note 1: the algorithm in [4] only works for integer k. However, it is easy to adapt to any $k > 1$.

Note 2: $m^{1-\frac{1}{k}}$ is not optimal, we can achieve $m^{1-\frac{2}{k}} log^{O(1)}(mn)$. See [5] and [6].

Note 3: The sampling is quite general. The empirical entropy, i.e. $\sum_i \frac{x_i}{n} log \frac{x_i}{n}$ in polylog n space [7].

References

[1] Indyk, Piotr. Stable distributions, pseudorandom generators, embeddings and data stream computation. *Journal of the ACM (JACM)* :53.3 (2006): 307-323.

[2] Li, Ping. Estimators and tail bounds for dimension reduction in $\ell_\alpha (0 < \alpha \leq 2)$ using stable random projections. *J. Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics*, 2008.

[3] Kane, Daniel M., Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.* ACM, 2010.

[4] Noga Alon, Yossi Matias, Mario Szegedy. The Space Complexity of Approximating the Frequency Moments. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 20–29, 1996.

[5] Indyk, Piotr, and David Woodruff. Optimal approximations of the frequency moments of data streams. *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing. ACM,* 2005.

[6] Braverman, Vladimir, et al. An Optimal Algorithm for Large Frequency Moments Using $O(n^{(1-2/k)})$ Bits. *LIPIcs-Leibniz International Proceedings in Informatics.* Vol. 28. 2014.

[7] Chakrabarti, Amit, Graham Cormode, and Andrew McGregor. A near-optimal algorithm for estimating the entropy of a stream. *ACM Transactions on Algorithms (TALG)* 6.3 (2010): 51.