

Data Stream Algorithms

In this lecture, the space complexity of approximating frequency moments in the data stream model is discussed.

In the *data stream model*, a computer has access to a data stream $A = (a_1, a_2, \dots, a_n)$ such that each data item $a_i \in T$ ($i = 1, \dots, n$, where n is the size of the stream and $T = \{1, \dots, t\}$) can be read only once in an order that is unknown to the computer.

For each $i \in T$, let

$$m_i = |\{j \mid a_j = i\}|$$

be the number of occurrences of i in the data stream A .

Definition. For each $k \in \mathbb{N}$, the k th frequency moment of $\{m_i\}$ is

$$F_k = \sum_{i=1}^t m_i^k.$$

Note that F_0 is the number of distinct elements in A ; F_1 is the size of A .

Definition. An algorithm \mathcal{A} computes an (ϵ, δ) -approximation of F_k if

$$\Pr[|\mathcal{A}(A) - F_k| \geq \epsilon F_k] < \delta.$$

Theorem 1. Let X_i ($i = 1, \dots, n$) be n (pair-wise) independent identically distributed random variables such that $\mathbf{E}[X_i] < \infty$ and $\mathbf{Var}[X_i] < \infty$. Then

$$\mathbf{Var} \left[\frac{1}{n} \sum_{i=1}^n X_i \right] = \frac{\mathbf{Var}[X_i]}{n}.$$

Theorem 2. (Chebyshev's inequality) Let X be a random variable with expected value E and variance V . Then for any $\epsilon > 0$,

$$\Pr[|X - E| \geq \epsilon\sqrt{V}] \leq \frac{1}{\epsilon^2}.$$

Theorem 3. Let X_1, X_2, \dots, X_n be independent identically distributed random variables with $\mathbf{E}[X_1] = E$ and $\mathbf{Var}[X_1] = V$. Let $X = \frac{1}{n} \sum_{i=1}^n X_i$. Then

$$\Pr[|X - E| \geq \epsilon E] \leq \frac{V}{n\epsilon^2 E^2}.$$

Proof. Note that $\mathbf{Var}[X] = \frac{\mathbf{Var}[X_1]}{n} = \frac{V}{n}$.

$$\begin{aligned} \Pr[|X - E| \geq \epsilon E] &= \Pr\left[|X - E| \geq \epsilon E \frac{\sqrt{\mathbf{Var}[X]}}{\sqrt{\mathbf{Var}[X]}}\right] \\ &\leq \frac{\mathbf{Var}[X]}{\epsilon^2 E^2} && \text{Chebyshev} \\ &= \frac{V}{n\epsilon^2 E^2}. \end{aligned}$$

□

Theorem 4. (Chernoff's inequality) Let X_i ($i = 1, \dots, n$) be independent identically distributed 0-1 random variables. Let $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$. Then $\mathbf{E}[\bar{X}] = \mathbf{E}[X_i]$ and for all $\epsilon \in [0, 1]$,

$$\Pr[|\bar{X} - \mathbf{E}[\bar{X}]| \geq \epsilon \mathbf{E}[\bar{X}]] \leq 2 \cdot e^{-\frac{2}{3} \epsilon^2 \mathbf{E}[\bar{X}] n}.$$

Theorem 5. Let \mathcal{A} be an $(\epsilon, \frac{1}{3})$ -approximation of F , i.e.,

$$\Pr[|A - F| \geq \epsilon F] < \frac{1}{3}.$$

Let $n = \lceil 243 \ln \frac{2}{\delta} \rceil$. Let X_1, X_2, \dots, X_n be the outputs of n independent runs of \mathcal{A} . Let X be the median of X_1, X_2, \dots, X_n . Then X is an (ϵ, δ) -approximation of F .

Proof. Define random variable T_i such that

$$T_i = \begin{cases} 1 & \text{if } |X_i - F| < \epsilon F \\ 0 & \text{otherwise.} \end{cases}$$

Then $\Pr[T_i = 1] \geq \frac{2}{3}$. Define $\bar{T} = \frac{1}{n} \sum_{i=1}^n T_i$. Then $\mathbf{E}[\bar{T}] \geq \frac{2}{3}$. Note that if more than half of X_i 's satisfies $|X_i - F| < \epsilon F$ then $|X - F| < \epsilon F$, i.e., if $\bar{T} > \frac{1}{2}$ then $|X - F| < \epsilon F$. And

$$\begin{aligned} \Pr[|X - F| \geq \epsilon F] &\leq \Pr[\bar{T} \leq \frac{1}{2}] \\ &\leq \Pr[|\mathbf{E}[\bar{T}] - \bar{T}| \geq \mathbf{E}[\bar{T}] - \frac{1}{2}] \\ &= \Pr\left[|\mathbf{E}[\bar{T}] - \bar{T}| \geq \left(1 - \frac{1}{2\mathbf{E}[\bar{T}]} \right) \mathbf{E}[\bar{T}]\right]. \end{aligned}$$

By the Chernoff's inequality (Theorem 4),

$$\begin{aligned} \Pr[|X - F| \geq \epsilon F] &\leq 2 \cdot e^{-\frac{1}{3} \left(1 - \frac{1}{2\mathbf{E}[\bar{T}]}\right)^2 \mathbf{E}[\bar{T}] n} \\ &= 2 \cdot e^{-\frac{1}{3} (\mathbf{E}^2[\bar{T}] - \frac{1}{2} \mathbf{E}[\bar{T}])^2 n}. \end{aligned}$$

Since $\mathbf{E}[\bar{T}] \geq 2/3$, we have

$$\begin{aligned} \Pr[|X - F| \geq \epsilon F] &= 2 \cdot e^{-\frac{1}{3} \left(\frac{4}{9} - \frac{1}{2} \frac{2}{3}\right)^2 n} \\ &= 2 \cdot e^{-\frac{1}{243} n} \\ &\leq \delta, \end{aligned}$$

i.e., X is an (ϵ, δ) -approximation of F . □

1 The First Frequency Moment — F_1

$c=0$ when a data item comes increment c with probability 2^{-c} output 2^c .
Algorithm $\mathcal{A}_{1,1}$

Let X_i ($i \in \{0, 1, \dots, n\}$) be the value of the counter c after reading i items, where n is the size of data stream A . The output of the algorithm is 2^{X_n} .

$$\Pr[X_n = a] = \Pr[X_{n-1} = a] \left(1 - \frac{1}{2^a}\right) + \Pr[X_{n-1} = a - 1] \frac{1}{2^{a-1}} \quad (1)$$

By (1),

$$\begin{aligned} \mathbf{E}[2^{X_n}] &= \sum_{a=0}^n \Pr[X_n = a] \cdot 2^a \\ &= \sum_{a=0}^n \left(\Pr[X_{n-1} = a] \cdot (2^a - 1) + \Pr[X_{n-1} = a - 1] \cdot 2 \right). \end{aligned}$$

Note that $\Pr[X_{n-1} = n] = 0$ and $\Pr[X_{n-1} = -1] = 0$. Then we have

$$\mathbf{E}[2^{X_n}] = \mathbf{E}[2^{X_{n-1}}] + 1. \quad (2)$$

Before the algorithm starts to read data, $c = 0$. So when the first data item comes, c is incremented with probability 1. Therefore $X_1 = 1$ with probability 1. Then solving the recursion (2), we get

$$\mathbf{E}[2^{X_n}] = n + 1, \quad (3)$$

i.e., the expected value of the output of the above algorithm is $F_1 + 1$, which is very close to F_1 .

Now, we calculate the variance of 2^{X_n} . First note that

$$\begin{aligned}
\mathbf{E}[2^{2^{X_n}}] &= \sum_{a=0}^n \Pr[X_n = a] \cdot 2^{2^a} \\
&= \sum_{a=0}^n \Pr[X_{n-1} = a] \left(1 - \frac{1}{2^a}\right) 2^{2^a} + \sum_{a=0}^n \Pr[X_{n-1} = a-1] \frac{1}{2^{a-1}} 2^{2^a} \\
&= \sum_{a=0}^n \Pr[X_{n-1} = a] \left(1 - \frac{1}{2^a}\right) 2^{2^a} + \sum_{a=0}^n \Pr[X_{n-1} = a-1] 2^{a+1} \\
&= \sum_{a=0}^n \Pr[X_{n-1} = a] 2^{2^a} - \sum_{a=0}^n \Pr[X_{n-1} = a] 2^a + \sum_{a=0}^n \Pr[X_{n-1} = a-1] 2^{a+1+2} \\
&= \mathbf{E}[2^{2^{X_{n-1}}}] + 3\mathbf{E}[2^{X_{n-1}}] \\
&= \mathbf{E}[2^{2^{X_{n-1}}}] + 3n.
\end{aligned}$$

Solving the above recursion, we have that $\mathbf{E}[2^{2^{X_n}}] \leq 3n^2/2$. Then,

$$\begin{aligned}
\mathbf{Var}[2^{X_n}] &= \mathbf{E}[2^{2^{X_n}}] - \mathbf{E}^2[2^{X_n}] \\
&\leq 3n^2/2 - (n+1)^2 \\
&< n^2/2.
\end{aligned}$$

Now consider the following algorithm.

Let $c_1 = \lceil \frac{3}{2\epsilon^2} \rceil$
repeat $\mathcal{A}_{1,1}$ c_1 times independently in parallel
each time, let Y_i be the output of $\mathcal{A}_{1,1}$
output $Y = \frac{1}{c_1} \sum_{i=1}^{c_1} Y_i$

Algorithm $\mathcal{A}_{1,2}$

Note that Y_i 's and Y are random variables and Y_i 's are independent and identically distributed. $\mathbf{E}[Y_i] = n+1$ and $\mathbf{Var}[Y_i] = \mathbf{Var}[2^{X_n}] < n^2/2$. $\mathbf{E}[Y] = n+1$ and $\mathbf{Var}[Y] = \mathbf{Var}[Y_i]/c_1 < \frac{n^2}{2c_1}$ by Theorem 1.

Then by the Theorem 3,

$$\Pr[|Y - (n+1)| \geq \epsilon n] \leq \frac{n^2/2}{c_1 \epsilon^2 n^2} \leq \frac{1}{3},$$

i.e., $\mathcal{A}_{1,2}$ is an $(\epsilon, \frac{1}{3})$ -approximation of F_1 .

Now, consider the following algorithm.

let $c_2 = \lceil 243 \ln \frac{2}{\delta} \rceil + 1$
repeat $\mathcal{A}_{1,2}$ c_2 times independently in parallel
each time, let Z_i be the output of $\mathcal{A}_{1,2}$
output $Z = \text{median of } Z_i\text{'s}$

Algorithm $\mathcal{A}_{1,3}$

By Theorem 5, $\mathcal{A}_{1,3}$ is an (ϵ, δ) -approximation of F_1 .

The total number of parallel copies of $\mathcal{A}_{1,1}$ is

$$c_1 \cdot c_2 = \left\lceil \frac{3}{2\epsilon^2} \right\rceil \cdot \left(\left\lceil 243 \ln \frac{2}{\delta} \right\rceil + 1 \right)$$

and each copy requires $O(\log \log(n))$ memory. So the total memory required for $\mathcal{A}_{1,3}$ is

$$O\left(\frac{1}{\epsilon^2} \log \frac{1}{\delta} \log \log n\right),$$

where n is the size of the stream.

2 Second Frequency Moment — F_2

Randomly pick t -bit ± 1 vector X_1, X_2, \dots, X_t
such that $\Pr[X_i = 1] = \Pr[X_i = -1] = \frac{1}{2}$
counter = 0
If the current item is i then
counter = counter + X_i
output counter².

Algorithm $\mathcal{A}_{2,1}$

The value of the counter when the algorithm $\mathcal{A}_{1,1}$ terminates is

$$\text{counter} = \sum X_i m_i.$$

Note that counter is a random variable as X_i 's are. Let

$$Y = \left(\sum X_i m_i \right)^2.$$

Claim. $\mathbf{E}[Y] = F_2$.

Proof.

$$\begin{aligned} \mathbf{E}[Y] &= \mathbf{E}\left(\sum X_i m_i\right)^2 \\ &= \mathbf{E}\left[\left(\sum X_i^2 m_i\right)^2\right] + \sum_{i < j} \mathbf{E}[2X_i X_j m_i m_j] \\ &= \sum m_i^2 \mathbf{E}[X_i^2] + 2 \sum_{i < j} m_i m_j \mathbf{E}[X_i X_j] && \text{linearity of expectation} \\ &= \sum m_i^2. \end{aligned}$$

End of proof of Claim. □

Remark. The above claim is true for X_i 's that are pairwise independent.
We want to bound

$$\Pr[|Y - F_2| \geq \epsilon F_2] < \frac{\mathbf{Var}[Y]}{\epsilon^2 F_2^2}$$

The variance of Y is

$$\mathbf{Var}[Y] = (\mathbf{E}[Y^2] - (\mathbf{E}[Y])^2).$$

The expectation of Y^2 is

$$\begin{aligned} \mathbf{E}[Y^2] &= \mathbf{E}\left(\sum X_i m_i\right)^4 \\ &= \mathbf{E}\left[\sum_i X_i^4 m_i^4 + 12 \sum_{i,j,k} X_i^2 m_i^2 X_j m_j X_k m_k + 6 \sum_{i<j} X_i^2 X_j^2 m_i^2 m_j^2\right] \\ &= \sum_i m_i^4 \mathbf{E}[X_i^4] + 6 \sum_{i<j} m_i^2 m_j^2 \mathbf{E}[X_i^2 X_j^2] \\ &= F_4 + 6 \sum_{i<j} m_i^2 m_j^2. \end{aligned}$$

Remark. The above proof is valid for X_i 's that are 4-wise independent.
So the variance of Y is

$$\mathbf{Var}[Y] = F_4 + 6 \sum_{i<j} m_i^2 m_j^2 - F_2^2 \leq F_2^2.$$

Let $c_1 = \lceil \frac{3}{\epsilon^2} \rceil$. Now, by Theorem 3, if we repeat $\mathcal{A}_{2,1}$ (pair-wise) independently for c_1 times to get outputs Y_1, Y_2, \dots, Y_{c_1} and output $\bar{Y} = \frac{1}{c_1} \sum_{i=1}^{c_1} Y_i$, then we have

$$\Pr[|\bar{Y} - F_2| \geq \epsilon F_2] < \frac{F_2^2}{c_1 \epsilon^2 F_2^2} < \frac{1}{3},$$

since $\mathbf{E}[\bar{Y}] = \mathbf{E}[Y_i] = F_2$ and $\mathbf{Var}[Y_i] \leq F_2^2$. So the following algorithm ($\mathcal{A}_{2,2}$) is an $(\epsilon, \frac{1}{3})$ -approximation of F_2 .

Let $c_1 = \lceil \frac{3}{\epsilon^2} \rceil$
repeat $\mathcal{A}_{2,1}$ c_1 times independently in parallel
each time, let Y_i be the output of $\mathcal{A}_{2,1}$
output $Y = \frac{1}{c_1} \sum_{i=1}^{c_1} Y_i$

Algorithm $\mathcal{A}_{2,2}$

Now, by Theorem 5, if we repeat an $(\epsilon, \frac{1}{3})$ -approximation $c_2 = \lceil 243 \ln \frac{2}{\delta} \rceil$ times independently and take the median, we get an (ϵ, δ) -approximation, which gives us the following algorithm.

let $c_2 = \lceil 243 \ln \frac{2}{\delta} \rceil$
repeat $\mathcal{A}_{2,2}$ c_2 times independently in parallel
each time, let Z_i be the output of $\mathcal{A}_{2,2}$
output $Z = \text{median of } Z_i\text{'s}$

Algorithm $\mathcal{A}_{2,3}$

Note that so far, if we count only the memory requirement for the counter operation with disregard the memory required for random bits, then the required memory size is $O(\log n)$, since the absolute value of the counter is bounded by n , where n is the size of the stream.

In the following, we discuss how to use a small number of random bits to implement this algorithm. Since all the random bits are generated at the beginning of the algorithm and used during the entire run of the algorithm, it takes memory to save the random bits.

As we remarked earlier in the discussion, the algorithm \in, ∞ has the aforementioned properties if X_i 's are pairwise independent, so complete independence is unnecessary. This gives us the possibility of using random hash functions in place of truly random bits.

Definition. Let H be a family of functions in $\Sigma^l \rightarrow \Sigma^k$. H is 4-universal if for all distinct $a_1, a_2, a_3, a_4 \in \Sigma^l$ and for all $u_1, u_2, u_3, u_4 \in \Sigma^k$,

$$\Pr_{h \in H}[(\forall i \in [1, 2, 3, 4])h(a_i) = u_i] = \frac{1}{|\Sigma^{4k}|}.$$

Let $l = \lceil \log t \rceil$ and let $k = 1$. Let

$$H = \{h_{abcd} \mid a, b, c, d \in \Sigma^l\},$$

where

$$h_{abcd}(x) = ax^3 + bx^2 + cx + d \pmod 2$$

over finite field $GF(2^{\lceil \log t \rceil})$.

Note that H is a 4-universal family of hash functions.

Let $X_i = h_{abcd}(i)$ for $i \in \{1, 2, \dots, t\}$.

When $a, b, c, d \in \Sigma^{\lceil \log t \rceil}$ are picked uniformly at random. The random variables X_i 's are 4-wise independent.

Note that a, b, c, d can be stored in $O(\log t)$ space and h_{abcd} can be computed on demand in $O(\log t)$ space.

So the total space requirement for $\mathcal{A}_{2,1}$ is $O(\log t + \log n)$. Since c_1, c_2 are both constants, the space requirement for $\mathcal{A}_{2,3}$ is also $O(\log t + \log n)$.

3 The number of distinct elements — F_0

Definition. Let H be a family of functions in $\Sigma^l \rightarrow \Sigma^k$. H is 2-universal if for all distinct $a, b \in \Sigma^l$ and for all $u, v \in \Sigma^k$,

$$\Pr_{h \in H}[h(a) = u \text{ and } h(b) = v] = \frac{1}{|\Sigma^{2k}|}.$$

Let

$$H = \{h_{ab} \mid a, b \in \Sigma^{\lceil \log t \rceil}\},$$

where

$$h_{ab}(x) = ax + b$$

over finite field $GF(2^{\lceil \log t \rceil})$.

Note that H is a 2-universal family of hash functions.

Define

$$T_l(i) = \begin{cases} 1 & \text{if } 1^l \sqsubseteq ai + b \\ 0 & \text{otherwise} \end{cases}$$

for $i \in \Sigma^{\lceil \log t \rceil}$.

Note that

$$T_l(i) = 0 \implies T_{l+1}(i) = 0. \tag{4}$$

If $a, b \in \Sigma^{\lceil \log t \rceil}$ are picked randomly,

$$\Pr[1^l \sqsubseteq ai + b] = 2^{-l}.$$

So for any fixed l

$$\Pr[T_l(i) = 1] = 2^{-l}.$$

```

Randomly pick  $a, b \in \Sigma^{\lceil \log t \rceil}$ 
// note that  $T_l(i)$ 's are implicitly defined
let  $l = 0$ 
let  $S = \emptyset$ 
when " $i$ " comes from the stream
  if  $T_l(i) = 1$  then  $S = S \cup \{i\}$ 
  if  $|S| > \alpha$  then
     $l = l + 1$ 
    replace  $S$  by sampling from  $S$  using vector  $T_l$ 
  end if
end when
output  $|S| \cdot 2^l$ 

```

Algorithm $\mathcal{A}_{3,1}$

We omit the proof of correctness.