Lecture 10 9/26/2025

AMS estimator and F_2 estimation

Recall we want to estimate frequency moment F_k of a stream $\sigma = e_1, e_2, \ldots, e_m$ $e_i \in [n]$

$$F_k = \sum_{i=1}^{n} (f_i)^k .$$

We will consider a more abstract and general estimation problem.

Let $g_i : \mathbb{R} \to \mathbb{R}$ be a real valued function with $g_i(0) = 0$

Say we want to estimate
$$g(\sigma) = \sum_{i=1}^{n} g_i(f_i)$$

Note that we are allowing different functions for different i.

$$F_k = \sum_{i=1}^{n} g(f_i) \text{ where } g(x) = x^k.$$

Alon-Matias-Szegedy (AMS) in their influential paper obtained an unbiased estimator for computing $g(\sigma)$.

AMS-Estimator(g)

- Sample e_J uniformly at random from stream
- Say $e_J = i$ where $i \in [n]$
- Let $R = |\{j \mid J \leq j \leq m \;\; e_j = i\}|$
- Output $m \cdot (g_i(R) - g_i(R-1))$.

Implementation via reservoir sampling

$s \leftarrow$ null
$m \leftarrow 0$
$R \leftarrow 0$
while (stream is not empty)
    $m \leftarrow m+1$
    e_m current item
    If $(s == e_m)$ $R \leftarrow R+1$

    with prob $\frac{1}{m}$
        $s \leftarrow e_m$
        $R \leftarrow 1$

end while
Output $g_i(R) - g_i(R-1)$ where $s = i$.

Analysis

Lemma: Let Y be the output of the algorithm. Then $E[Y] = g(\sigma)$.

Proof:

$$E[Y] = m \sum_{i=1}^{n} E[Y \mid e_J = i] \Pr[e_J = i]$$

$$= m \sum_{i=1}^{n} \frac{\bar{f}_i}{m} E[Y \mid e_J = i]$$

$$= m \sum_{i=1}^{n} \frac{\bar{f}_i}{m} \cdot \sum_{j=1}^{\bar{f}_i} \frac{g_i(j) - g_i(j-1)}{\bar{f}_i}$$

$$= \sum_{i=1}^{n} g_i(\bar{f}_i).$$

Thus we can use AMS estimator for F_k as well. But we need to understand the variance of the estimator.

<u>Lemma:</u> For $g(\sigma) = \sum_{i=1}^{n} (\bar{f}_i)^k$ with $k \geq 1$

$$Var(Y) \leq k F_1 F_{2k-1} \leq k n^{1-\frac{1}{k}} F_k^2 .$$

<u>Proof:</u> $Var(Y) \leq E[Y^2]$

$$E[Y^2] = \sum_{i=1}^{n} \Pr[e_j = i] \cdot \sum_{l=1}^{\bar{f}_i} \frac{m^2}{\bar{f}_i} \left(l^k - (l-1)^k\right)^2$$

$$\leq \sum_{j=1}^{n} \frac{\bar{f}_i}{m} \cdot \frac{m^2}{\bar{f}_i} \sum_{l=1}^{\bar{f}_i} \left(l^k - l^{k-1}\right)\left(l^k - (l-1)^k\right)$$

$$\leq F_1 \sum_{i=1}^{n} \sum_{l=1}^{\bar{f}_i} k\, l^{k-1} \left(l^k - (l-1)^k\right)$$

$$\leq k F_1 \sum_{i=1}^{n} \bar{f}_i^{\,k-1} f_i^k$$

$$\leq k F_1 F_{2k-1} .$$

$$F_1 F_{2k-1} = \left(\sum_{i=1}^{n} \bar{f}_i\right)\left(\sum_{i=1}^{n} \bar{f}_i^{\,2k-1}\right)$$

$$\leq \left(\sum_{i=1}^{n} \bar{f}_i\right)\left(F_\infty\right)^{k-1}\left(\sum_{i=1}^{n} \bar{f}_i^{\,k}\right)$$

$$\leq \left(\sum_i \bar{f}_i\right)\left(\sum_i f_i^k\right)^{\frac{k-1}{k}} F_k$$

$$\leq n^{1-\frac{1}{k}}\left(\sum_i \bar{f}_i^{\,k}\right)^{\frac{1}{k}}\left(\sum_i f_i^k\right)^{\frac{k-1}{k}} F_k$$

$$\leq n^{1-\frac{1}{k}} F_k^2 .$$

Since $\mathrm{Var}(Y) \leq k n^{1-\frac{1}{k}} F_k^2$

and $E[Y] = F_k$

Using median trick we can get an (ε, δ) approx using $O\left(\frac{1}{\varepsilon^2} n^{1-\frac{1}{k}} \log\frac{1}{\delta}\right)$ space.

For F_2 estimation we get (ε, δ) approx in $\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^2} \log \frac{1}{\delta}\right)$ space.

Can we do better?

For F_2 AMS showed that

$$O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta} \operatorname{polylog}(n)\right) \text{ suffices!}$$

For $k > 2$ the right bound is

$$\tilde{O}\left(n^{1-\frac{2}{k}}\right).$$

For $1 \leq k \leq 2$ one can get

$$O\left(\frac{1}{\varepsilon} \log \frac{1}{\delta} \operatorname{polylog}(n)\right).$$

AMS F_2 Estimation

- Choose $h : [n] \to \{-1, 1\}$ from a 4-wise independent hash family $\mathcal{H}$.
- $z \leftarrow 0$
- While (stream is not empty)

  e_m is current element

  $z \leftarrow z + h(e_m)$.

  end While
- Output z^2.

4-wise independent family can be stored via $O(1)$ $\log n$ bit numbers

z requires one number.

Analysis

Let $Y_i = h(i)$ $\qquad Y_i \in \{-1, 1\}$

$E[Y_i] = 0 \quad E[Y_i^2] = 1$

$Y_1, Y_2, \ldots, Y_n$ are 4-wise independent

$$Z = \sum_{i=1}^{n} \bar{f}_i Y_i$$

output is Z^2

$$E[Z^2] = \sum_{i=1}^{n} \bar{f}_i^2 E[Y_i^2] + 2\sum_{i \neq j} f_i f_j \cdot E[Y_i Y_j]$$

$$= \sum_{i=1}^{n} \bar{f}_i^2 = F_2.$$

$$Var(Z^2) = E[Z^4] - F_2^2$$

$$E[Z^4] = E\Big[\sum_i \sum_j \sum_k \sum_l f_i f_j f_k f_l Y_i Y_j Y_k Y_l\Big]$$

any term with only one occurrence of a term Y_i becomes 0.

$$= \sum_{i=1}^{n} f_i^4 \qquad + 6 \sum_{i=1}^{n} \sum_{j=i+1}^{n} f_i^2 f_j^2$$

Thus

$$Var(Z^2) = \sum_{i=1}^{n} f_i^4 + 6 \sum_{i=1}^{n} \sum_{j=i+1}^{n} f_i^2 f_j^2$$

$$- \left(\sum_{i=1}^{n} f_i^2 \right)^2$$

$$= 4 \sum_{i=1}^{n} \sum_{j=i+1}^{n} f_i^2 f_j^2$$

$$\leq 2 F_2^2 .$$

$\square$

$\Rightarrow$ (ε, δ) approx requires

$$O\left(\frac{1}{\varepsilon^2} \ln \frac{1}{\delta}\right) \text{ counters.}$$

We make an observation that non-negativity of f_i did not play a role in the proof.

This will lead us to a generalization of the streaming model.

Recall we had $e_t \in [n]$ for each time t.

Now we have $e_t = (i_t, \Delta_t)$ where $i_t \in [n]$ and Δ_t is an update to cordinate i_t.

We use $\bar{x} \in \mathbb{R}^n$ that starts at $\bar{0}$ and is updated as

$$x_{i_t} = x_{i_t} + \Delta_t \quad \text{after } e_t.$$

Now F_2 becomes $\|\bar{x}\|_2^2$.

We see that the AMS-F_2 estimator works in this model.

AMS F_2 Estimation

- Choose $h : [n] \to \{-1, 1\}$ from a 4-wise independent hash family $\mathcal{H}$.

- $z \leftarrow 0$
- While (stream is not empty)

  $e_t \leftarrow (i_t, \Delta_t)$

  $z \leftarrow z + \Delta_t h(i_t)$.

  end While
- Output z^2.

Exercise: $E[z^2] = \|x\|_2^2$

$\mathrm{Var}(z^2) \leq 2\|x\|_2^4$ . .

But $\|x\|_2^2$ is the length of the vector $\bar{x}$.

Should remind you of dimensionality reduction!

Interpreting AMS-F_2 estimator as a "<u>linear</u> <u>sketch</u>".

We can view the streaming computation as

$$\begin{bmatrix} +1 & -1 & \cdots & +1 & -1 \end{bmatrix} \begin{bmatrix} \\ \bar{x} \\ \\ \end{bmatrix} = z.$$

$\{-1,1\}^n$ vector obtained from h.

Recall that in dimensionality reduction we picked a $K \times n$ matrix G where we chose G_{ij} as an independent Gaussian.

In F_2 estimation we are picking A where each row of A is obtained from a 4-wise independent hash function with entries in $\{-1, 1\}$.

$$\begin{bmatrix} +1 & -1 & +1 & \cdots & -1 \\ +1 & & & & \\ & & & & \end{bmatrix}$$

In dimensionality reduction we chose $K = \Theta(\frac{1}{\varepsilon^2} \ln \frac{1}{\delta})$ and showed that $\| \frac{1}{\sqrt{k}} G \bar{x} \|$ is a

(ε, δ) approximation to $\|x\|_2$.

but in F_2 estimation we seem

to be getting same!

$\Theta\left(\frac{1}{\varepsilon^2} \ln \frac{1}{\delta}\right)$ rows suffice to get

ε-approx with prob $(1-\delta)$

But we are only using 4-wise

independence in each row. And

also $\{-1, 1\}$.

Why not use this for dimensionality

reduction?

The difference is the following.

Ax with $k = \Theta(\frac{1}{\varepsilon^2} \ln \frac{1}{\delta})$ has sufficient information to recover a $(1-\varepsilon)$ approx for $\|x\|_2$ but $c\|Ax\|_2$ is itself not the way we compute the approximation.

We use median estimator which is not a <u>linear</u> function.

Nevertheless the information that the algorithm computes is a linear sketch. <u>Ax</u>.

A sketch of a data stream σ is some function $C(\sigma)$ that is a compact representation of σ.

We want sketches to have <u>composability</u>. Give σ_1 and σ_2 and sketches $C(\sigma_1)$ and $C(\sigma_2)$ we would like to compute $C(\sigma_1 \cdot \sigma_2)$ from $C(\sigma_1)$ and $C(\sigma_2)$.

A particularly nice sketch is a linear sketch.

$$C(\sigma) = C(\sigma_1) + C(\sigma_2).$$

The F_2 estimate can be seen as a linear sketch.

$$A\bar{x} \qquad \begin{array}{c} h_1 \\ h_2 \\ \vdots \\ h_k \end{array} \begin{bmatrix} - & - & - & - & - \\ - & - & - & - \\ \vdots \\ - & - & - & - \end{bmatrix}$$

each row corresponds to a hash function. Note that the way we use the output of the sketch to compute some information about the data can be some non-linear function of the sketch itself. Linear sketches naturally allow for <u>deletions</u>.