Applied Algorithms

CSCI-B505 / INFO-I500

Lecture 24.

Algorithms on Streaming Data-II

M. Oguzhan Kulekci

- Count-min Sketch
- Half-decent estimators and frequency moments

Heavy-Hitter Detection in Distributed Environment

- Misra-Gries algorithm is an elegant solution to detect heavy hitters.
- Assume a scenario where we monitor different sources in a distributed environment.
- It is difficult to sum up Misra-Gries running on distinct points (in its original setting.
- Here is another **probabilistic data structure** that we can use to detect the HH and works well in distributed algorithms.

The Count-Min Sketch

- We aim frequency estimation, detecting the items that appear within a certain frequency. We need to achieve it in small space. Otherwise, it would be trivial by maintaining the frequency vector.

- Errors are unavoidable, but should be bounded with **provable** guarantees.

- **Sketch** is like **mapping** input stream data that arrives from a large alphabet to a target bounded space, simply via **hash** functions.

The Count-Min Sketch

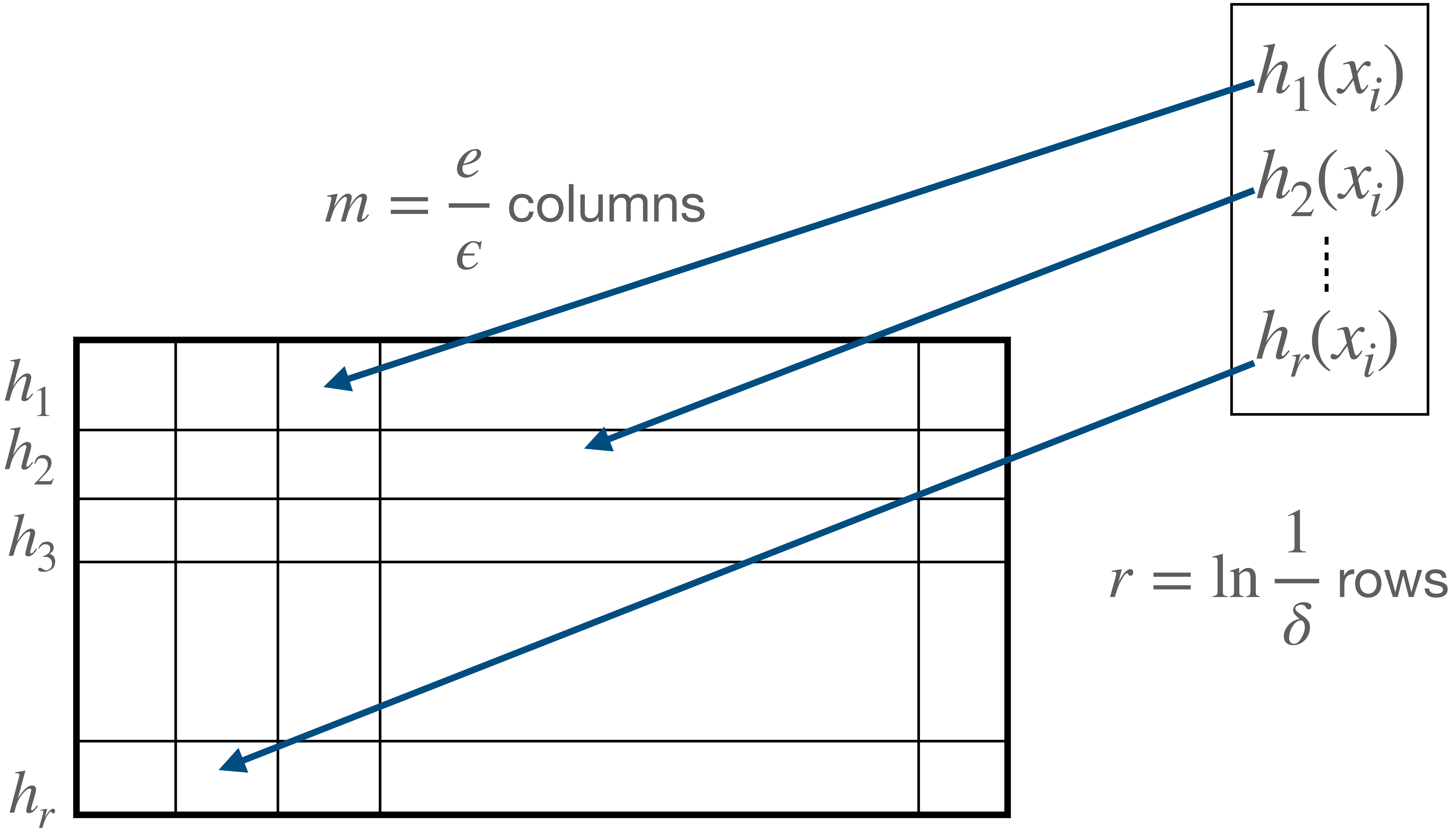
- The count-min sketch scheme
  - Returns all items that appear more than $\phi \cdot n$ times on the input stream
  - The **probability** that it will return an item that appears less than $(\phi - \epsilon) \cdot n$ times is $\leq (1 - \delta)$
  - The items with frequencies in the range $[(\phi - \epsilon) \cdot n, \phi \cdot n)$ can be reported as ϕ-HH items.
- Notice that we have **two** parameters now, ϵ **and** δ**.**

The Count-Min Sketch

r hash-functions

$h_1, h_2, \ldots, h_r$

$[n] \to [m]$



When x_i arrives, we compute its corresponding value with all hash-functions and increment the number in the corresponding cell.

The hash-functions are 2-universal, which basically means the collision probability is under well-control

The Count-Min Sketch

1, 3, 7, 3, 5

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

0	1	0	0	0
1	0	0	0	0
1	0	0	0	0

- $h_1(x) = 2x+9 \bmod 5$
- $h_2(x) = 3x+7 \bmod 5$
- $h_3(x) = 7x+3 \bmod 5$

The Count-Min Sketch

1, 3, 7, 3, 5

0	1	0	0	0
1	0	0	0	0
1	0	0	0	0

$\Rightarrow$

1	1	0	0	0
1	1	0	0	0
1	0	0	0	1

- $h_1(x) = 2x+9 \bmod 5$
- $h_2(x) = 3x+7 \bmod 5$
- $h_3(x) = 7x+3 \bmod 5$

The Count-Min Sketch

1, 3, 7, 3, 5

1	1	0	0	0
1	1	0	0	0
1	0	0	0	1

$\Rightarrow$

1	1	0	1	0
1	1	0	1	0
1	0	1	0	1

- $h_1(x) = 2x+9 \bmod 5$
- $h_2(x) = 3x+7 \bmod 5$
- $h_3(x) = 7x+3 \bmod 5$

The Count-Min Sketch

1, 3, 7, 3, 5

1	1	0	1	0
1	1	0	1	0
1	0	1	0	1

$\Rightarrow$

2	1	0	1	0
1	2	0	1	0
1	0	1	0	2

- $h_1(x) = 2x+9 \bmod 5$
- $h_2(x) = 3x+7 \bmod 5$
- $h_3(x) = 7x+3 \bmod 5$

The Count-Min Sketch

1, 3, 7, 3, 5

2	1	0	1	0
1	2	0	1	0
1	0	1	0	2

$\Rightarrow$

2	1	0	1	1
1	2	1	1	0
1	0	1	1	2

- $h_1(x) = 2x+9 \bmod 5$
- $h_2(x) = 3x+7 \bmod 5$
- $h_3(x) = 7x+3 \bmod 5$

The Count-Min Sketch

2	1	0	1	1
1	2	1	1	0
1	0	1	1	2

```
Forall i = 1..σ and j = 1..r
    Report i, if all CM[i][h_j(i)] ≥ φ · n
```

Return all 0.4-HH items: Report all $x_i \in \{1,2,\ldots,\sigma\}$ that appears $\geq 0.4n$

$$0.4 \cdot 5 = 2$$

For $x = 3$, $h_1(3) = 0,\ \ h_2(3) = 1,\ \ h_3(3) = 4,$

and $CM[0][0] = 2, CM[1][1] = 2, CM[2][4] = 2$ satisfies the condition.

So x=3 is a 0.4-HH.

The Count-Min Sketch

- For all true HH, it is clear that it will work.
- What is the probability that a false item with a frequency $\leq (\phi - \epsilon)n$ will be reported ?

f_x: frequency of x.

$\Delta_i = C[i, h_i(x)] - f_x$: the over-count of x on the ith row.

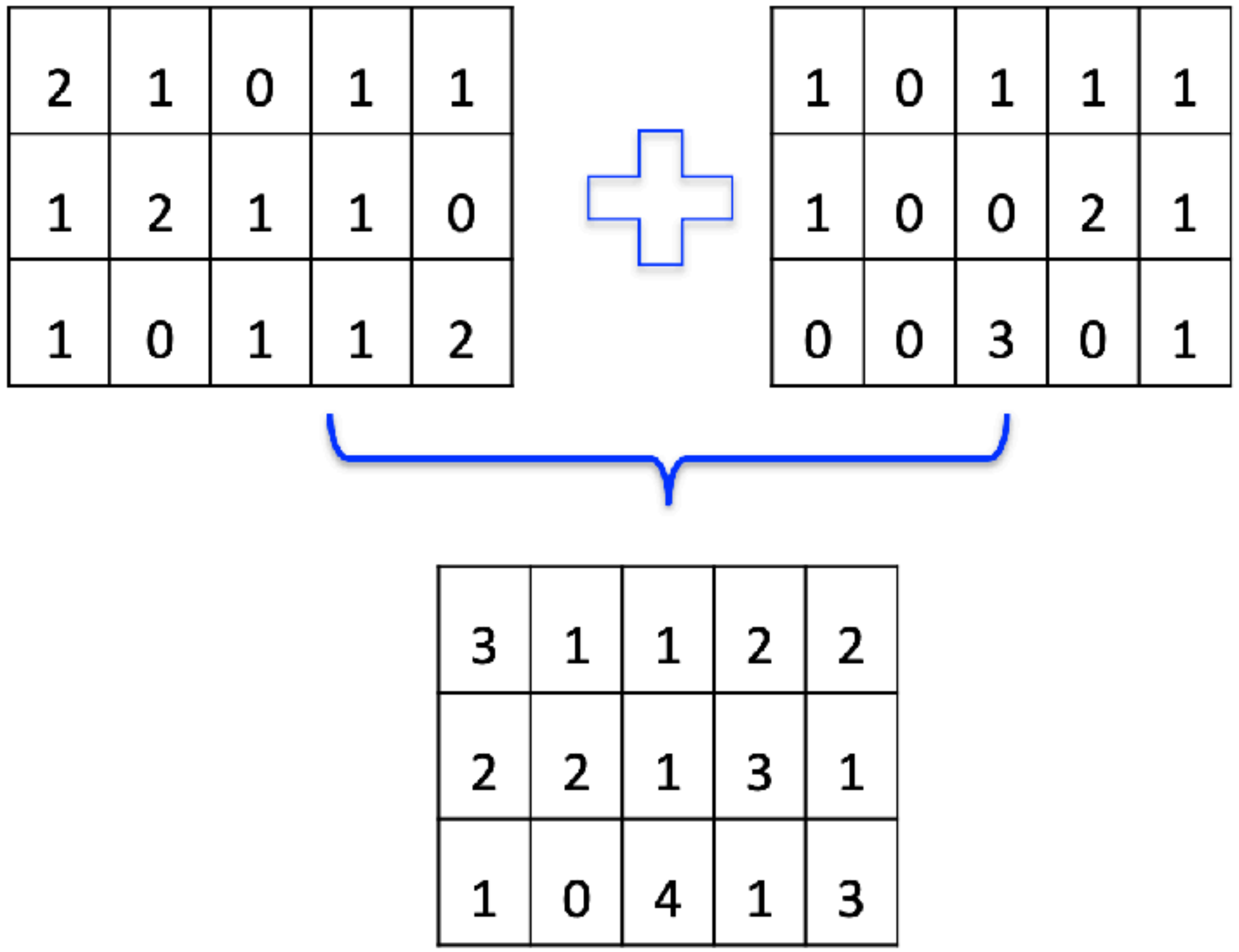
$E[\Delta_i] \leq n/m = \epsilon n/e$

$\Pr[\Delta_i > \epsilon n] < E[\Delta_i]/(\epsilon n) \leq 1/e \qquad \text{(Markov Inequality)}\ Pr(X > a) < \frac{E[x]}{a}$

$$\Pr\left[\min\{\Delta_i \mid 1 \leq i \leq r\} > \epsilon n\right] < (1/e)^r = (1/e)^{\ln(1/\delta)} = \delta$$

The Count-Min Sketch

- Space complexity is $O(\frac{1}{\epsilon} \cdot \log \frac{1}{\delta})$ due to the matrix size
- Better accuracy if we keep the parameters small, but that will increase the space as well.
- To answer the queries, yes we need to pass over the alphabet, but each verification is expected to finish very fast.
- How about the distributed computability property ?



Estimating the Skewness of a Stream

Second frequency moment of a sequence is $F_2 = \sum_{i=1}^{\sigma} f_i^2$, where

- $x_1, x_2, \ldots, x_n$ is the input stream
- $x_i \in 1,2,3,\ldots,\sigma$
- f_i is the frequency of i on the input.

$$\langle 1,2,3,4 \rangle \rightarrow F_2 = 1^2 + 1^2 + 1^2 + 1^2 = 4$$

$$\langle 1,1,3,4 \rangle \rightarrow F_2 = 2^2 + 0^2 + 1^2 + 1^2 = 6$$

$$\langle 1,1,3,3 \rangle \rightarrow F_2 = 2^2 + 0^2 + 2^2 + 0^2 = 8$$

$$\langle 1,1,1,4 \rangle \rightarrow F_2 = 3^2 + 0^2 + 0^2 + 1^2 = 10$$

- F_2 is a measure of skewness
- with many applications …

Estimating the Skewness of a Stream

Second frequency moment of a sequence is $F_2 = \sum_{i=1}^{\sigma} f_i^2$, where

- $x_1, x_2, \ldots, x_n$ is the input stream
- $x_i \in 1,2,3,\ldots,\sigma$
- f_i is the frequency of i on the input.

- F_2 is a measure of skewness
- with many applications …

- We seek for an estimation $\hat{F}_2$ such that $Pr(|\hat{F}_2 - F_2| > \epsilon F_2) < \delta$
- Surely, in small space

$$\langle 1,2,3,4 \rangle \rightarrow F_2 = 1^2 + 1^2 + 1^2 + 1^2 = 4$$

$$\langle 1,1,3,4 \rangle \rightarrow F_2 = 2^2 + 0^2 + 1^2 + 1^2 = 6$$

$$\langle 1,1,3,3 \rangle \rightarrow F_2 = 2^2 + 0^2 + 2^2 + 0^2 = 8$$

$$\langle 1,1,1,4 \rangle \rightarrow F_2 = 3^2 + 0^2 + 0^2 + 1^2 = 10$$

Estimating the Skewness of a Stream

The strategy is :

- Find **many** half-decent estimators for F_2
- Run them and take their **average** for a better accuracy
- Perform the above two steps many times and take the **median** as the result to keep error low.

Half-decent estimator:

- Assume the hash function

$$h : \{1,2,\ldots,\sigma\} \to \{+1,-1\}$$

```
halfDecent(h):
  Z=0
  for i=1 to n
     Z = Z + h(x_i)
  return Z^2
```

Estimating the Skewness of a Stream

$$E[Z^2] = E\left[\left(\sum_{i=1}^{\sigma} h(i) f_i\right)^2\right] = E\left[\left(\sum_{i=1}^{\sigma} h_i f_i\right)^2\right]$$

$$= \quad E\left[\sum_{i=1}^{\sigma} h_i^2 f_i^2 + \sum_{i \neq j} h_i h_j f_i f_j\right]$$

$$= \quad \sum_{i=1}^{\sigma} \underbrace{E[h_i^2]}_{1} f_i^2 + \sum_{i \neq j} \underbrace{E[h_i]E[h_j]}_{0} f_i f_j$$

$$= \quad F_2$$

```
halfDecent(h):
  Z=0
  for i=1 to n
     Z = Z + h(x_i)
  return Z^2
```

Assuming hash functions are pairwise independent and with the linearity of expectations

Estimating the Skewness of a Stream

Procedure:

- Generate $s_1 \cdot s_2$ independent half-decent hash functions for $s_1 = 15/\epsilon^2,$ and $s_2 = 4\log(1/\delta)$.
- For each packet of s_1 hash functions, take the average
- We are left with s_2 average estimates. The median of these averages is the final result.
- The result will be deviating from the actual value by more than ϵ only with a low probability of δ. (*Proof is beyond our coverage here…*)

$$Pr(|\hat{F}_2 - F_2| > \epsilon F_2) < \delta$$

Reading assignment

- https://www.cs.dartmouth.edu/~ac/Teach/data-streams-lecnotes.pdf
- Check internet resources to learn more about the count-min sketch, half-decent estimators, frequency moment, …