Lecture 2: Sketching

Streaming algorithms: Setting where data appears sequentially, and the goal is to process this in an online fashion, using few resources
↳ usually space.

Problem 1: Given a stream of numbers, track majority if it exists.

n numbers in stream

$$x_1, \dots, x_n \longrightarrow \log n \text{ bit \#s.}$$

$$m = \text{maj}(x_1, \dots, x_n) \text{ if } \# \{ i : x_i = m \} > n/2.$$

naive: $n \log n$ bits of memory.

Claim: Can do with $O(\log n)$!

Algo:

```
count = 0, guess = NULL.
For x_i in stream:
    If count = 0,
        current = x_i, count = 1.
    else if x_i = guess
        count ++
    else
        count --
output guess
```

2 1 4 2 4 4 1 4 2 2 2 2 2 2

Question: how to analyze?

Hint: consider signed counter

Heavy hitters problem: Given $x_1, \cdots, x_n$

say that y is an HH if

$\#\{i : x_i = y\} \geq n/k \leftarrow$ some parameter $k = n/2, \cdots 1$

Hard for even reasonable k!

Problem: Output the most common element of the stream (the "heaviest hitter").

Claim: Requires $\Omega(n \log n)$ space!

$x_i \in \{1, \cdots, n^2\}$

OO $\underbrace{x_1, x_2 \cdots x_n}$ | $x \in S$? $x \notin S$

$\rightarrow \binom{n^2}{n}$ such sequences.

If I had $\leq B$ bits of memory, $\leq 2^B$ distinct states. But each distinct state has different answer.

useful approx: when $k \ll n$, $\binom{n}{k} \approx n^k$

$$2^B \geq \binom{n^2}{n} \approx n^{2n}$$

$$B \geq n \log n.$$

For $k \leq n/2$, do same except repeat last element many times!

(Breaks only at majority).

$\Omega(n \log n)$ for all "reasonable" k.

Relax : ε - HH. params k, ε.

Given stream $x_1, \dots, x_n$:

1). If x occurs $\geq n/k$ times, output it

2). If we output x, then it occurs $\geq n/k - \varepsilon n$ times.

Space: $O(1/\varepsilon)$

e.g. If $\varepsilon = 1/2k$: output all $x \geq n/k$

any output $x \geq \frac{n}{2k}$.

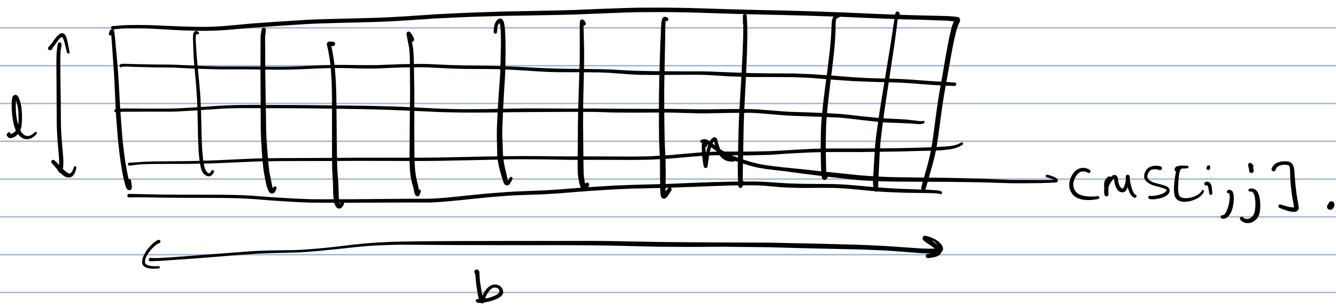
using $O(k)$ memory

<u>Tool:</u> Count-Min Sketch

Implements 2 operations: $\mathrm{Inc}(x)$, $\mathrm{Count}(x)$.

Params b = # buckets $(b \approx 1/\varepsilon)$

ℓ = # hash functions. $(\ell \approx O(1))$



$h_1, \dots, h_\ell : U \to \{0, \dots, b-1\}$.

are your "nice" hash functions.

$\mathrm{Inc}(x)$: $\mathrm{CMS}[i, h_i(x)]{+}{+} \quad \forall i = 1, \dots, \ell$.

$\mathrm{Count}(x)$: $\min_i \mathrm{CMS}[i, h_i(x)]$

why does this work? Let x occur c_x times in stream.

Know: $\mathrm{Count}(x) \geq c$ (why?).

need to bound <u>overestimation</u>

Let

$$Z_i = \mathrm{CMS}[i, h_i(x)] = c_x + \sum_{\substack{y \neq x \\ h_i(y) = h_i(x)}} c_y = (*)$$

$\forall x \neq y, \quad \Pr[h_i(y) = h_i(x)] = 1/b.$

$$\Rightarrow \quad \mathbb{E}[(*)] = \sum_{y \neq x} \mathbb{E}\left[\mathbb{1}[h_i(y) = h_i(x)]\right] = \frac{n - C_x}{b} \leq \frac{n}{b}.$$

Set $\quad b = 2/\varepsilon.$

$$\Rightarrow \quad \leq \frac{\varepsilon n}{2}$$

Markov's Inequality: $Y \geq 0$ is r.v.,

$$P[Y \geq \alpha] \leq \frac{\mathbb{E}[Y]}{\alpha}.$$

$$\Rightarrow \Pr[(*) \geq \varepsilon n] \leq \frac{\varepsilon n/2}{\varepsilon n} \leq 1/2.$$

$$\Rightarrow \quad \Pr[Z_i - C_x \geq \varepsilon n] \leq 1/2.$$

$$\Pr[\min(Z_1, \cdots, Z_\ell) \geq C_x + \varepsilon n] \leq \left(\tfrac{1}{2}\right)^\ell$$

$\ell = \log_2 1/\delta. \longrightarrow \leq \delta.$

$\Rightarrow$ For $\mathrm{count}(x)$ to be accurate to εn w.p. $1-\delta$,

need to set $\quad b = 2/\varepsilon, \quad \ell = \log 1/\delta$

space $= \quad O\left(\frac{1}{\varepsilon} \log 1/\delta\right)$ words of memory.