Lecture 11 10/1/2025

Count-min and Count Sketches

In terms of frequency moments F_∞ is the most frequently occurring item in the stream.

It is a brittle measure. In most application we want to know the "heavy hitters", items that occur very frequently.

From a theoretical perspective we

from a ... frequency ...
will call an index $i \in [n]$ a
heavy hitter if $\bar{f}_i \geq \alpha F_1 = \alpha m$
for some sufficiently large constant $\alpha \in (0,1)$

Alternatively $\bar{f}_i \geq \frac{m}{k}$ for some
integer k.

A classical algorithm shows that
one can identify items i with
$\bar{f}_i \geq \frac{m}{k}$.

Misra-Gries(k)

... data structure D

- we have a data structure D that stores k items along with a counter for each. D is initialized to empty set

- $m \leftarrow 0$
- while (stream is not empty) do

    $m \leftarrow m+1$

    e_m is current item

    If $e_m \in D$ then

        increment counter for e_m in D

    Else

        If D has $< k$ elements

            add e_m to D with counter value 1

        Else

            decrease counter value by

1 for all current elements

delete from D any element
with counter set to 0

- end while

- Output values stored in D
and the counters values.

Implicitly it defines an estimate
$\tilde{f}_i$ for each i
if $i \in D$ at the end then
$\tilde{f}_i$ is the counter value
otherwise it is 0.

Theorem: $\forall i \in [n]$

$$\hat{f}_i - \frac{m}{k+1} \le \hat{f}_i \le f_i$$

Hence if i is a heavy hitter it will be in D. Space usage is $O(k)$.

Although Misra-Gries is nice it does not allow deletions and also does not provide a sketch.

Count-min and Count sketches are a way to use hashing to identify heavy hitters and they have led to many applications

Basic idea is simple.

Suppose we use a hash function $h : [n] \to [ck]$ for some sufficiently large constant k.

Then h spreads the n items into ck buckets. Suppose the heavy hitters are $i_1, i_2, \ldots, i_k$

Then we expect that they will

not collide and we can use
separate counts in each bucket.
We will use amplification as
usual by considering multiple
hash functions rather than a
single one.

[Cormode-Muthukrishnan]

Count-Min Sketch(w, d)
- $h_1, h_2, \dots, h_d$ are d independent
pairwise indep hash functions
from $[n] \to [w]$.

- While (stream is not empty)

  $e_t = (i_t, \Delta_t)$ is current item

  for $l = 1$ to d do

  $$C[l, h(i_t)] \leftarrow C[l, h(i_t)] + \Delta_t .$$

- end while

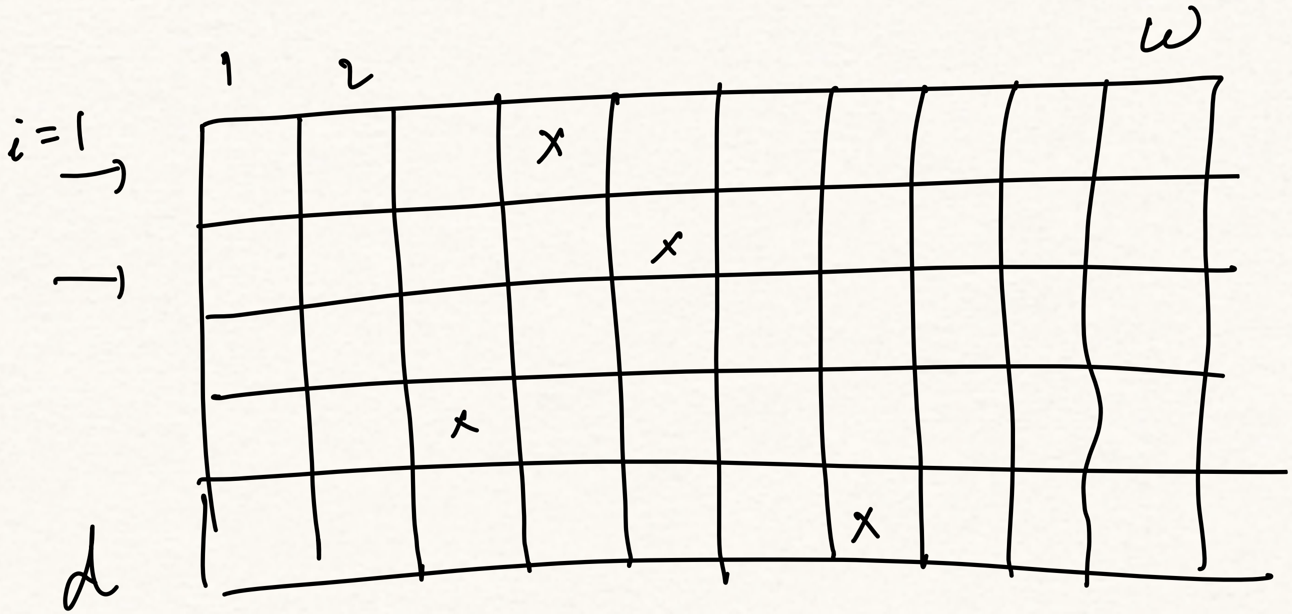
- for $i = 1$ to $[n]$

  $$\tilde{x}_i = \min_{l=1}^{d} C[l, h(i)]$$

w is width of the sketch

d is [illegible] independent hash function

d is # of independ[illegible]
we use.



$$C[\ell, s] = \sum_{i : h_\ell(i) = s} x_i$$

Lemma: Consider strict turnstile model ($\bar{x} \geq 0$). Let $d = \Omega(\ln \frac{1}{\delta})$
and $w \geq \frac{2}{[illegible]}$. Then, $\forall i \in [n]$

and $w \ldots \varepsilon$

(i) $\tilde{x}_i \geq x_i$

(ii) $\Pr\left[\tilde{x}_i \geq x_i + \varepsilon \|x\|_1\right] \leq \delta$.

Proof: Fix i.

For $l \in [d]$

$$Z_l = C[l, h_l(i)] = x_i + \sum_{i' \neq i : h_l(i') = h_l(i)} x_{i'}$$

$$Z_l - x_i = \sum_{i' \neq i : h_l(i') = h_l(i)} x_{i'}$$

$$E[Z_l - x_i] = \frac{1}{w} \sum_{i' \neq i} x_{i'} \leq \frac{\|x\|_1 - x_i}{w}.$$

By pairwise independence.

$$\mathbb{E}[Z_l - x_i] \le \frac{\varepsilon}{2}\|\bar{x}\|_1$$

By Markov

$$\Pr[Z_l - x_i \ge \varepsilon\|\bar{x}\|_1] \le \frac{1}{2}.$$

Thus

$$\Pr\left[\min_l (Z_l - x_i) \ge \varepsilon\|x\|_1\right] \le \left(\frac{1}{2}\right)^d \le \delta.$$

by independence

□.

Choosing $d = \Omega(\log n)$ we have

$$\tilde{x}_i \le x_i + \varepsilon\|x\|_1 \quad \forall i \in [n]$$

with high probability.

Count Min gives over estimates.

Total space is $O(dw)$ counters

$O(\frac{1}{\varepsilon} \log n)$.

Advantage: $\frac{1}{\varepsilon}$ dependence, simple

Disadvantage: only handles $\bar{x} \geq 0$.

Exercise: Show that Count Min is linear sketch.

a [illegible]

Count Sketch

Similar to Count Min in using d independent hash functions but uses F_2 estimation ideas and median estimator instead of min.

CountSketch(w, d)

- $h_1, h_2, \ldots, h_d$ independent hash functions from $[n] \to [w]$

indep hash functions

- $g_1, g_2, \ldots, g_d$ indep
  from $[n] \to \{-1, +1\}$.

- While (stream is not empty) do
  $e_t \leftarrow (i_t, \Delta_t)$.
  for $l = 1$ to d do

  $$C[l, h_l(i_t)] \leftarrow C[l, h_l(i_t)] + g_l(i_t)\Delta_t$$

  endfor
  end while

- for $i \in [n]$

$$\tilde{x}_i = \operatorname*{median}_{l=1}^{d} \left(g_l(i)\, C[l, h_l(i)] \right)$$

$\tilde{x}_i$ can be negative even if

$\bar{x} \geq 0$.

cancellations happen like in F_2 estimation

Lemma: Let $d \geq 4 \ln \frac{1}{\delta}$ and

$w \geq \frac{3}{\varepsilon^2}$. Then for any $i \in [n]$

(i) $E[\tilde{x}_i] = x_i$ and

(ii) $\Pr\left[|\tilde{x}_i - x_i| \geq \varepsilon \|\bar{x}\|_2\right] \leq \delta$.

Proof: Fix i. For $l \in [d]$

To make analysis easier, let

$Y_{i'}$ for $i' \neq i$ be the indicator

for $h_l(i) = h_l(i')$

$$Z_l = g_l(i)\, C[l, h_l(i)]$$

$$= g_l(i)\left[g_l(i) x_i + \sum_{i' \neq i} g_l(i') Y_{i'} x_{i'}\right]$$

$$= X_i + \sum_{i' \neq i} g_\ell(i) g_\ell(i') Y_{i'} X_{i'}$$

$E[Z_\ell] = X_i$ by pairwise indep of g_ℓ.

We note that $E[Y_{i'}] = \frac{1}{w}$

and $E[Y_{i'}^2] = \frac{1}{w}$ by pairwise indep of h_ℓ.

$$\mathrm{Var}(Z_\ell) = E[(Z_\ell - X_i)^2]$$

$$= E\left[\left(\sum_{i' \neq i} g_l(i) g_l(i') Y_{i'} x_{i'}\right)^2\right]$$

$$= \sum_{i' \neq i} x_{i'}^2 E[Y_{i'}^2]$$

$$= \frac{1}{w} \sum_{i' \neq i} x_{i'}^2 .$$

$$\leq \frac{\|\bar{x}\|_2^2}{w} . \leq \frac{\varepsilon^2}{3} \|\bar{x}\|_2^2$$

Hence using Chebyshev

$$E\left[|Z_l - x_i| \geq \varepsilon \|\bar{x}\|_2\right] \leq \frac{1}{3}.$$

Via Chernoff bounds

$$\Pr\left[\,\left|\mathrm{med}(Z_1, Z_2, \dots Z_d) - X_i\right| \geq \varepsilon \|\bar{x}\|_2\right] \leq \delta.$$

$\square$.

<u>Important</u>: Sketches do not store directly the "identity" of the heavy hitters. Given $i \in [n]$ we

Can estimate $\tilde{x}_i$ from the sketch.
But outputting all i such that $\tilde{x}_i$ is high requires a linear scan through $[n]$. Can maintain multiple data structures and use additional information to find the heavy hitters in $\tilde{O}(k)$ space and time.

Sparse Recovery

One nice and powerful application of Count Sketch is for sparse recovery. Suppose $\bar{x} \in R^n$ is sparse or close to sparse. Meaning that only k of the coordinates are non-zero. Can we recover x without knowing which of the coordinates are going to be important? Want to use only $\tilde{O}(k)$ space.

<u>Defn:</u> Given $\bar{x} \in \mathbb{R}^n$ let

$$\mathrm{err}_2^k(\bar{x}) = \min_{\bar{z} : \|z\|_0 \leq k} \|\bar{x} - \bar{z}\|_2 .$$

That is, what is the best k-sparse approximation to $\bar{x}$.

Offline, easy to compute.

$z_i^* = \bar{x}_i$ if i is among the largest absolute value k coordinates of $\bar{x}$

$= 0$ otherwise.

Can we find z^* in streaming setting?

<u>Theorem:</u> Count Sketch with $w = \frac{3K}{\varepsilon^2}$ and $d = \Omega(\log n)$ allows us to find a $\bar{z}$ such that $\|\bar{z}\|_0 \leq K$ and with high probability

$$\|\bar{x} - \bar{z}\|_2 < (1+\varepsilon)\, err_2^k(\bar{x}).$$

In particular if $\bar{x}$ is k-sparse then exact recovery.

Compressed Sensing and RIP Matrices

Count Sketch guarantees that we can recover any sparse $\bar{x}$ with high probability. Can we guarantee probability 1 with a linear sketch?

Yes!

There exist $\ell \times n$ matrices Π for $\ell = O(k \log \frac{n}{k})$ such that given any K-sparse $\bar{x} \in \mathbb{R}^n$

one can recover $\bar{x}$ from $\Pi \bar{x}$.

Note that $\Pi \bar{x}$ takes $O(l)$ space

and since $l = O(K \log \frac{n}{K})$ we

are not storing much more than

what we want to recover.

Such matrices are called RIP

matrices for "restricted isoperimetry property".

Turns out that a random $l \times n$ matrix with each entry chosen independently from a $N(0,1)$ Gaussian distribution satisfies the RIP. But cannot easily verify that a given matrix is RIP.

This area is called Compressed Sensing and has several applications in signal processing.