Heavy Hitters
Tail Bounds

Anna Karlin

Stream model

- Input elements (e.g. Google queries) enter/arrive one at a time.
- We cannot possibly store the stream.

Question: How do we make critical calculations about the data stream using a limited amount of memory?

Sources of this kind of data

- Sensor data
  - E.g. millions of temperature sensors deployed in the ocean
- Image data from satellites or surveillance camers
  - E.g. London
- Internet and web traffic
  - E.g. millions of streams of IP packets
- Web data
  - E.g. Search queries on Google, clicks on Bing, etc.

Example Applications

- Mining query streams
  - Google wants to know which queries are more frequent today than yesterday.
- Mining click streams
  - Facebook wants to know which of its ads are getting an unusual number of hits in the last hour.
- Mining social network news feeds
  - E.g., looking for trending topics on Twitter and Facebook, trending videos on TikTok

More Applications

- Sensor networks
  - Many sensors feeding into a central controller.
- IP packets
  - Gather congestion information for optimal routing
  - Detect denial-of-service attacks

Problem

- Input: sequence of n elements $x_1, x_2, \ldots, x_n$ from a known universe U (e.g., 8-byte integers).
- Goal: perform a computation on the input, in a single left to right pass where
  - Elements processed in real time
  - Can't store the full data. => minimal storage requirement to maintain working "summary"

Heavy Hitters: Keys that occur many times

$n = 11$

$\lceil \frac{n}{4} \rceil = 3.$

32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4,

32, 12

$f_{32}^{6} = 2$

$x_1, x_2, \ldots, x_n$

Applications:

- Determining popular products
- Computing frequent search queries
- Identifying heavy TCP

find all elts that occur more than $\frac{n}{100}$

$\forall \quad t \leq n$

f_x^t : # times element x has appeared in $x_1, x_2, \ldots, x_t$

Goal: Find/output all elements with $f_x^n > \frac{n}{k}$

(These elts are "heavy hitters") $(k = 100)$

Output has size $O(k)$

$\boxed{\text{Provably impossible to solve this problem exactly with sublinear space}}$

How many elts can there be with $f_x^+ \geq \frac{n}{100}$?

a) ∞
b) n
$\boxed{\text{c) } 100}$
d) 10

Modified goal: (solve ε-HH problem)

① If $f_x^n \geq \frac{n}{k}$, added to list we output.

② If some elt, say y, added to list, then w.p. $\geq 1-\delta$

$$f_y^n \geq \frac{n}{k} - \varepsilon n$$

Suppose $k = 20$ $\underline{\varepsilon = 0.01}$ $\delta = \underline{\frac{1}{2^{10}}}$

① if $f_x^n \geq 0.05n$ times, then in our list

② for every y in list w.p. $\geq 1 - \frac{1}{2^{10}}$, $f_y^n \geq 0.05n - 0.01n$

$=0.04n$

$k, \varepsilon, \delta.$ $\Longrightarrow$ determine

$b, \ell.$

Count-min sketch

- Maintain a short summary of the information that still enables answering queries.

- Cousin of the Bloom filter
    - Bloom Filter solves the “membership problem”.
    - We want to extend it to solve a counting problem.

COUNT-MIN SKETCH

Want:

① If $f_x^n \geq \frac{n}{k}$, added to list we output. HH ✓

② If some elt, say y, added to list, then w.p. $\geq 1-\delta$

$$f_y^n \geq \frac{n}{k} - \varepsilon n$$

designer specifies $k, \varepsilon, \delta \Rightarrow b, l$

keep 2D array

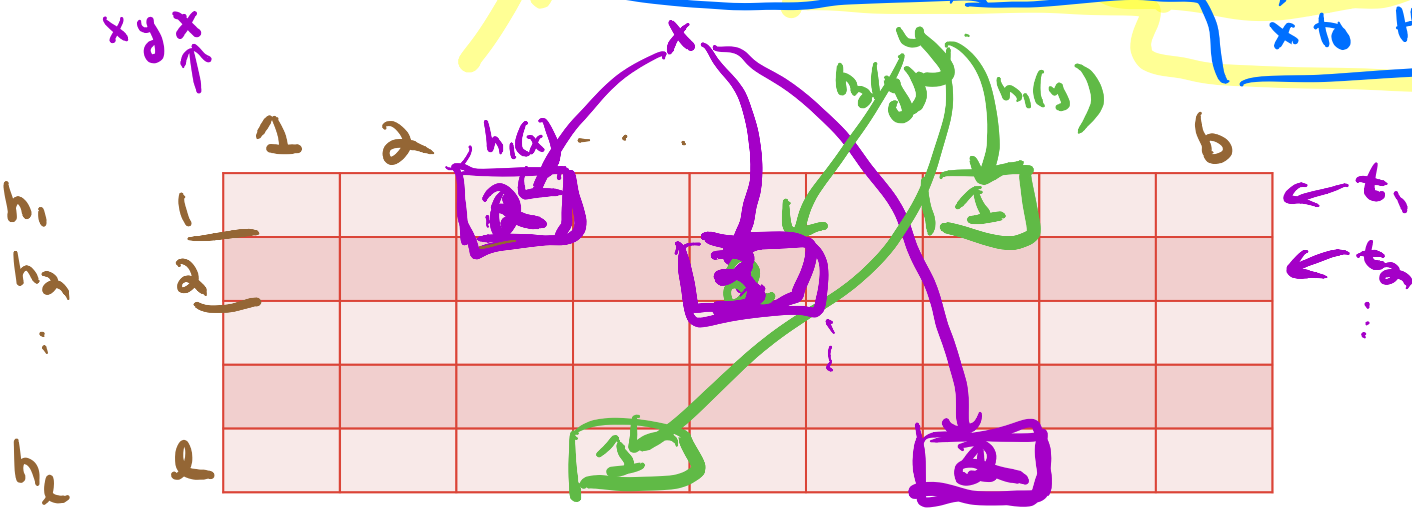
l hash tables, each of size b.

when elt x shows up.

Update (x): $\forall \; 1 \le j \le \ell$ increment $t_j[h_j(x)]$

Count (x): return $\min_{1 \le j \le \ell} t_j[h_j(x)]$

if Count$(x) \ge \frac{n}{k}$, add x to HH list



after $x_1 \dots x_t$ arrived
$t_j[h_j(x)]$ guaranteed

a) $\le f_x^t$
b) $= f_x^t$
c) $\ge f_x^t$
d) I don't know

after $x_1 \dots x_t$ arrived
Count(x) guaranteed

a) $\le f_x^t$
b) $= f_x^t$
c) $\ge f_x^t$
d) I don't know

Assumptions

① hash functions behave like random maps

$$h_1, \ldots, h_\ell : U \longrightarrow \{0, 1, \ldots, b-1\}$$

$$\forall x \neq y \quad \Pr(h_j(x) = h_j(y)) = \frac{1}{b}$$

$$\boxed{b \cdot \frac{1}{b^2} = \frac{1}{b}}$$

② hash fns $h_1, \ldots, h_\ell$ are indep of each other.

If hash fn h random, $x \neq y$

$\Pr(h(x) = h(y)) =$

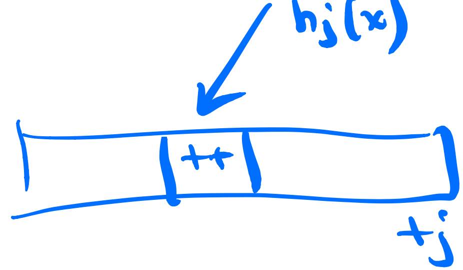
a) $\boxed{\frac{1}{b}}$

b) $\frac{1}{b^2}$

c) 0

d) I don't know

Fix time t. ($x_1, \dots, x_t$ have just arrived)



$$Z_j^t \triangleq \underbrace{t_j[h_j(x)]}_{\geq\ f_x^t}$$

(# of times x occurs in $x_1 \dots x_t$)

$$\boxed{Z_j^t = f_x^t + \sum_{\substack{\text{distinct elts } y \neq x \\ y \in \{x_1, \dots, x_t\}}} f_y^t \, W_{xy}}$$

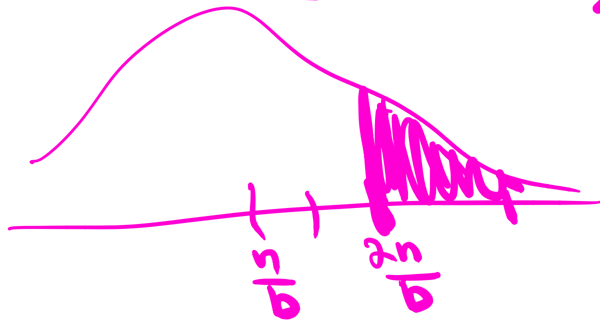
$$W_{xy} = \begin{cases} 1 & \text{if } h_j(x) = h_j(y) \\ 0 & \text{o.w.} \end{cases}$$

$$E(Z_j) = f_x^t + \underbrace{\boxed{\sum_{\substack{\text{distinct } y \\ y \neq x}} f_y^t}}_{\leq\ t - f_x^t\ \leq\ t\ \leq\ n} \underbrace{E(W_{xy})}_{\frac{1}{b}}$$

$$E(z_j) \le f_x^t + \boxed{\frac{n}{b}} \equiv E(\underbrace{z_j - f_x^t}_{X}) \le \frac{n}{b}$$

$$\underbrace{\Pr(\underbrace{z_j - f_x^t}_{} \ge \frac{2n}{b})}_{\frac{n}{b}} \qquad \Pr\left(X \ge \frac{2n}{b}\right)$$

$$\le \Pr(\underbrace{X \ge 2E(X)}_{X \ge 0})$$



Markov's Inequality.

$$\frac{\text{If } X \ge 0 \text{ r.v.}}{\Pr(X \ge cE(X)) \le \frac{1}{c}}$$

Count-Min Sketch

- Elegant small space data structure.
- Space used is independent of n.
- Is implemented in several real systems.
  - AT&T used in network switches to analyze network traffic.
  - Google uses a version on top of Map Reduce parallel processing infrastructure and in log analysis.
- Huge literature on sketching and streaming algorithms (algorithms like Distinct Elements, Heavy Hitters and many many other very cool algorithms).