Lecture 6: F_2 approximation, AMS sketching, Heavy Hitters

09-11-2025 Lecturer: Ali Vakilian | Scribe: Reetish Padhi | Editor: Ali Vakilian

1 F_2 Estimation Problem

In the previous lecture, we studied a general method to compute the k-th moment $\mathbf{F}_k$ of a given stream. We recall that the generic AMS sampler for $\mathbf{F}_2$ gives us a $(1 \pm \varepsilon)$ - estimation in $O(\frac{\sqrt{n}}{\epsilon^2})$ space.

In this lecture, we focus our attention to the case where $k = 2$. Formally the problem can be stated as, given a stream $E = (e_1, e_2, \ldots, e_N)$ where $e_i \in [n], 1 \leq i \leq N$ with $f = (f_1, f_2, \ldots, f_n)$ denoting the frequencies of each item $j \in [n]$. We also look to improve on the generic AMS estimator.

Optional Reading: note on k-wise independence

Definition 1.1 (k-wise Independence)**.** A set of random variables $Y_1, Y_2, \ldots, Y_n$ is said to be k-wise independent if for any subset of k distinct indices $I = \{i_1, i_2, \ldots, i_k\} \subseteq [n]$, and any corresponding values $y_1, y_2, \ldots, y_k$, we have,

$$\Pr(Y_{i_1} = y_1, Y_{i_2} = y_2, \ldots, Y_{i_k} = y_k) = \prod_{j=1}^{k} \Pr(Y_{i_j} = y_j)$$

In essence, any subset of k variables behaves as if they were fully independent. We note that k wise independence implies j wise independence for $j < k$.

Example 1.2. Consider flipping two fair coins. Let the events be:

- A: The first coin is heads, $\Pr(A) = 1/2$.
- B: The second coin is heads, $\Pr(B) = 1/2$.
- C: The two coins show different outcomes, $\Pr(C) = 1/2$.

We claim that these events are pairwise (2-wise) independent.

To verify this we can simply check that, $\Pr(A \cap B) = 1/4 = \Pr(A)\Pr(B)$, $\Pr(A \cap C) = 1/4 = \Pr(A)\Pr(C)$ and $\Pr(B \cap C) = \Pr(B)\Pr(C) = 1/4$. However, they are not 3-wise independent, because $\Pr(A \cap B \cap C) = 0$, but $\Pr(A)\Pr(B)\Pr(C) = 1/8$. Alternatively, one can also verify that $\Pr(C|A) = \Pr(C)$ and a similar equality holds for the other pairs.

2 AMS Estimator

In their seminal paper, Alon et al. [1996] proposed the efficient AMS-$\mathbf{F}_2$ estimation algorithm as described below.

Algorithm 1 AMS-Sample (Stream)

let $h : [n] \to \{-1, 1\}$ be chosen from a 4-wise independent hash family $\mathcal{H}$
$z \leftarrow 0$
for each item e_i in the stream **do**
    $z \leftarrow z + h(e_i)$
return z^2

We must first show that the estimator z^2 is unbiased. But first, let us define the set of random variables $Y_1, \ldots, Y_n$, where $Y_i = h(e_i)$ and by construction of the hash function is chosen uniformly from $\{-1, 1\}$. We require these variables to be *4-wise independent.* A simple calculation reveals the following identities,

- $\mathbb{E}[Y_i] = (-1) \cdot \frac{1}{2} + (1) \cdot \frac{1}{2} = 0.$
- $\mathbb{E}[Y_i^2] = (-1)^2 \cdot \frac{1}{2} + (1)^2 \cdot \frac{1}{2} = 1.$
- For $i \neq j$, by pairwise independence, $\mathbb{E}[Y_i Y_j] = \mathbb{E}[Y_i]\mathbb{E}[Y_j] = 0.$

The AMS estimator is the random variable $Z = \sum_{i=1}^{n} f_i \cdot Y_i$. In the streaming context, we initialize a counter $z = 0$. For each item e_j in the stream, we update $z \leftarrow z + Y_{e_j}$. The final value of z is our random variable Z. The estimate for F_2 is Z^2.

Lemma 2.1. *Z^2 is an unbiased estimator for F_2.*

Proof.

$$\begin{aligned}
\mathbb{E}[Z^2] = \mathbb{E}\left[\left(\sum_{i=1}^{n} f_i Y_i\right)^2\right] &= \mathbb{E}\left[\sum_{i=1}^{n} f_i^2 Y_i^2 + \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} f_i f_j Y_i Y_j\right] \\
&= \sum_{i=1}^{n} f_i^2 \mathbb{E}[Y_i^2] + \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} f_i f_j \mathbb{E}[Y_i Y_j] \quad \text{(by linearity of expectation)} \\
&= \sum_{i=1}^{n} f_i^2 \cdot (1) + \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} f_i f_j \cdot (0) = \sum_{i=1}^{n} f_i^2 = F_2
\end{aligned}$$

Thus, Z^2 is an unbiased estimator for F_2. □

To analyze the quality of the estimator, we need to bound the variance of Z^2 too.

Lemma 2.2. $\mathrm{Var}[Z^2] \leq 2F_2^2$.

Proof. $\mathrm{Var}[Z^2] = \mathbb{E}[Z^4] - (\mathbb{E}[Z^2])^2$. We already know $\mathbb{E}[Z^2] = F_2$. Now we compute $\mathbb{E}[Z^4]$.

$$\mathbb{E}[Z^4] = \mathbb{E}\left[\left(\sum_{i=1}^{n} f_i Y_i\right)^4\right] = \sum_{i,j,k,l \in [n]} f_i f_j f_k f_l \mathbb{E}[Y_i Y_j Y_k Y_l]$$

This is where 4-wise independence is crucial. For the term $\mathbb{E}[Y_i Y_j Y_k Y_l]$, if any index appears an odd number of times (e.g., just once), the expectation is 0. To see this, suppose i is such that from $i \neq j, k, l$, then $\mathbb{E}[Y_i Y_j Y_k Y_l] = \mathbb{E}[Y_i]\mathbb{E}[Y_j Y_k Y_l] = 0$ from what we computed earlier. The non-zero terms arise only when each index appears an even number of times. This leads to two possibilities,

1. $i = j = k = l$: Terms are of the form $f_i^4 \mathbb{E}[Y_i^4] = f_i^4 \cdot 1 = f_i^4$. The sum of these is $\sum f_i^4 = F_4$.

2. Indices appear in two pairs, e.g., $i = j, k = l$ with $i \neq k$: Terms are of the form $f_i^2 f_k^2 \mathbb{E}[Y_i^2 Y_k^2] = f_i^2 f_k^2 \mathbb{E}[Y_i^2]\mathbb{E}[Y_k^2] = f_i^2 f_k^2$. There are $3\binom{n}{2}$ such distinct terms. Thus, the sum containing terms of this form can be written as $6 \sum_{i<j} f_i^2 f_j^2$.

So this leads to the equation, $\mathbb{E}[Z^4] = \sum_{i=1}^n f_i^4 + 6 \sum_{i<j} f_i^2 f_j^2$. We also have the equality,

$$(F_2)^2 = \left(\sum f_i^2\right)^2 = \sum f_i^4 + 2 \sum_{i<j} f_i^2 f_j^2.$$

The variance of Z^2 can now be computed as follows,

$$\begin{aligned}
\mathrm{Var}[Z^2] = \mathbb{E}[Z^4] - (\mathbb{E}[Z^2])^2 &= \left(\sum f_i^4 + 6 \sum_{i<j} f_i^2 f_j^2\right) - \left(\sum f_i^2\right)^2 \\
&= \left(\sum f_i^4 + 6 \sum_{i<j} f_i^2 f_j^2\right) - \left(\sum f_i^4 + 2 \sum_{i<j} f_i^2 f_j^2\right) \\
&= 4 \sum_{i<j} f_i^2 f_j^2 \leq 2 \left(\sum f_i^2\right)^2 = 2F_2^2
\end{aligned}$$

Finally, we get the following bound on the variance of Z^2, $\mathrm{Var}[Z^2] \leq 2F_2^2$. □

2.1 Achieving an (ε, δ)-Approximation

Since the variance $\mathrm{Var}[Z^2] \leq 2F_2^2$ is large, a single estimator Z^2 is not reliable. We improve its accuracy using the usual averaging and median trick,

1. **Averaging:** Run $k = O(1/\varepsilon^2)$ independent copies of the algorithm to get estimators $X_1, \ldots, X_k$. Let $\bar{X} = \frac{1}{k} \sum X_i$. By Chebyshev's inequality, this average provides an $(\varepsilon, 1/4)$-relative estimate.

2. **Median Trick:** To boost the success probability, run $m = O(\log(1/\delta))$ parallel groups of averaged estimators. The final estimate is the median of the results from these m groups. This gives an (ε, δ)-relative estimate.

The total space required is $O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta} \log n)$ to store the counters and the description of the hash functions.

2.2 Negative updates

So far the examples only consider streams where elements are being inserted into the stream. However, in some practical settings we might have applications where the frequency of elements may be reduced (or even made negative) as we pass through the stream. Some examples include, Amazon inventory management and bank balances. A fundamental difference in these two applications is that bank balances can be negative (in cases of overdraft) however, inventory cannot fall below 0.

3 Linear Sketching

Definition 3.1 (Linear Sketch)**.** A sketch is a small summary of a data stream (or more generally a large scale data). A sketch is *linear* if the sketch of a concatenated stream can be computed from the sketches of the individual streams as shown below,

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

In particular, linear sketches are of form

$$\text{sketch}(S) = \Pi \cdot S,$$

where Π is a $k \times n$ matrix for some small k (ideally, $k \ll n$).

For example, the described AMS algorithm provides a linear sketch for F_2 estimation. The sketch is the vector of counters $z = (z_1, \ldots, z_k)^\top$, where each z_j is an independent AMS estimator. The sketch matrix Π has entries $\Pi_{ji} = h_j(i)$, where h_j is the 4-wise independent hash function for the j-th estimator. Linear sketches are powerful because they naturally handle dynamic streams, where items can be inserted or deleted (negative updates).

3.1 The AMS Algorithm as a Linear Sketch

The full AMS algorithm, incorporating averaging and the median trick to achieve an (ϵ, δ)-approximation, can be formalized as a single linear sketch.

Algorithm 2 AMS-$\mathbf{F}_2$ as linear sketching algorithm

1: let $m = k \times t$
2: Let Π be a $m \times n$ matrix with entries in $\{-1, 1\}$
   1. rows are independent
   2. elements of each row are 4-wise independent
3: $z \leftarrow 0$ is a $m \times 1$ vector
4: **for** each item i_j **do**
5: $\quad z \leftarrow z + M(e_{i_j})$
6: **return** z (the sketch of the stream)

After processing the entire stream, the sketch vector z holds the values of m different estimators. To get the final, robust estimate for F_2, we perform the averaging and median steps on the components of z. This gives us the final estimate,

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

where G_g are the partitions of m rows $(m = tk)$.

Takeaway 3.1

The AMS algorithm can be formulated as a linear sketching algorithm. We note that the sketch z derived from the AMS algorithm is linear however the final estimator $\hat{F}_2$ is a nonlinear function of z.

4 Heavy Hitters

The heavy hitters problem is about finding the most frequent items in a stream. This is related to the F_∞ moment, but since F_∞ is hard to estimate, we aim for slightly different guarantees. Given a stream and a parameter k, the goal is to find all items i such that their frequency f_i exceeds a certain threshold, for example $f_i > m/k$, where m is the total length of the stream.

4.1 Finding the Majority Element

A simple version of this problem is finding an item that appears more than $m/2$ times. The Boyer-Moore Voting algorithm solves this using constant space.

Algorithm 3 Boyer-Moore Voting algorithm

1: Initialize a counter $c \leftarrow 0$ and a stored item $s \leftarrow$ null.
2: For each item e_j in the stream:
- If $e_j = s$, increment the counter: $c \leftarrow c + 1$.
- Else if $c = 0$, set $s \leftarrow e_j$ and $c \leftarrow 1$.
- Else, decrement the counter: $c \leftarrow c - 1$.

3: Return s as the candidate majority item.

Lemma 4.1. *If there is a majority item i (with frequency $f_i > m/2$), the Boyer-Moore algorithm will return $s = i$.*

Proof sketch. Consider the true majority item, i. Each time we see an item that is *not* i, its effect is to potentially decrement the counter c. In the worst case, every non-i item pairs up with an instance of i to decrement the counter. Since there are more occurrences of i than all other items combined, the counter for i can never be zeroed out by the non-i items once s is set to i. □

Note: If no majority item exists, the algorithm will still return a candidate. A second pass over the data is required to verify if the returned candidate s is truly a majority element.

References

Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29, 1996.