DS - 563 / CS - 543 Lecture 2 / 2023-01-24

Today

- Wrap up Count/Min Sketch
- Heavy hitters
- Second moment estimation via AMS sketch

Count/Min Sketch Recap

Goal:

- items arrive in arbitrary order (from some set X)
- multiple copies allowed
- provide estimates "what fraction of items is $y \in X$?"
- use small space

Last time:

- $h: X \to \{1, \dots, k\} = [k]$ random hash function
- keep array $A[1..k]$
- initially $A[i] = 0$ for all $i \in [k]$
- insertion of element $x \in X$:
  $A[h(x)] \leftarrow A[h(x)] + 1$
- estimate for $y \in X$:

$$g(y) = \frac{A[h(y)]}{\sum_{i=1}^{k} A[i]}$$

Properties:

$f(x)$ = exact number of occurrences of x

$$S = \sum_{i=1}^{k} A[i] = \sum_{x \in X} f(x)$$

Always: $\frac{f(y)}{S} \leq g(y)$

With probability $1/2$: $g(y) \leq \frac{f(y)}{S} + \frac{2}{k}$

To get additive ε approximation, set $k = \lceil 2/\varepsilon \rceil$

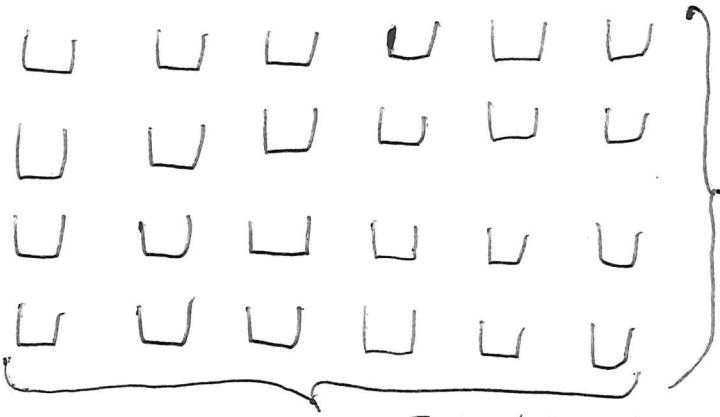
How to make probability of error at most $\delta \in (0, 1/2)$?

- Run $t = \lceil \log(1/\delta) \rceil$ independent copies
- Query y: return the <u>minimum</u> of estimates from all copies

$$\Pr[\text{all wrong}] \leq \left(\frac{1}{2}\right)^t \leq \delta$$

$\Pr[\text{all wrong}]$ = overestimate by more than ε

Visually:



$k = \lceil 2/\varepsilon \rceil$ buckets each

Total space: $O\left(\frac{1}{\varepsilon} \log(1/\delta)\right)$

What is missing?

How do we store random hash functions?

- We can't (lots of space!)
- Pairwise independence suffices

  For $x \neq y$: $\mathbb{E}[C_{x,y}] = \Pr[h(x) = h(y)] \leq \frac{1}{k}$

- Or even

  $\mathbb{E}[C_{x,y}] \leq \frac{O(1)}{k}$ ← fixed constant

  (just increase k by a constant factor)

- discussion section: sample construction for strings

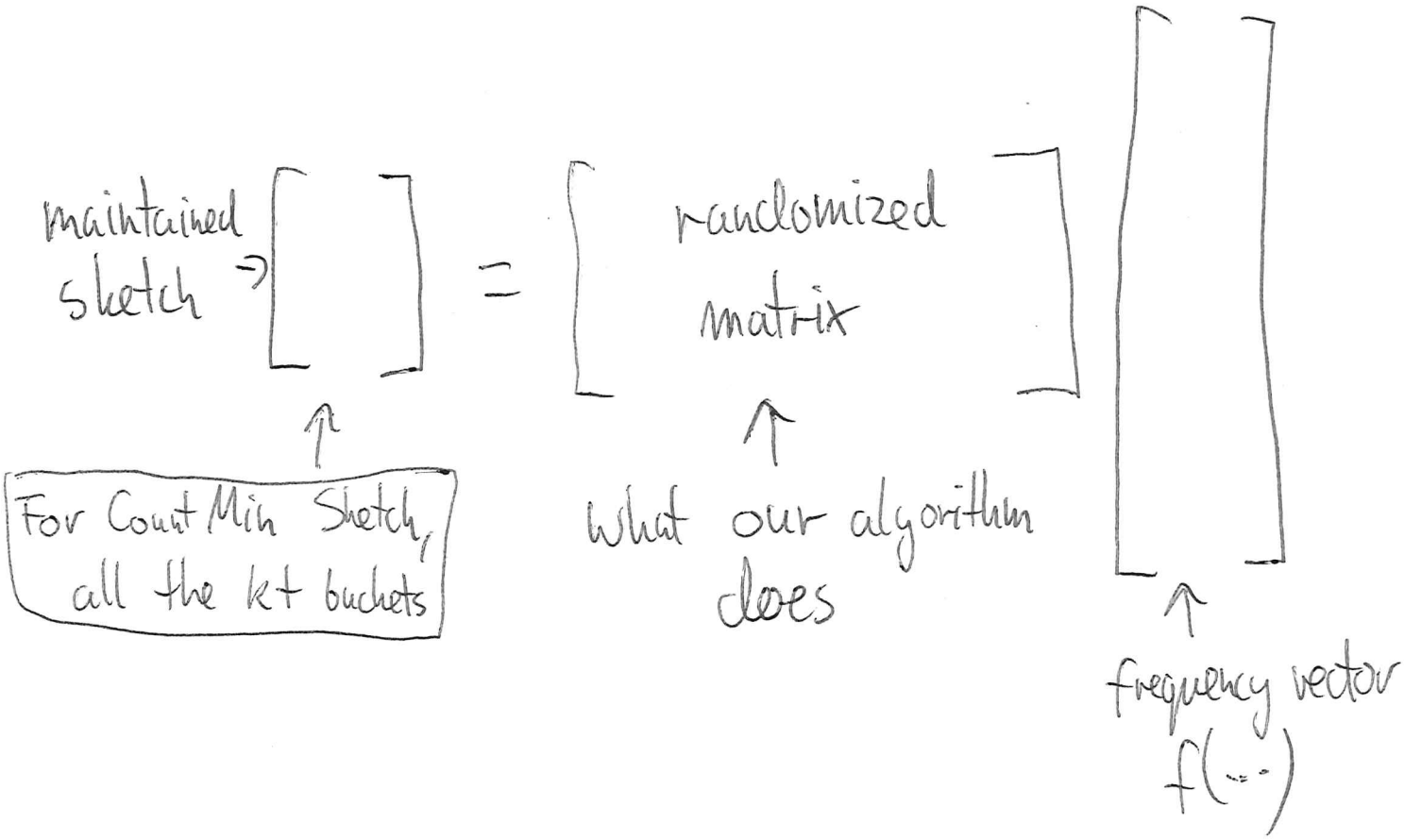
Nice properties of CountMin Sketch:

- can handle deletions (decrease corresponding counters)
- can be computed separately for subsets and easily combined (use the same hash functions!)

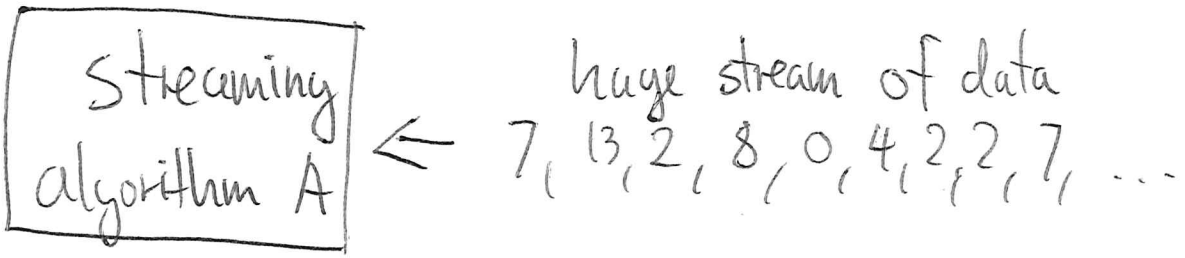
Application: different data centers handling different parts of the data set

Important concept: Linear sketches

Count Min Sketch is an example



Important concept: streaming algorithms



- A reads and processes items one by one
- A should use much less space than the input stream size

Heavy hitters: list frequent elements in the stream

Setting: same as for CountMin Sketch

X - universe from which elements of the stream come from

$f(x)$ - number of occurrences of $x \in X$

S - total number of items

Our task: For some $\varepsilon \in (0,1)$, (ε: parameter)

return $H \subseteq X$ s.t.

$$\forall x \in X: \quad f(x) \geq 2\varepsilon \cdot S \Rightarrow x \in H$$
$$f(x) \leq \varepsilon S \Rightarrow x \notin H$$

Approach:

- find candidates $H' \subseteq X$
- use Count Min Sketch to verify:
  output all $x \in H'$ for which
  (MS says "fraction of $x \geq 2\varepsilon$"

How to find <u>small</u> H'?

TO BE CONTINUED