Streaming Algorithms

- Defn: In data stream model, there's an input stream $\sigma =: \langle a_1, \to a_m \rangle$ whose elements are tokens a_i from the universe $[n] = \{1, \to n\}$.

The CPU has space s; which is a rnd-access memory.

Goal: Solve qns. about σ with $s \leq O(\log m + \log n)$.

↳ m or n is HUGE.

1) # distinct elements in σ

- Let $d := \#\{a_1, \dots, a_m\}$. Estimate d by $\hat{d}$ s.t. $d/3 \leq \hat{d} \leq 3 \cdot d$, whp & using space only $s \leq O(\log n)$.

[Trivial: $s = d \in [n]$.] (like constant-many tokens!)

- Alon, Matias & Szegedy (1999): Their algo. uses a p.i.-hash function $h : [n] \to [n]$, that requires only log-space. And the valuation map $v_2(b) := \max\{i : 2^i \mid b\}$.

- Idea: Keep track of $v_2(h(a_i))$, for $a_i \in \sigma$.
  Whp one of the d elements has an h-image divisible by $2^{\lfloor \lg d \rfloor} \approx d$.

- Algo: 0) Choose rnd h from 2.i.-hash-family;
  1) $z \leftarrow 0$;
  2) for (token j in σ)
     3) If $v_2(h(j)) > z$ then $z \leftarrow v_2(h(j))$;
  4) OUTPUT $2^{z+0.5}$;

Analyse:

- For each token $j \in [n]$ & $r \geq 0$, define

$$X_{r,j} := \begin{cases} 1, & \text{if } v_2(h(j)) \geq r \\ 0, & \text{else} \end{cases}$$

& $Y_r := \sum \{X_{r,j} \mid j \text{ s.t. } j \in \sigma\}$.

- Let $T :=$ terminal value of z (when algo. stops).

▷ $Y_r > 0 \iff T \geq r$.

▷ $Y_r = 0 \iff T \leq r-1$.

▷ $E[X_{r,j}] = P(2^r \text{ divides } h(j)) = 1/2^r$.

▷ $E[Y_r] = d/2^r$.

▷ $\mathrm{var}(Y_r) = \sum_{j\in\sigma} \mathrm{var}(X_{r,j}) \le \sum_{j\in\sigma} E[X_{r,j}^2]$

(p.i. hash)

$= \sum_{j\in\sigma} E[X_{r,j}] = E[Y_r] = d/2^2$.

- Output is $\hat{d} := 2^{T+0.5}$.
- Let $a \in \mathbb{Z}$ be the smallest : $2^{a+0.5} \ge 3d$.
  Let $b \in \mathbb{Z}$ " " largest : $2^{b+0.5} \le d/3$.

▷ $P(\hat{d} \ge 3d) = P(T \ge a) = P(Y_a > 0) = P(Y_a \ge 1)$

$\le E[Y_a]/1 = d/2^a \le \frac{\sqrt{2}}{3}$ (< 1).

▷ $P(\hat{d} \le d/3) = P(T \le b) = P(Y_{b+1} = 0)$

$\le P(|Y_{b+1} - \mathbb{E}[Y_{b+1}]| \ge d/2^{b+1}) \le \dfrac{\mathrm{var}(Y_{b+1})}{(d/2^{b+1})^2}$

$= 2^{b+1}/d \le \sqrt{2}/3$.

▷ $P(\frac{d}{3} \le \hat{d} \le 3d) \ge 1 - \frac{2\sqrt{2}}{3} > 0$.

Boosting: The prob. can be made $1 - 2^{-\Omega(k)}$ by running k independent (& parallel) copies of the algo. for the stream σ. And OUTPUT the Median.

Chernoff. bd. gives

Thm 1: $P(d/3 \le \text{output} \le 3d) \ge 1 - 2^{-\Omega(k)}$ & $s \le k \cdot \log n$.

2) Detecting a Heavy-hitter

- Want to check whether σ has element of "unusually high" frequency. (Perhaps, it's an attack on the server!)

- Let f_j := frequency of j in σ.
  - $\sum_{j \in [n]} f_j = m =: F_1$ (first moment)

- Idea: · Compare F_1 with the second moment $F_2 := \sum_j f_j^2$.
  · $F_2 \geq 0.9 \times F_1^2 \Rightarrow$ Perhaps a heavy-hitter!

▷ Compute F_1 by a counter in space $\lg m$.

Qn: How to compute F_2 ?

- [AMS99] algo. to compute F_2 uses a hash. fn.
  $h : [n] \to \{-1, 1\}$ from 4-independent hash family.
  (Exercise: Construct h by using cubic polynomials modulo a prime.)

Algo: 0) Pick rnd 4-indep. hash $h : [n] \to \{\pm 1\}$;
1) $Z \leftarrow 0$;
2) for (token j in σ)
3) $Z \leftarrow Z + h(j)$;
4) Output $Y := Z^2$; // Is Y related to F_2 ?

Analyse: ▷ $Z = \sum_{j \in [n]} f_j \cdot h(j)$.

▷ $E[Z] = \sum_j f_j \cdot E[h(j)] = 0$.

▷ $E[Y] = E\left[\left(\sum_j f_j \cdot h(j)\right)^2\right] = E\left[\sum f_i^2 + \sum_{i \neq j} f_i \cdot f_j \cdot h(i) \cdot h(j)\right]$

$= F_2 + \sum_{i \neq j} f_i \cdot f_j \cdot E[h(i)] \cdot E[h(j)] = F_2$.

↑ 2-wise indep. h

⇒ We're on the right track!

▷ $Var(Y) = E[Y^2] - E[Y]^2 = E[Z^4] - E[Y]^2$

$= E[(\sum_j f_j \cdot h(j))^4] - F_2^2$

$= E[\sum_j f_j^4] + 3 \cdot E[\sum_{i \neq j} f_i^2 f_j^2] - F_2^2$

※ 4-wise-indep. of h

$= \sum_j f_j^4 + 3 \cdot \sum_{i \neq j} f_i^2 f_j^2 - \sum_j (f_j^2)^2 - \sum_{i \neq j} f_i^2 f_j^2$

$= 2 \cdot \sum_{i \neq j} f_i^2 f_j^2$

$\leq 2 \cdot F_2^2$

▷ Chebyshev $\Rightarrow P(|Y - F_2| > \alpha \cdot F_2) \leq \frac{2F_2^2}{(\alpha F_2)^2} = 2/\alpha^2$.

$\Rightarrow$ Y approx. F_2 well!

Boosting: Reduce the variance by keeping k indep. estimators $\{Y_1, \to Y_k\}$ &
OUTPUT $Y' := \frac{Y_1 + \dots + Y_k}{k}$.

▷ $E[Y'] = E[Y] = F_2$ &

▷ $\mathrm{var}(Y') = \mathrm{var}(Y)/k$.

$\Rightarrow$ gives a much better approx. of F_2, in space $O(k \cdot \lg m + k \cdot \lg n)$.