Distinct Elements in Vanilla Streaming Model

Goal: Let $F_0 = |\{j : f_j > 0\}|$. Given (ϵ,δ)-approximation algorithm for F_0.

Let $X_{(x_1,v_1)}, \dots, X_{(x_k,v_k)}$ be random variables indicating whether $h(x_i)=v_i$ for $h \in H$. H is pairwise-independent hash family iff these indicator variables are pairwise-independent

Preliminaries: Pairwise independent family of hash functions.

A family H of hash functions mapping $[n]\to[r]$ is said to be pairwise independent if for any distinct $x,y \in [n]$ and $v,w \in [r]$, $\Pr_{h\in H}[h(x)=v \text{ and } h(y)=w] = \Pr_{h\in H}[h(x)=v] \cdot \Pr_{h\in H}[h(y)=w] = \frac{1}{r^2}$

Note: implies $h(x)$ is uniformly distributed in $[r]$, for any x.

Example: Assume r is prime. For any $a,b \in [r]$, let $h_{a,b}(x) := ax+b \bmod r$.
Let $H = \{h_{a,b} : a,b \in [r]\}$

Claim: H is a pairwise independent hash family.

Proof: Fix $x \neq y \in [n]$, $v,w \in [r]$. What is $\Pr_{h\in H}[h(x)=v \text{ and } h(y)=w]$?

It is $\dfrac{|\{h_{a,b} : h_{a,b}(x)=v \text{ and } h_{a,b}(y)=w\}|}{\underbrace{|H|}_{r^2}}$. Claim: numerator is 1 :)

$ax+b=v \pmod r$
$ay+b=w \pmod r$.
Two linearly independent equations in 2 unknowns have a unique solution. ✓

Note: $h_{a,b}$ takes only $O(\log r)$ bits to store, and it can be evaluated with $O(1)$ multiplications and addition mod r.

Fun fact: if $\underbrace{r = 2^{61}-1 \text{ or } r = 2^{127}-1}_{\text{Mersenne primes}}$, then reducing mod r is just a bit-wise shift and & operation, and multiplication mod r can be done with $O(1)$ machine multiplications.

Outline for rest of lecture
- Recall template for designing (ϵ,δ)-multiplicative approximation algorithm for F_0
  - First, develop "basic estimator" that is unbiased but might have high variance. (say variance v)
  - Take the average of "many" pairwise-independent copies of the basic estimator to drive down variance
  - Apply Chebyshev to conclude the above is a $(1\pm\epsilon)$-mult. approx. w.p. $\geq \frac{3}{4}$.
  - Take median of $O(\log(\frac{1}{\delta}))$ copies of the above to get a $(1\pm\epsilon)$-mult. approx w.p. $\geq 1-\delta$.

E.g. last time the basic estimator for F_k had variance at most $V \leq k \cdot n^{1-\frac{1}{k}} \cdot F_k^2$.

So took $t = \frac{V}{\epsilon^2 F_k^2} = \frac{k \cdot n^{1-\frac{1}{k}}}{\epsilon^2}$ mutually independent copies of the basic estimator and looked at their average. We then took the median of $O(\log(\frac{1}{\delta}))$ copies of the above, for total space usage

$$O\left(\frac{k \cdot n^{1-\frac{1}{k}}}{\epsilon^2} \cdot \log\left(\frac{1}{\delta}\right) \cdot \log(n \cdot m)\right)$$

Today: We will give a basic estimator $\hat{F}_0$ that is ~~not~~ close to unbiased

We will ~~not~~ bound its variance so we won't get a (ε,δ)-mult. approx.

We will use Markov's inequality to show it ~~w.p.~~ satisfies

$$\frac{F_0}{10} \le \hat{F}_0 \le 10 F_0 \quad \text{with probability at least } \frac{3}{4}$$

Taking the median of $O(\log(\frac{1}{\delta}))$ copies then gives an estimator $\hat{F}_0'$ s.t. $\frac{F_0}{10} \le \hat{F}_0' \le 10F_0$ with probability at least $1-\delta$.

Tomorrow we will see the ~~"ε-δ template" for obtaining~~ an actual (ε,δ)-mult. approx. alg. ~~same~~ for F_0 following the template

sort of

A Simple (Suboptimal) Distinct Elements Algorithm.

Given a number v, let $\text{zeros}(v)$ denote the number of "trailing zeros" in the binary representation of v. This is the same as the largest i such that 2^i divides v.

Assume $r > n$ is a power of two.

Algorithm: • Choose a random h from a pairwise independent hash family $\mathcal{H}$ of hash functions mapping $[n]$ to $[r]$.

• $z \leftarrow 0$

• For each stream update a_i:

  If $\text{zeros}(h(a_i)) > z$, then $z \leftarrow \text{zeros}(h(a_i))$

• Output 2^z.

i.e. the algorithm simply outputs the largest number of trailing zeros it ever saw in a hash value.

Note: Space usage is $O(\underline{\log\log n})$ bits for storing z, and $O(\log n)$ bits for storing h.

Note also that the algorithm implicitly ignores duplicates and its final estimate is independent ~~[illegible]~~ of the order of the stream.

Intuition: If there are d distinct items, then we expect about $\frac{d}{2^i}$ of them to have at least i trailing zeros. When $i = \log d$, this number is 1, ~~[illegible]~~ When $i >> \log d$, this number is very close to 0, so 2^z should be fairly close to $2^{\log d + \frac{1}{2}} = \sqrt{2}\cdot d$

Analysis: Claim: With probability at least $1 - (\frac{1}{8}+\frac{1}{4}) \geq \frac{5}{8}$, the algorithm will output a number $\hat{d}$ such that $\frac{d}{4} \leq \hat{d} \leq 4d$.

Proof: For each $j \in [n]$ and $t \in \mathbb{Z}_+$, let $X_{t,j} = \begin{cases} 1 & \text{if } \mathrm{zeros}(h(j)) \geq t \\ 0 & \text{otherwise} \end{cases}$ and $Y_t = \sum_{j : f_j > 0} X_{t,j}$. Let z_{max} be the final value of z computed by the algorithm. Note:

$$\left.\begin{aligned} Y_t > 0 &\Leftrightarrow z_{max} \geq t \\ Y_t = 0 &\Leftrightarrow z_{max} \leq t-1 \end{aligned}\right\} \Rightarrow \boxed{\begin{array}{l} \text{So it's enough to show that} \\ \Pr[Y_{\lceil \log(8d) \rceil} \geq 1] \leq \frac{1}{8} \text{ and} \\ \Pr[Y_{\lceil \log(\frac{d}{8}) \rceil} = 0] \leq \frac{1}{4} \end{array}}$$

Fact 1: $\mathbb{E}[Y_t] = \sum_{j : f_j > 0} \mathbb{E}[X_{t,j}] = \sum_{j : f_j > 0} \Pr_{h \in H}[2^t \text{ divides } h(j)] = \sum_{j : f_j > 0} \frac{1}{2^t} = \frac{d}{2^t}$

Fact 2: $\mathrm{Var}[Y_t] \leq \frac{d}{2^t}$.

Proof: $\mathrm{Var}[Y_t] = \mathrm{Var}\left[\sum_{j : f_j > 0} \mathrm{Var}[X_{t,j}]\right) \overset{\text{Pairwise independence of } X_{t,j}\text{s}}{=} \sum_{j : f_j > 0} \mathrm{Var}[X_{t,j}]$

$\leq \sum_{j : f_j > 0} \mathbb{E}[X_{t,j}^2] \overset{X_{t,j} \text{ is } \{0,1\}\text{-valued}}{=} \sum_{j : f_j > 0} \mathbb{E}[X_{t,j}] = \frac{d}{2^t}$.

- Let $a = \lceil \log_2(8d) \rceil$. $\Pr[Y_a \geq 1] \overset{\text{markov}}{\leq} \frac{\mathbb{E}[Y_a]}{1} \leq \frac{d}{2^a} \leq \frac{1}{8}$

- Let $b = \lceil \log_2(\frac{d}{8}) \rceil$. $\Pr[Y_b = 0] \leq \Pr[|Y_b - \underbrace{\mathbb{E}[Y_b]}_{\frac{d}{2^b}}| \geq \frac{d}{2^b}]$

$$\overset{\text{chebyshev}}{\leq} \frac{\mathrm{Var}(Y_b)}{\left(\frac{d}{2^b}\right)^2} \overset{\text{Fact 2}}{\leq} \frac{2^b}{d} \overset{\text{definition of } b}{\leq} \frac{2}{8} = \frac{1}{4} \quad \blacksquare$$

To amplify the $\overset{\text{upper bound}}{\text{probability}}$ of a bad estimate from $\frac{3}{8}$ to δ, repeat $O(\log(\frac{1}{\delta}))$ times and take the median.

Next time: rather than an estimate $\hat{d}$ s.t. $\frac{d}{8} \leq \hat{d} \leq 8d$ with probability $\geq 1-\delta$, we'll see an actual (ϵ, δ) approximation algorithm using space about $O(\log n + \frac{1}{\epsilon^2})$. ← optimal but we'll miss it by a $(\log(\frac{1}{\epsilon}) + \log\log n$ factor).

The rough idea will be to run about $\frac{1}{\epsilon^2}$ copies of the simple algorithm in parallel, and aggregate their estimates into a single, smarter estimate in a clever way