Last class we saw a $(3,\delta)$-approx. for the # of distinct elements in a data stream.

Setup: Data stream $S = \{s_1, \dots, s_m\}$

where m is HUGE

& for each i, $s_i \in \{1, 2, \dots, n\}$

Let $f = (f_1, \dots, f_n)$ where

$$f_i = |\{j : 1 \leq j \leq m, s_j = i\}|$$

$= \#$ of occurrences of i in S

Our goal is to compute $D = |\{i : f_i > 0\}|$

$= \#$ of distinct elements in S

Note, $D = F_0$ where F_k was defined a couple of lectures ago.

Last class: We saw an approx. alg. that outputs $\hat{D}$ where:

$$\Pr\left(\frac{D}{3} \leq \hat{D} \leq 3D\right) \geq .04$$

& by taking the median of $O(\log(1/\delta))$ trials we can boost this success prob. to $\geq 1-\delta$.

this required space $O(\log(1/\delta)\log n)$.

<u>Today</u>: We'll boost the approximation factor:

for any $\epsilon > 0$, we'll guarantee that

$$\Pr\left(D(1-\epsilon) \le \hat{D} \le D(1+\epsilon) \right) \ge 1-\delta.$$

& we'll do this with space $O\left(\left(\frac{1}{\epsilon^2}\right)\log\left(\frac{1}{\delta}\right)\log n\right)$.

<u>Idea</u>: Same basic approach as last time, which is to find max of $\text{zeros}(h(k))$ for $k \in S$, <u>using</u> a pairwise independent hash function.

Recall, for $k \in S$, $\Pr\left(\text{zeros}(h(k)) \ge r\right) = 2^{-r}$,

hence, we expect $\frac{D}{2^z}$ to have $\text{zeros}(h(k)) \ge z$.

Our approach will be to keep track of how many items have $\text{zeros}(h(k)) \ge z$

where z is close to ~~$\max_k \text{zeros}$~~ $\max_k \text{zeros}(h(k))$.

Then, we'll output $|B| 2^z$ as our estimate of D

where B is the bucket containing those k with $\text{zeros}(h(k)) \ge z$.

We'll keep $|B| \le O(1/\epsilon^2)$ & if it exceeds that ~~that~~ ~~to~~ then we'll increase z.

Alg. [BJKST '04] = Bar-Yossef, Jayram, Kumar, Sivakumar & Trevisan.

1. Choose a random pairwise independent hash function $h:[n]\to[n]$.
   (We need n to be prime so if not choose $n \le p \le 2n$ which is prime)

   Do this by choosing a, b indpt. & uniformly from $\{0, 1, \ldots, p-1\}$ & setting $h(i) = a + bi \mod p$.

2. Set $z = 0$ & $B = \emptyset$.

3. Process data stream S in a one-by-one manner.
   For element $k \in S$:
   - if $\text{zeros}(h(k)) \ge z$ then:
     - $B = B \cup (h(k), \text{zeros}(h(k)))$
     - while $|B| \ge c/\epsilon^2$: (specify c below)
       - $z = z + 1$
       - Remove all (α, β) from B where $\beta < z$.

4. Output $|B| 2^z$

Recall from last class,

for $k \in \{1, \dots, n\}$, and integer $\ell \geq 0$,

let $X_{\ell,k} = \begin{cases} 1 & \text{if zeros}(h(k)) \geq \ell \\ 0 & \text{o/w} \end{cases}$

& let $Y_\ell = \sum_{k: f_k > 0} X_{\ell,k} = |\{k \in S : \text{zeros}(h(k)) \geq \ell\}|$

Consider the final value of z.

The alg. outputs $\hat{d} = Y_z 2^z$

Let's bound the prob. that $\hat{d}$ is a poor approx. of d:

we want to show that $\hat{d} > (1+\epsilon)d$ or $\hat{d} < (1-\epsilon)d$ are unlikely.

$\hat{d} > (1+\epsilon)d \Leftrightarrow Y_z 2^z > (1+\epsilon)d \Leftrightarrow Y_z 2^z - d > \epsilon d$

& $\hat{d} < (1-\epsilon)d \Leftrightarrow Y_z 2^z < (1-\epsilon)d \Leftrightarrow Y_z 2^z - d < -\epsilon d$

So, the alg. FAILS if $|Y_z 2^z - d| \geq \epsilon d$

or $\left| Y_z - \frac{d}{2^z} \right| \geq \frac{\epsilon d}{2^z}$

Note, $E[Y_z] = \frac{d}{2^z}$

Note, $Pr(X_{l,k} = 1) = 2^{-l}$ & $E[X_{l,k}] = 2^{-l}$

Hence, $E[Y_l] = \frac{d}{2^l}$

and $Var(Y_l) = \sum_{k : f_k > 0} Var(X_{l,k}) \leq \sum_k E[X_{l,k}^2] = \sum_k E[X_{l,k}]$

since $X_{l,k}$ is 0-1 r.v.

$$\leq \frac{d}{2^l}$$

$$Pr(FAIL) = Pr\left(\left|Y_z - \frac{d}{2^z}\right| > \frac{\epsilon d}{2^z}\right)$$

$$= \sum_{r=1}^{\log n} Pr\left(\left|Y_r - \frac{d}{2^r}\right| > \frac{\epsilon d}{2^r}, \ \& \ z = r\right)$$

we'll choose s later

$$\leq \sum_{r=1}^{s-1} Pr\left(\left|Y_r - \frac{d}{2^r}\right| > \frac{\epsilon d}{2^r}\right) + \sum_{r=s}^{\log n} Pr(z = r)$$

$$= \sum_{r=1}^{s-1} Pr\left(\left|Y_r - \frac{d}{2^r}\right| > \frac{\epsilon d}{2^r}\right) + Pr(z \geq s)$$

$$= \sum_{r=1}^{s-1} Pr\left(\left|Y_r - \frac{d}{2^r}\right| > \frac{\epsilon d}{2^r}\right) + Pr\left(Y_{z-1} \geq \frac{c}{\epsilon^2}\right)$$

bound using Chebyshev's ineq.

bound using Markov's ineq.

because for z to $\uparrow$ we need in the alg. that Y_{z-1} is too big.

$$\Pr\left(\left|Y_r - \frac{d}{2^r}\right| \geq \frac{\epsilon d}{2^r}\right) \leq \Pr\left(\left|Y_r - E[Y_r]\right| \geq \frac{\epsilon d}{2^r}\right)$$

$$\leq \frac{Var(Y_r)}{\left(\frac{\epsilon d}{2^r}\right)^2} \leq \frac{\left(d/2^r\right)}{\left(\frac{\epsilon d}{2^r}\right)^2} = \frac{2^r}{\epsilon^2 d}.$$

Note, $\sum_{r=1}^{s-1} 2^r \leq 2^s$

& hence $\sum_{r=1}^{s-1} \frac{2^r}{\epsilon^2 d} \leq \frac{2^s}{\epsilon^2 d}$

Now, choose s to be the largest integer where:

$$\frac{d}{2^s} < \frac{24}{\epsilon^2}$$

note that by taking s one smaller $\frac{d}{2^s}$ goes down by $\frac{1}{2}$

So we know $\frac{d}{2^s} \geq \frac{12}{\epsilon^2}$

& note, $2^s \leq \frac{d\epsilon^2}{12}$, i.e., $\frac{2^s}{d} \leq \epsilon^2/12$.

So: $s = O(\log(c' d \epsilon^2))$ for some constant c'.

$$\Pr\left(Y_{s-1} \geq \frac{c}{\epsilon^2}\right) \leq \frac{E[Y_{s-1}]}{c/\epsilon^2} \leq \frac{\epsilon^2}{c}\frac{d}{2^{s-1}} \leq \frac{2\epsilon^2}{c}\frac{d}{2^s}$$

$$\leq \frac{2\epsilon^2}{c}\frac{24}{\epsilon^2}$$

$$\text{since } \frac{d}{2^s} < \frac{24}{\epsilon^2}$$

Therefore, combining these 2 bounds we have:

$$\Pr(\text{FAIL}) \leq \frac{2^s}{\epsilon^2 d} + \frac{2\epsilon^2}{c}\frac{24}{\epsilon^2}$$

$$\leq \frac{1}{\cancel{\epsilon^2}}\frac{\cancel{\epsilon^2}}{12} + \frac{48}{c} = \frac{1}{12} + \frac{48}{c} \leq \frac{1}{6}$$

for $c \geq 12 \cdot 48$.

Now use the median of $O(\log(1/\delta))$ indp't. trials to boost the success prob. to $\geq 1-\delta$.

Space: $\underbrace{O(\log n)}_{\text{per item}} \times \underbrace{\frac{c}{\epsilon^2}}_{\substack{\text{Max \#}\\ \text{in } B}} + \underbrace{O(\log n)}_{\text{to choose/store } a,b} + \underbrace{O(\log\log n)}_{\text{for } z \le \log n}$

$= O\left(\frac{1}{\epsilon^2}\log n\right)$

Can reduce it to $O\left(\log n + \frac{1}{\epsilon^2}\left(\log(1/\epsilon) + \log\log n\right)\right)$ by using a hash function to keep track of the elements of B.