COS 521 Lecture 4: Streaming Algorithms

Guest lecture by Jelani Nelson

Transcribed by Michael Zhu

February 13, 2013

1 Chernoff bound

The Chernoff bound is a tail bound for sums of independent real variables.

Theorem: Let X_1, ..., X_n be independent random variables each in $[0, 1]$. Let $X = \sum_i X_i$. Let $\sigma^2 = Var[X]$. Then $Pr[|X - E[X]| > \lambda\sigma] \leq c_1 \max\{e^{-c_2\lambda^2}, e^{-c_3\lambda\sigma}\}$.

Proof: We will prove an upper bound on $Pr[X - E[X] > \lambda\sigma]$ (the proof of the other inequality is similar). Define $Y_i = X_i - E[X_i]$, and note that the Y_i's are independent. We have $Y = \sum_i Y_i = \sum_i \Big(X_i - E[X_i]\Big) = X - E[X]$. Let $t \in (0, 1)$ be some constant to be determined later.

$$\begin{aligned}
Pr[X - E[X] > \lambda\sigma] &= Pr[e^{t(X-E[X])} > e^{t\lambda\sigma}] \\
&= Pr[e^{tY} > e^{t\lambda\sigma}] \\
&< e^{-t\lambda\sigma} E[e^{tY}] \\
&= e^{-t\lambda\sigma} \prod_i E[e^{tY_i}] \\
&\leq e^{-t\lambda\sigma} \prod_i e^{et^2 Var[X_i]} \\
&= e^{-t\lambda\sigma} e^{et^2 \sum_i Var[X_i]} \\
&\leq e^{-t\lambda\sigma + et^2\sigma^2}
\end{aligned}$$

The first inequality follows by Markov's inequality. To derive the second inequality, we use Taylor's theorem. For twice-differentiable functions f, we have that $f(x) = f(0) + f'(0)x + \frac{f"(\alpha)x^2}{2}$ for some $\alpha \in (0, x)$. Applied to $f(Y_i) = e^{tY_i}$, we get

$$e^{tY_i} \leq 1 + tY_i + \frac{e^t t^2 Y_i^2}{2}$$

Taking the expectation of both sides,

$$\begin{aligned} E[e^{tY_i}] &\leq 1 + \frac{e^t t^2 E[Y_i^2]}{2} \\ &\leq 1 + et^2 Var[X_i] \\ &\leq e^{et^2 Var[X_i]} \end{aligned}$$

Now setting $t = \min(\frac{1}{16}, \frac{\lambda}{2e\sigma})$, the desired result follows.

2 Streaming Algorithm for counting distinct elements

Streaming algorithms are efficient algorithms designed to process large data streams "on the fly" by making one or more passes over the input data but with limited amounts of memory available for storage. The distinct element problem asks us to count the number of distinct elements i_1, i_2, ..., i_m $\in \{1, ..., n\}$. The algorithm is required to output $F_0 = \#$ distinct $i's$. The problem was first studied in Flajolet, Martin 1983. Using $O(\log n)$ bits of memory, the algorithm outputs $\tilde{F_0} \in [\frac{F_0}{6}, 6F_0]$ with probability at least 2/3. The error probability can be reduced to $1 - \delta$ using Chernoff bound.

Algorithm (idealized):

1. Pick random hash function $h : \{1, ..., n\} = [n] \to [0, 1]$
2. Maintain $X = \min_{i \in stream} h(i)$ in memory
3. At the end of the stream, output $1/X$

Intuition:

If we have t independent random variables in the interval $[0, 1]$, we expect these random variables to be evenly spaced in the interval $[0, 1]$, that is, taking on values $\frac{1}{t+1}, \frac{2}{t+1}, ..., \frac{t}{t+1}$. Suppose there are t distinct elements x_1, ..., x_t in the input stream. Since the hash values of the distinct elements in the stream can be treated as independent random variables, we expect the distinct hash values $h(x_1)$, ..., $h(x_t)$ to be evenly spaced in $[0, 1]$. We expect the smallest hash value to be approximately $\frac{1}{t+1}$ and the inverse of this to yield a good approximation to the number of distinct queries.

Analysis:

Event E_1: There are no hash values less than $\frac{1}{6F_0}$

Event E_2: There is at least one hash value less than $\frac{6}{F_0}$

Note that E_1 implies $X \geq \frac{1}{6F_0}$ and $\frac{1}{X} \leq 6F_0$ while E_2 implies $X \leq \frac{6}{F_0}$ and $\frac{1}{X} \geq \frac{F_0}{6}$. If events E_1 and E_2 both occur, then the estimate $\frac{1}{X} \in [\frac{F_0}{6}, 6F_0]$ output by the algorithm is in the correct interval.

To bound the probability that the algorithm makes an error, we first show that the probability E_1 doesn't occur is less than 1/6. Let a_1, a_2, ..., a_{F_0} be the distinct values in the stream. Let Z_i be an indicator random variable for the event $h(a_i) < \frac{1}{6F_0}$. Let $Z = \sum_{i=1}^{F_0} Z_i$ be the number of distinct i's such that $h(a_i) < \frac{1}{6F_0}$. By a simple calculation, $E[Z] = \sum_{i=1}^{F_0} E[Z_i] = \sum_{i=1}^{F_0} \frac{1}{6F_0} = \frac{1}{6}$. Using Markov's inequality, $Pr(\text{not } E_1) = Pr(Z \geq 1) \leq \frac{1}{6}$.

A similar calculation shows that the probability E_2 doesn't occur is also less than 1/6. Let Z_i' be an indicator random variable for the event $h(a_i) < \frac{6}{F_0}$. Let $Z' = \sum_{i=1}^{F_0} Z_i'$, and we have $E[Z'] = \sum_{i=1}^{F_0} E[Z_i'] = 6$. Since the Z_i's are independent random variables, we have $Var[Z'] = \sum_{i=1}^{F_0} Var[Z_i'] = \sum_{i=1}^{F_0} \left(E[Z_i'^2] - E[Z_i']^2 \right) < \sum_{i=1}^{F_0} E[Z_i'] = 6$. Using Chebyshev's inequality, we conclude that $Pr[\text{not } E_2] \leq Pr[|Z' - E[Z']| \geq 6) \leq \frac{Var[Z']}{36} \leq \frac{1}{6}$.

In reality, we can't use truly random hash functions since real numbers and real intervals cannot be represented on the computer. Since the above proof only needs the sum of variance property to hold, we only need h to be a pairwise independent hash function. Constructing h and storing h takes $O(\log n)$ bits of memory.

3 Boosting success probability

To boost the success probability of the streaming algorithm presented in the previous section,

1. Repeat $t = O(\log \frac{1}{\delta})$ times with independent randomness over the trials. Get t estimates $\tilde{F_0^1}$, $\tilde{F_0^2}$, ..., $\tilde{F_0^t}$.
2. Output the median of the t estimates over $\tilde{F_0^i}$

The claim is that the median of the t estimates is in $[\frac{F_0}{6}, 6F_0]$ with probability at least $1 - \delta$. As long as a strict majority of trials give us a good answer (in the correct interval), then the median will be a good answer.

Analysis:

Let X_i be an indicator random variable for the event that trial # i gave a good answer (in the correct interval). Let $X = \sum_{i=1}^{t} X_i$. We want to show that $Pr[X > \frac{t}{2}] \geq 1 - \delta$. By linearity of expectation, $E[X] = \sum_{i=1}^{t} Pr[X_i = 1] \geq 2t/3$. We can bound the error probability as $Pr[X \leq t/2] \leq Pr[|X - E[X]| > (\frac{2}{3} - \frac{1}{2})t]$ and use the Chernoff bound with an appropriate choice of t to show that the error probability is less than δ.

4 Lower bounds in streaming

Both randomization and approximation are necessary for getting a good solution to the distinct elements problem in sublinear space. We consider the case of no randomness and no approximation. Suppose there exists a deterministic, exact streaming algorithm which computes the number of distinct elements in sublinear space. We can design a universal compression algorithm as follows, which is a contradiction.

Encoding algorithm for any $x \in \{0,1\}^n$:

1. Create stream D containing $\{i \mid x_i = 1\}$.
2. Run streaming algorithm on D
3. Output memory contents of streaming algorithm

Decoding algorithm:

1. Calculate F_0, the number of distinct elements.
2. For each $i = 1, 2, ...n$:
   (a) Add i to the stream.
   (b) Ask if F_0 increased ($x_i = 1$ iff F_0 didn't increase)
3. Output $\{i \mid i^{th}$ iteration of for loop, F_0 didn't increase$\}$

5 F_2 estimation

Given a stream of elements in the range $\{1, ..., n\}$, the stream defines a frequency vector $x \in \mathbb{R}^n$ where x_i is the number of times that the element i appears. When v copies of element i are seen, we have the update $x_i \leftarrow x_i + v$. The goal is to output $F_2 = \|x\|_2^2 = \sum_{i=1}^{n} x_i^2$. A $O(\log n)$-space algorithm was introduced by Alan, Matias, Szegedy (1996).

Algorithm:

Construct a m by n matrix $\Pi = [\sigma_{ij}]$ where the entries σ_{ij} are drawn at random independently from $\{-1, +1\}$. Initially, $x \in \mathbb{R}^n$ is the 0 vector. The algorithm maintains $\Pi x \in \mathbb{R}^m$ in memory, where Πx is initially the 0 vector. The update $x_i \leftarrow x_i + v$ can be formulated in vector notation as $x \leftarrow x + ve_i$. We can compute $\Pi(x + ve_i) = \Pi x + v\Pi_i$ by adding v times Π_i (the i^{th} column of Π) to Πx. At the end of the stream, the algorithm outputs $\frac{1}{m}\|\Pi x\|_2^2$.

Analysis:

$$\begin{aligned}
\frac{1}{m}\|\Pi x\|_2^2 &= \frac{1}{m}\sum_{i=1}^{m}(\Pi x)_i^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\Pi_{ij}x_j\right)^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\sum_{j'=1}^{n}\sigma_{ij}\sigma_{ij'}x_j x_{j'}\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\sigma_{ij}^2 x_j^2 + \sum_{j\neq j'}\sigma_{ij}\sigma_{ij'}x_j x_{j'}\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n} x_j^2 + \sum_{j\neq j'}\sigma_{ij}\sigma_{ij'}x_j x_{j'}\right)
\end{aligned}$$

Assuming pairwise independence, the expectation is

$$\begin{aligned}
E\left[\frac{1}{m}\|\Pi x\|_2^2\right] &= \frac{1}{m}\sum_{i=1}^{m}\left(\|x\|_2^2 + \sum_{j\neq j'}E[\sigma_{ij}\sigma_{ij'}]x_j x_{j'}\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(\|x\|_2^2 + \sum_{j\neq j'}E[\sigma_{ij}]E[\sigma_{ij'}]x_j x_{j'}\right) \\
&= \|x\|_2^2 = F_2
\end{aligned}$$

Assuming our hash function is 4-wise independent, we can bound the variance by

$$\begin{aligned}
Var\left[\frac{1}{m}\|\Pi x\|_2^2\right] &= E\left[\left(\frac{1}{m}\|\Pi x\|_2^2 - E[\frac{1}{m}\|\Pi x\|_2^2]\right)^2\right] \\
&= E\left[\left(\frac{1}{m}\sum_{i=1}^{m}\sum_{j\neq j'}\sigma_{ij}\sigma_{ij'}x_j x_{j'}\right)^2\right] \\
&= \frac{2}{m^2}\sum_{i=1}^{m}\sum_{j\neq j'}E[\sigma_{ij}^2\sigma_{ij'}^2]x_j^2 x_{j'}^2 \\
&= \frac{2}{m^2}\sum_{i=1}^{m}\sum_{j\neq j'}x_j^2 x_{j'}^2 \\
&\leq \frac{2}{m}(\sum_j x_j^2)^2 \\
&= \frac{2}{m}\|x\|_2^4
\end{aligned}$$

By Chebyshev's inequality, we have

$$\begin{aligned}
Pr\left[\left|\frac{1}{m}\|\Pi x\|_2^2 - \|x\|_2^2\right| > \epsilon\|x\|_2^2\right] &\leq \frac{\frac{2}{m}\|x\|_2^4}{\epsilon^2\|x\|_2^4} \\
&= \frac{2}{m\epsilon^2}
\end{aligned}$$

For any $\epsilon > 0$ and $\delta > 0$, we can choose m sufficiently large so $\frac{1}{m}\|\Pi x\|_2^2 \in [(1-\epsilon)F_2, (1+\epsilon)F_2]$ with probability at least $1-\delta$