Algorithms for Big Data (Fall 25)

Lecture 6

F_2-Estimation, AMS Sketch, Heavy Hitters

Ali Vakilian (vakilian@vt.edu)

VIRGINIA TECH

Estimating F_2

Input: A data stream $S = (e_1, e_2, e_3, \dots, e_N)$, that are seen one by one, where each $e_i \in [n]$ (for known n or an upper bound on n).

- Let f_i denotee the frequency of item i in the stream
- Consider vector $\boldsymbol{f} = (f_1, \dots, f_n)$

The Goal: Compute F_2

The generic AMS estimator gives $(1 \pm \epsilon)$-estimation in $O(\frac{1}{\epsilon^2}\sqrt{n})$ space.

Can we do better?

k-wise independence

Consider flipping two unbiassed coins, and define following three events:

- **A:** The first coin is Heads ($\Pr(\mathbf{A}) = 1/2$)
- **B:** The second coin is Heads ($\Pr(\mathbf{B}) = 1/2$)
- **C:** Two coins show different outcomes ($\Pr(\mathbf{C}) = 1/2$)

Are they pairwise independent?

- $\Pr(\mathbf{A} \cap \mathbf{B}) = 1/4$
- $\Pr(\mathbf{A} \cap \mathbf{C}) = 1/4$ (and similarly $\Pr(\mathbf{A} \cap \mathbf{C}) = \mathbf{1/4}$)

Are they 3-wise independent?

- $\Pr(\mathbf{A} \cap \mathbf{B} \cap \mathbf{C}) = 0 \neq 1/8 = \Pr(\mathbf{A}) \times \Pr(\boldsymbol{B}) \times \Pr(\mathbf{C})$

Estimating F_2 (AMS data structure)

$Z = \sum_{i=1}^{n} f_i \cdot Y_i$

- $\mathbb{E}[Y_i] =$
- $\mathrm{Var}[Y_i] = \mathbb{E}[Y_i^2] =$
- For $i \neq j$, $\mathbb{E}[Y_i Y_j] =$

$Z^2 = \sum_{i=1}^{n} f_i^2 \cdot Y_i^2 + 2\sum_{i \neq j} f_i f_j Y_i Y_j$

So, $\mathbb{E}[Z^2] = \sum_{i=1}^{n} f_i^2 = F_2$

AMS-F_2-Estimate (stream):

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

$z \leftarrow 0$

foreach item e_i in the stream:

$z \leftarrow z + h(e_i)$

return z^2

let $Y_1, \ldots, Y_n$ be 4-wise independent r.v.s

$z \leftarrow z + Y_{e_i}$

Variance Computation of AMS Estimator

$\mathrm{Var}[Z^2] = \mathbb{E}[Z^4] - \left(\mathbb{E}[Z^2]\right)^2$. So, needs to bound $\mathbb{E}[Z^4]$.

$$\mathbb{E}[Z^4] = \sum_{i\in[n]}\sum_{j\in[n]}\sum_{k\in[n]}\sum_{\ell\in[n]} f_i f_j f_k f_\ell \mathbb{E}[Y_i Y_j Y_k Y_\ell]$$

4-wise independence implies $\mathbb{E}\left[Y_i Y_j Y_k Y_\ell\right] = 0$ if there is an index among i, j, ℓ, k that occurs only once; otherwise, it is 1.

$$\mathbb{E}[Z^4] = \sum_{i\in[n]}\sum_{j\in[n]}\sum_{k\in[n]}\sum_{\ell\in[n]} f_i f_j f_k f_\ell \mathbb{E}[Y_i Y_j Y_k Y_\ell]$$
$$= \sum_{i\in[n]} f_i^4 + 6 \cdot \sum_{i\in[n]}\sum_{j\in[i+1\cdots n]} f_i^2 f_j^2$$

$\boldsymbol{F_4}$

Variance Computation of AMS Estimator

$$
\begin{aligned}
\mathrm{Var}[Z^2] &= \mathbb{E}[Z^4] - \left(\mathbb{E}[Z^2]\right)^2 \\
&= F_4 + 6 \cdot \textstyle\sum_{i\in[n]} \sum_{j\in[i+1\cdots n]} f_i^2 f_j^2 - \left(\mathbb{E}[Z^2]\right)^2 \\
&= F_4 + 6 \cdot \textstyle\sum_{i\in[n]} \sum_{j\in[i+1\cdots n]} f_i^2 f_j^2 - \left(F_4 + 2 \cdot \sum_{i\in[n]} \sum_{j\in[i+1\cdots n]} f_i^2 f_j^2\right) \\
&= 4 \cdot \textstyle\sum_{i\in[n]} \sum_{j\in[i+1\cdots n]} f_i^2 f_j^2 \\
&\le 2 \cdot F_2
\end{aligned}
$$

Estimating F_2 (AMS data structure)

$\mathbb{E}[Z^2] = F_2$ and $\text{Var}[Z^2] \leq 2F_2^2$

- Averaging $\frac{8}{\epsilon^2}$ estimator
  - By Chebyshev, $(\epsilon, \frac{1}{4})$-relative estimate
- Reduce error by median trick over $O(\log 1/\delta)$ averaged estimators,
  - By Chernoff, (ϵ, δ)-relative estimate

Total space is $\boldsymbol{O(\log\frac{1}{\delta} \cdot \frac{1}{\epsilon^2} \cdot \log n)}$

AMS-F_2-Estimate (stream):

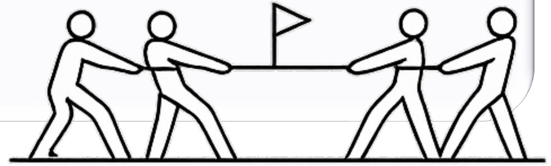
let $h: [n] \to \{-1,1\}$ be chosen from a 4-wise independent hash family $\mathcal{H}$

$z \leftarrow 0$

foreach item e_i in the stream:

$z \leftarrow z + h(e_i)$

return z^2



Tug-of-War sketch

Negative Updates

- So far, we only studied the “insertion only” streaming.

- However, we can think of the setting where an item can be deleted.

    - Amazon inventory management: (items may be added to the inventory, or sold)

    - Bank Account Balances: (a deposit to, or withdrawal from the account)

Negative Updates

- So far, we only studied the "insertion only" streaming.

- However, we can think of the setting where an item can be deleted.
- In case of vector processing, we may allow for update $\Delta_i \in \{-1,1\}$ on item e_i
  - i.e., $f_{e_i} \leftarrow f_{e_i} + \Delta_i$
- In particular, at the end, some coordinates may have negative values.

Can we still use AMS algorithm to compute F_2 in this setting?

In **HW 1**: You'll answer this question.

Linear Sketching

Linear Sketch

A sketch of a stream S is a summary data structure $\text{sketch}(S)$ (most relevant to our goal if it is of small size).

Does the AMS algorithm provide a **sketch** for F_2-estimation?

A sketch is linear, if

$$\text{sketch}(S_1 \circ S_2) = \text{sketch}(S_1) \circ \text{sketch}(S_1)$$

Equivalently, $\text{sketch}(S) = \Pi\, S \quad (S \in \mathbb{R}^m$ and $\Pi \in \mathbb{R}^{k \times m})$

Does the AMS algorithm provide a **linear sketch** for F_2-estimation?

AMS as a Sketch

AMS-F_2-Sketch:

let $m = c \log\left(\frac{1}{\delta}\right)/\epsilon^2$

let Π be a $m \times n$ matrix with $\{-1, +1\}$ entries

(i) rows are independent and

(ii) in each row, entries are 4-wise indep.

$z \leftarrow 0$ is a $m \times 1$ vector initialized to $\mathbf{0}$

foreach item $\boldsymbol{i_j}$ in the stream do:

$z \leftarrow z + Me_{i_j}$

return z as sketch

AMS as a Sketch

- the sketch is mergeable.
- $z_{S \cup T} = z_S + z_T$

How to get the final estimate from the sketch z?

$$\hat{F}_2 = \text{median}_{g=1\ldots k}\left(\frac{1}{t}\sum_{j\in G_g} {z_j}^2\right)$$

where $G_1, \ldots, G_k$ are partition of the m rows ($m = tk$).

AMS-F_2-Sketch:

let $m = k \times t$

let Π be a $m \times n$ matrix with $\{-1, +1\}$ entries

(i) rows are independent and

(ii) in each row, entries are 4-wise indep.

$z \leftarrow 0$ is a $m \times 1$ vector initialized to $\mathbf{0}$

foreach item $\boldsymbol{i_j}$ in the stream do:

$z \leftarrow z + Me_{i_j}$

return z as sketch

Sketching

- **Sketching is a powerful algorithmic technique** with broad applications in both theory and practice.

- **Linear sketches are a particularly important subclass**, as they naturally handle dynamic data streams that include deletions or negative updates.

- The theory of sketching is **deeply connected to fundamental concepts** like dimensionality reduction (the Johnson-Lindenstrauss Lemma) and subspace embeddings.

Heavy Hitters

Streaming Models: Revisit

- The goal is to estimate a function of a vector $x \in \mathbb{R}^n$ which is initially all 0's vector.
- Each element e_j of the stream is a tuple of (i_j, Δ_j) where $i_j \in [n]$ and $\Delta_j \in \mathbb{R}$ is the update to coordinate i_j; i.e., this updates

$$x_{i_j} \leftarrow x_{i_j} + \Delta_j$$

  - $\Delta_j > 0$ for all j: cash register, insertion only (when $\Delta_j = 1$) streams
  - Δ_j is arbitrary: dynamic or turnstile stream
  - Δ_j is arbitrary but $x \geq 0$ at all times: strict turnstile model

Frequent Items Problem (F_∞-Moment)

Recall: What is F_∞?

- F_∞ is very brittle and hard to estimate with low memory.
- Even strong lower bounds even for very weak relative approximations.

Hence, we need to settle for weaker (additive) guarantees.

Heavy Hitters Problem: Find all items i such that $f_i > \frac{m}{k}$ for some fixed k.

Heavy hitters are **very** frequent items.

Finding Majority Element (interview question)

Offline: given an array/list A of m integers, is there an element that occurs more than $m/2$ times in A?

Streaming: is there an i such that $f_i > m/2$?

Boyer-Moore Voting

Lemma. If there exists a majority item i, the algorithm outputs $s = i$ and $c \geq f_i - \frac{m}{2}$.

Why it works?

Majority (in streams):

```
let c ← 0, s ← null

foreach item e_j in the stream do:
    if e_j = s then
        c ← c + 1
    else if c = 0
        c ← 1 and s ← e_j
    else
        c ← c − 1
return c and s
```

Boyer-Moore Voting

Lemma. If there exists a majority item i, the algorithm outputs $s = i$ and $c \geq f_i - \frac{m}{2}$.

Why it works?

What if no majority item exists?

How to verify?

Majority (in streams):

let $c \leftarrow 0, s \leftarrow null$

foreach item e_j in the stream do:

  if $\boldsymbol{e_j} = s$ **then**

    $c \leftarrow c + 1$

  else if $c = 0$

    $c \leftarrow 1$ and $s \leftarrow \boldsymbol{e_j}$

  else

    $c \leftarrow c - 1$

return c and s