UNIVERSITÄT DES SAARLANDES

max planck institut informatik

Sublinear Algorithms

Lecture 03: Streaming III

Karl Bringmann

May 18, 2020

European Research Council
Established by the European Commission

Recap: Distinct Elements

approximate the number of distinct items among $x_1, \ldots, x_m \in [n]$

Data structure problem:

maintain set D and its size t

update(x): add x to D

query(): output $\tilde{t}$ with

$$\mathbb{P}\left[\left|\tilde{t} - t\right| \geq \varepsilon t\right] \leq \delta$$

Idealized Setting: FM (Flajolet,Martin 1985)

1) Pick random function $h: [n] \to [0,1]$

2) On update(): Maintain $X = \min_i h(x_i)$

3) On query(): Output $\tilde{t} = 1/X - 1$

Let $y_1, \ldots, y_t$ be the distinct items in the stream

Suppose that $y_1, \ldots, y_t$ are *random* in $[0,1]$

Then we expect $1/\min_i y_i \approx t$

Recap: Distinct Elements

approximate the number of distinct items among $x_1, \dots, x_m \in [n]$

Theoretical Variant: (Bar-Yossef et al. 2002)

1) Pick a prime m with $n^3 \leq m \leq n^{O(1)}$
2) Pick pairwise independent hash function $h' : [n] \to [m]$
3) Denote $h(x) \coloneqq h'(x)/m \in (0,1]$
4) Maintain a set L containing the $k \coloneqq \lceil 36/\varepsilon^2 \rceil$ smallest distinct values among $h(x_1), \dots, h(x_m)$
5) On query():

   If $|L| < k$: Output $\tilde{t} = |L|$

   Otherwise: Output $\tilde{t} = k/\max(L)$

Idealized Setting: FM (Flajolet,Martin 1985)

1) Pick random function $h: [n] \to [0,1]$
2) On update(): Maintain $X = \min_i h(x_i)$
3) On query(): Output $\tilde{t} = 1/X - 1$

Space usage: $O\left(\frac{1}{\varepsilon^2} \log n\right)$

Recap: Pairwise Independence

Random variables $X_1, \dots, X_n$ are **independent** if for any $j_1, \dots, j_n$ we have

$$\mathbb{P}[X_1 = j_1 \text{ and } \dots \text{ and } X_n = j_n] = \mathbb{P}[X_1 = j_1] \cdot \dots \cdot \mathbb{P}[X_n = j_n]$$

$k=n$

Random variables $X_1, \dots, X_n$ are **pairwise independent** if
for any $i \neq i'$ the random variables X_i and $X_{i'}$ are independent.

$k=2$

Lem: For pairwise independent $X_1, \dots, X_n$ we have

$$\mathrm{Var}[X_1 + \dots + X_n] = \mathrm{Var}[X_1] + \dots + \mathrm{Var}[X_n]$$

Random variables $X_1, \dots, X_n$ are **k-wise independent** if
for any distinct $i_1, \dots, i_k$ the random variables $X_{i_1}, \dots, X_{i_k}$ are independent.

Recap: Pairwise Independent Hash Function

Let p be a prime with $p \geq n$, and pick $h \in \mathcal{H}$ uniformly at random

Let $\mathcal{H}$ be the set of all functions $h: [n] \to [m]$ of the form

$h(i) = (a \cdot i + b) \bmod p$ where $a, b \in [p]$

Each hash value $h(i)$ is **uniformly distributed** in $[m]$

The random variables $h(1), \ldots, h(n)$ are **pairwise independent**

A function $h \in \mathcal{H}$ can be represented by the pair $(a, b) \in [p]^2$, using $2\lceil \log p \rceil$ bits

We can sample a function $h \in \mathcal{H}$ in time $O(1)$

k-wise Independent Hash Function

Let p be a prime with $p \geq n$, and pick $h \in \mathcal{H}$ uniformly at random

> Let $\mathcal{H}$ be the set of all functions $h: [n] \to [p]$ of the form
>
> $h(i) = \left(a_0 + a_1 \cdot i + \cdots + a_{k-1} \cdot i^{k-1}\right) \bmod p$ where $a_0, \dots, a_{k-1} \in [p]$

Each hash value $h(i)$ is **uniformly distributed** in $[m]$

The random variables $h(1), \dots, h(n)$ are **k-wise independent**

A function $h \in \mathcal{H}$ can be represented by the tuple $(a_0, \dots, a_{k-1}) \in [p]^k$, using $k[\log p]$ bits

We can sample a function $h \in \mathcal{H}$ in time $O(k)$

k-wise Independent Hash Function

Let p be a prime with $p \geq n$, and pick $h \in \mathcal{H}$ uniformly at random

Let $\mathcal{H}$ be the set of all functions $h: [n] \to [p]$ of the form

$h(i) = (a_0 + a_1 \cdot i + \cdots + a_{k-1} \cdot i^{k-1}) \bmod p$ where $a_0, \ldots, a_{k-1} \in [p]$

$\{0, 1, \ldots, 2^{\lambda}\}$

$\{-1, 1\}$

Arbitrary Codomain: Can we get codomain $Y = \{y_1, \ldots, y_t\}$ for $t \ll p$?

$\sigma(i) := y_{\underline{h(i) \bmod t}}$ is *almost uniform*, that is, $\mathbb{P}[\sigma(i) = y_j] = \underline{1/t} \pm \underline{O(1/p)}$

Lem: Fix k. For any n and Y, there is a family $\mathcal{H}$ of functions from $[n]$ to Y such that

- $h \in \mathcal{H}$ can be stored using $O(\log(n + |Y|))$ bits, and sampled in time $O(1)$,
- for random $h \in \mathcal{H}$ and fixed i, the value $h(i)$ is (~~almost~~) **uniformly distributed** in Y,
- for random $h \in \mathcal{H}$, the random variables $h(1), \ldots, h(n)$ are k**-wise independent.**

Outline

1) **Turnstile Model + Moment Estimation**
2) Point Query + Heavy Hitters

Generalized Streaming Model

maintain a vector $x \in \mathbb{Z}^n$ under updates of the form $x_i = x_i + \Delta$

General data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update(i, Δ): $x_i = x_i + \Delta$

query(): approximate $f(x)$

Insertion-only:

Each update has $\Delta = 1$

This is what we studied so far

(E.g. distinct elements: #non-zero x_i's)

$z_1, \ldots, z_m \in [n]$

$x_i = |\{j \mid z_j = i\}|$

Generalized Streaming Model

maintain a vector $x \in \mathbb{Z}^n$ under updates of the form $x_i = x_i + \Delta$

General data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update(i, Δ): $x_i = x_i + \Delta$

query(): approximate $f(x)$

©Karen Foto

Insertion-only: Each update has $\Delta = 1$

one-way

This is what we studied so far

(E.g. distinct elements: #non-zero x_i's)

Strict Turnstile: $\Delta \in \mathbb{Z}$ (may be negative!)

two-way
closed room

Promise: $x_i \geq 0$ for all i at all times

General Turnstile: $\Delta \in \mathbb{Z}$, $x_i \in \mathbb{Z}$

two-way
open

Second Moment Estimation

Goal: $O(\log n)$ space

approximate $F_2 = \|x\|_2^2 = \sum_{i=1}^n x_i^2$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update(i, Δ): $x_i = x_i + \Delta$

query(): output $\tilde{F}_2$ with

$\mathbb{P}\left[\left|\tilde{F}_2 - F_2\right| \geq \varepsilon F_2\right] \leq \delta$

AMS Sketch: (Alon, Matias, Szegedy 1999)

1) Pick 4-wise independent hash function $\sigma: [n] \to \{-1,1\}$
2) Maintain $y = \sum_{i=1}^n \sigma(i) \cdot x_i$
3) Output y^2

initially $x = (0, \dots, 0)$

assume $|\Delta| = n^{O(1)}$

so entries of x are $O(\log n)$-bit integers

update(i, Δ): $y = y + \sigma(i) \cdot \Delta$

Second Moment Estimation

approximate $F_2 = \|x\|_2^2 = \sum_{i=1}^n x_i^2$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Standard Approach:

> **Lem:** AMS Sketch is an unbiased estimator,
> that is, $\mathbb{E}[y^2] = \|x\|_2^2$

AMS Sketch: (Alon, Matias, Szegedy 1999)

1) Pick 4-wise independent $\sigma: [n] \to \{-1,1\}$
2) Maintain $y = \sum_{i=1}^n \sigma(i) \cdot x_i$
3) Output y^2

$$y^2 = \sum_i x_i^2 + \sum_{i \neq j} \sigma(i)\sigma(j) x_i x_j$$

Proof:

$$\mathbb{E}[y^2] = \mathbb{E}[(\textstyle\sum_{i=1}^n \sigma(i) x_i)^2]$$

$$= \mathbb{E}[\textstyle\sum_{i,j} \sigma(i)\sigma(j) x_i x_j]$$

independent

$$= \sum_i \underbrace{\mathbb{E}[\sigma(i)^2]}_{1} x_i^2 + \sum_{i\neq j} \underbrace{\mathbb{E}[\sigma(i)\sigma(j)]}_{= \underbrace{\mathbb{E}[\sigma(i)]}_{0} \cdot \underbrace{\mathbb{E}[\sigma(j)]}_{=0}} x_i x_j$$

$$= \boxed{\sum_i x_i^2}$$

...since σ is pairwise independent

Second Moment Estimation

approximate $F_2 = \|x\|_2^2 = \sum_{i=1}^n x_i^2$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Standard Approach:

Lem: AMS Sketch is an unbiased estimator, that is, $\mathbb{E}[y^2] = \|x\|_2^2$

Lem: We have $\mathbb{E}[y^4] \le 3\mathbb{E}[y^2]^2$.

AMS Sketch: (Alon, Matias, Szegedy 1999)

1) Pick 4-wise independent $\sigma: [n] \to \{-1,1\}$
2) Maintain $y = \sum_{i=1}^n \sigma(i) \cdot x_i$
3) Output y^2

Proof: $\mathbb{E}[y^4] = \mathbb{E}[(\sum_{i=1}^n \sigma(i)x_i)^4] \overset{?}{=} \sum_{i=1}^n \mathbb{E}[(\sigma(i)x_i)^4] + 6\sum_{i<j} \underbrace{\mathbb{E}[(\sigma(i)x_i)^2]}_{x_i^2} \cdot \underbrace{\mathbb{E}\left[\left(\sigma(j)x_j\right)^2\right]}_{x_j^2}$

...since σ is 4-wise independent and has expectation 0, see Exercise Sheet 1

$= \sum_{i=1}^n x_i^4 + 6\sum_{i<j} x_i^2 \cdot x_j^2 \le 3\left(\sum_{i=1}^n x_i^4 + 2\sum_{i<j} x_i^2 \cdot x_j^2\right) = 3\left(\sum_{i=1}^n x_i^2\right)^2 = 3\mathbb{E}[y^2]^2$

Second Moment Estimation

approximate $F_2 = \|x\|_2^2 = \sum_{i=1}^n x_i^2$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Standard Approach:

Lem: AMS Sketch is an unbiased estimator, that is, $\mathbb{E}[y^2] = \|x\|_2^2$

Lem: We have $\mathbb{E}[y^4] \leq 3\mathbb{E}[y^2]^2$.

By Chebyshev: $\mathbb{P}[|y^2 - F_2| \geq \varepsilon F_2] \leq \boxed{\frac{2}{\varepsilon^2}}$

AMS Sketch: (Alon, Matias, Szegedy 1999)

1) Pick 4-wise independent $\sigma: [n] \to \{-1,1\}$
2) Maintain $y = \sum_{i=1}^n \sigma(i) \cdot x_i$
3) Output y^2

Boosting via Chebyshev: AMS+ = average over $\frac{8}{\varepsilon^2}$ runs of AMS has error prob. $1/4$

Boosting via Chernoff: AMS++ = median of $8\log(2/\delta)$ runs of AMS+ has error prob. δ

Second Moment Estimation

approximate $F_2 = \|x\|_2^2 = \sum_{i=1}^n x_i^2$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update(i, Δ): $x_i = x_i + \Delta$

query(): output $\tilde{F}_2$ with

$$\mathbb{P}\left[\left|\tilde{F}_2 - F_2\right| \geq \varepsilon F_2\right] \leq \delta$$

AMS Sketch: (Alon, Matias, Szegedy 1999)

1) Pick 4-wise independent $\sigma: [n] \to \{-1,1\}$

2) Maintain $\boxed{y} = \sum_{i=1}^n \sigma(i) \cdot \boxed{x_i}$

3) Output y^2

AMS+: average over $O(1/\varepsilon^2)$ runs of AMS

AMS++: median of $O(\log(1/\delta))$ runs of AMS+

Space usage: $O\left(\underbrace{\frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)}\underbrace{\log n}\right)$ bits

Remark 1: Moment Estimation

approximate $F_p = \|x\|_p^p = \sum_{i=1}^n |x_i|^p$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

For $0 \leq p \leq 2$ and constant δ: There is streaming algorithm using $\mathrm{poly}(\varepsilon^{-1} \log n)$ space.

For $p > 2$ and constant ε, δ: Space complexity is $n^{1-2/p}$ up to logfactors.

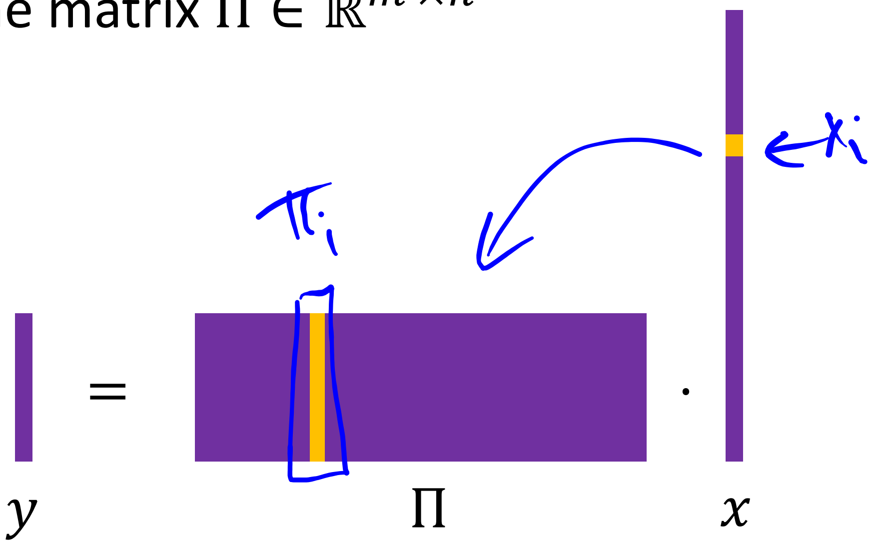
[Alon,Matias,Szegedy'99, Bar-Yossef,Jayram,Kumar,Sivakumar'04, Indyk,Woodruff'05, Indyk'06]

Remark 2: Linear Sketch

... is an algorithm that maintains Πx, for some matrix $\Pi \in \mathbb{R}^{m \times n}$

Want to maintain vector $x \in \mathbb{Z}^n$

Pick a suitable matrix Π
and instead maintain $y \in \mathbb{Z}^m$ with $y = \Pi x$



We cannot store Π explicitly!

Implicit representation of Π: Given i, j we can efficiently compute the entry $\Pi_{i,j}$

On update(i, Δ): $y = y + \underline{\Delta \cdot \Pi_i}$, where Π_i is the i-th column of Π $\rightarrow$ time $O(m)$

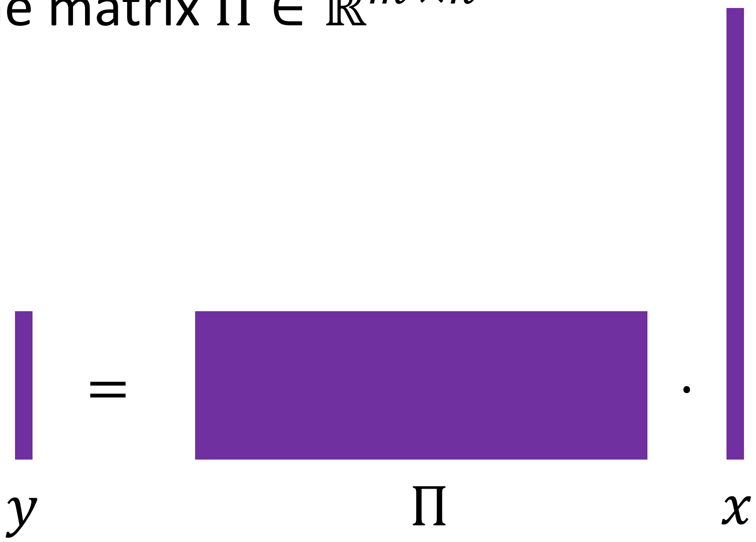
Π may be **randomized** (that is, chosen from some probability distribution)

Remark 2: Linear Sketch

… is an algorithm that maintains Πx, for some matrix $\Pi \in \mathbb{R}^{m \times n}$

Want to maintain vector $x \in \mathbb{Z}^n$

Pick a suitable matrix Π
and instead maintain $y \in \mathbb{Z}^m$ with $y = \Pi x$



Any algorithm in strict/general turnstile model can be converted into a linear sketch,
at the cost of at most a logarithmic factor in the space bound.

[Li,Nguyen,Woodruff'14]

Remark 2: Linear Sketch

approximate $F_2 = \|x\|_2^2 = \sum_{i=1}^{n} x_i^2$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update($\boldsymbol{i}, \boldsymbol{\Delta}$): $x_i = x_i + \Delta$

query(): output $\tilde{F}$ with

$$\mathbb{P}\left[\left|\tilde{F} - F_2\right| \geq \varepsilon F_2\right] \leq \delta$$

AMS+ Sketch: (Alon, Matias, Szegedy 1999)

1) Pick 4-wise independent hash functions $\sigma_1, \dots, \sigma_s : [n] \to \{-1,1\}$

2) Maintain $y_j = \sum_{i=1}^{n} \boxed{\sigma_j(i)} \cdot x_i$

3) Output $\frac{1}{s} \sum_{j=1}^{s} y_j^2$

This is a linear sketch!

$(y_1, \dots, y_s) = \boxed{y} = \Pi x$ with $\Pi_{j,i} = \underline{\sigma_j(i)}$

Π is implicitly represented

Outline

1) Turnstile Model + Moment Estimation
2) **Point Query + Heavy Hitters**

$\# \{ i \mid x_i > \varepsilon \|x\|_1 \} \leq 1/\varepsilon$

Point Query

Goal: $O(\log n)$ space

approximate $x_i \pm \underline{\varepsilon \|x\|_1}$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update(i, Δ): $x_i = x_i + \Delta$

query(i): output $\tilde{x}_i$ with

$\mathbb{P}[\underline{|\tilde{x}_i - x_i|} \geq \underline{\varepsilon \|x\|_1}] \leq \delta$

CountMin Sketch: (Cormode,Muthukrishnan'05)

1) Pick 2-wise independent hash function $h: [n] \to \underline{[t]}$ for $t = \lceil 4/\varepsilon \rceil$

2) Maintain counters $C_j = \sum_{\underline{i \text{ s.t. } h(i)=j}} x_i$

That is, initially $C_1, \ldots, C_t = 0$

On update(i, Δ): Add Δ to $\underline{C_{h(i)}}$

3) On query(i): output $\boxed{C_{h(i)}}$

$\left(x_i \pm 1/3 \rightarrow x_i \begin{smallmatrix} \nearrow 0 \\ \searrow 1 \end{smallmatrix} \rightarrow n \text{ bits} \right)$

$\|x\|_1 = \sum_i |x_i|$

Point Query

approximate $x_i \pm \varepsilon\|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Lem: $\mathbb{P}\left[\left|C_{h(i)} - x_i\right| > \varepsilon\|x\|_1\right] \le 1/4$

Proof: Fix i. For $j \neq i$:

$$Z_j = \begin{cases} 1, & \text{if } h(j) = h(i) \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbb{E}[Z_j] = \mathbb{P}[Z_j = 1] = 1/t$$

$$C_{h(i)} = \boxed{x_i} + \sum_{j \neq i} x_j Z_j$$

$$\left|C_{h(i)} - x_i\right| \le \sum_{j \neq i} |x_j| \cdot Z_j$$

(Markov)

$$\mathbb{P}\left[\left|C_{h(i)} - x_i\right| > \varepsilon\|x\|_1\right] \le \mathbb{P}\Big[\overbrace{\textstyle\sum_{j \neq i} |x_j| \cdot Z_j}^{\ge 0} > \varepsilon\|x\|_1\Big] \le \frac{\sum_{j \neq i} |x_j|/t}{\varepsilon\|x\|_1} \le \frac{1}{\varepsilon t} \le \frac{1}{4} \checkmark$$

CountMin Sketch: (Cormode,Muthukrishnan'05)

1) Pick 2-wise independent hash function $h: [n] \to [t]$ for $t = \lceil 4/\varepsilon \rceil$

2) Maintain counters $C_j = \sum_{i \text{ s.t. } h(i)=j} x_i$

   That is, initially $C_1, \dots, C_t = 0$

   On update(i, Δ): Add Δ to $C_{h(i)}$

3) On query(i): output $C_{h(i)}$

Point Query

approximate $x_i \pm \varepsilon \|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in turnstile model

Data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update($\boldsymbol{i}, \boldsymbol{\Delta}$): $x_i = x_i + \Delta$

query($\boldsymbol{i}$): output $\tilde{x}_i$ with

$\mathbb{P}[|\tilde{x}_i - x_i| \geq \varepsilon \|x\|_1] \leq \delta$

Space usage: $O\left(\frac{1}{\varepsilon} \log\left(\frac{1}{\delta}\right) \log n\right)$

In **strict** turnstile model we can use *minimum* instead of *median*

CountMin Sketch: (Cormode,Muthukrishnan'05)

1) Pick 2-wise independent hash function $h: [n] \to [t]$ for $t = \lceil 4/\varepsilon \rceil$

2) Maintain counters $C_j = \sum_{i \text{ s.t. } h(i)=j} x_i$

That is, initially $C_1, \ldots, C_t = 0$

On update(i, Δ): Add Δ to $C_{h(i)}$

3) On query(i): output $C_{h(i)}$

CM++: median of $O(\log(1/\delta))$ runs of CM

$\|x\|_1 = \sum_i |x_i|$

$\le 1/\varepsilon$ many

Heavy Hitters

Goal: $O(\text{polylog}\, n)$ space

compute all i with $x_i \geq \varepsilon \|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in **strict** turnstile model

Data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update(i, Δ): $x_i = x_i + \Delta$

query($\cancel{i}$): output a set R s.t.

R contains **all** i with $x_i \geq \varepsilon \|x\|_1$

R contains **no** i with $x_i < \frac{\varepsilon}{2} \|x\|_1$

with failure probability δ

Heavy Hitters: Dyadic Trick

compute all i with $x_i \geq \varepsilon \|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in $\boxed{\textbf{strict}}$ turnstile model

Data structure problem:

maintain vector $x \in \mathbb{Z}^n$

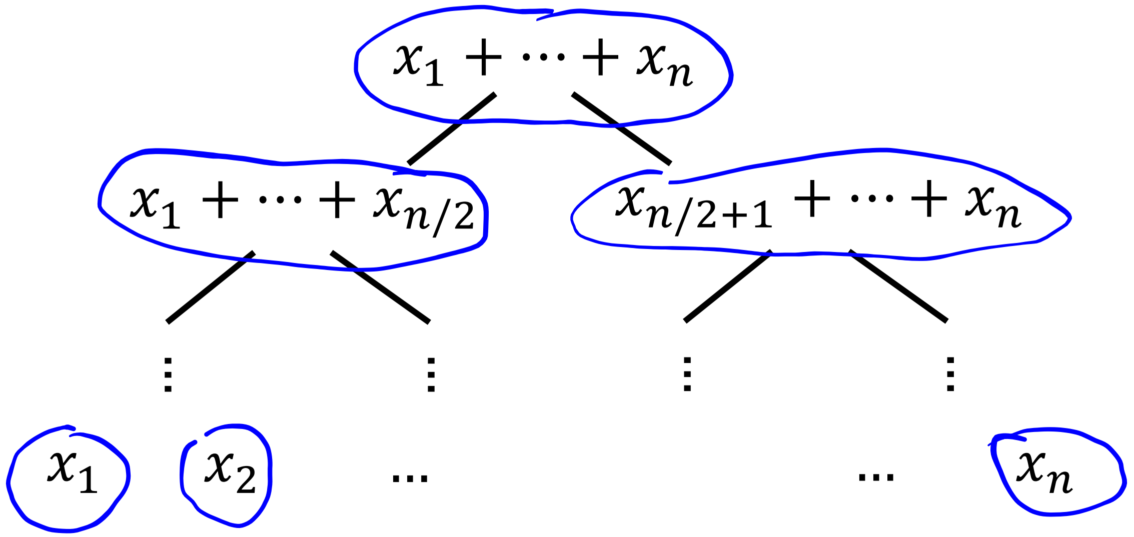
update(i, Δ): $x_i = x_i + \Delta$

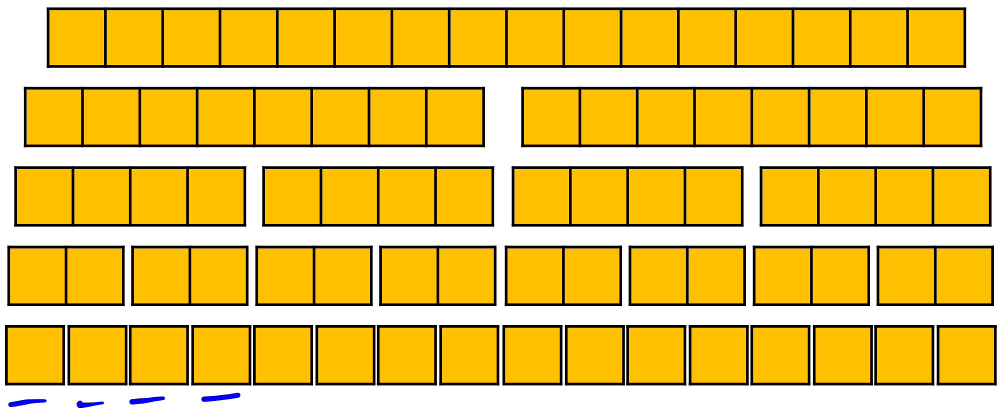
query(~~i~~): output a set R s.t.

R contains **all** i with $x_i \geq \varepsilon \|x\|_1$

R contains **no** i with $x_i < \frac{\varepsilon}{2} \|x\|_1$

with failure probability δ





Heavy Hitters: Dyadic Trick

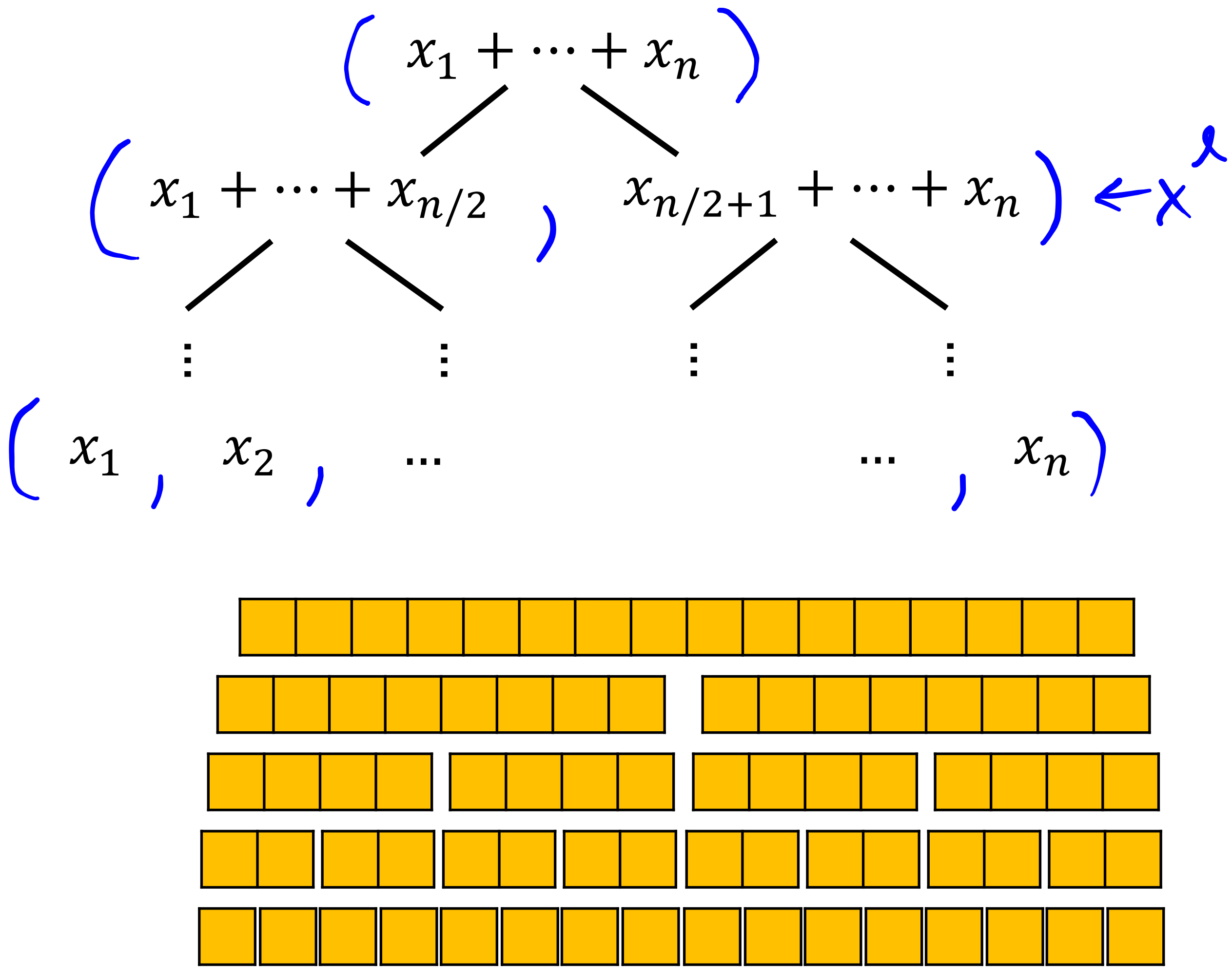
compute all i with $x_i \geq \varepsilon \|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in **strict** turnstile model

Level ℓ corresponds to vector $\boxed{x^\ell} \in \mathbb{Z}^{2^\ell}$ with

$$x_i^\ell = x_{(i-1)n/2^\ell+1} + \cdots + x_{in/2^\ell}$$

Store a CM++ sketch for every vector x^ℓ

with $\mathbb{P}\left[\left|\tilde{x}_i^\ell - x_i^\ell\right| \geq \frac{\varepsilon}{4} \left\|x^\ell\right\|_1\right] \leq \frac{\delta\varepsilon}{4 \log n}$



Heavy Hitters: Dyadic Trick

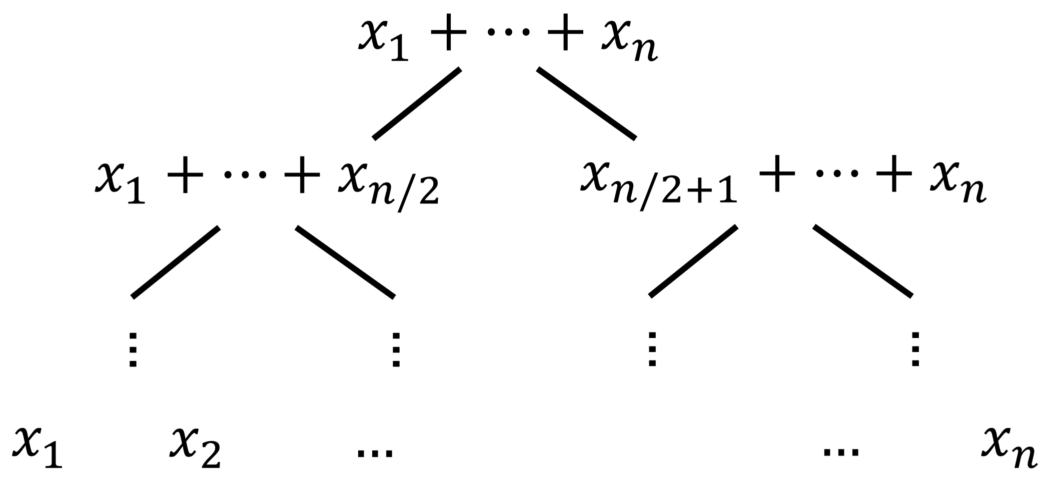
compute all i with $x_i \geq \varepsilon \|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in **strict** turnstile model

Level ℓ corresponds to vector $x^\ell \in \mathbb{Z}^{2^\ell}$ with

$$x_i^\ell = x_{(i-1)n/2^\ell+1} + \cdots + x_{in/2^\ell}$$

Store a CM++ sketch for every vector x^ℓ

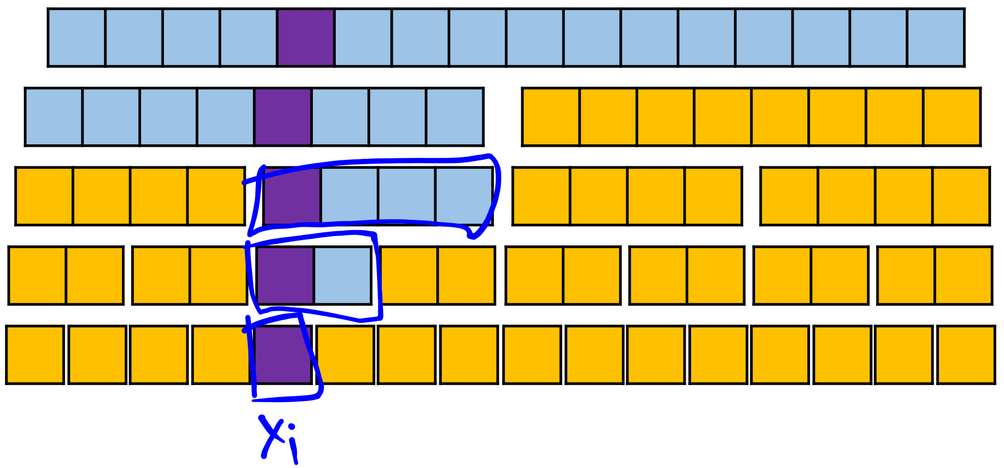
with $\mathbb{P}\left[\left|\tilde{x}_i^\ell - x_i^\ell\right| \geq \frac{\varepsilon}{4}\left\|x^\ell\right\|_1\right] \leq \frac{\delta\varepsilon}{4\log n}$



On update(i, Δ):

Affects one entry of every vector x^ℓ

So perform $O(\log n)$ updates of CM++



Heavy Hitters: Dyadic Trick

compute all i with $x_i \geq \varepsilon \|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in **strict** turnstile model

Level ℓ corresponds to vector $x^\ell \in \mathbb{Z}^{2^\ell}$ with

$$x_i^\ell = x_{(i-1)n/2^\ell+1} + \cdots + x_{in/2^\ell}$$

Store a CM++ sketch for every vector x^ℓ

with $\mathbb{P}\left[\left|\tilde{x}_i^\ell - x_i^\ell\right| \geq \frac{\varepsilon}{4}\left\|x^\ell\right\|_1\right] \leq \frac{\delta\varepsilon}{4\log n}$

DFS($\boxed{x_i^\ell}$): node (ℓ, i)

Use CM++ sketch to decide if $x_i^\ell \geq \frac{3}{4}\varepsilon\|x\|_1$

If CM++ says „larger“: // x_i^ℓ is heavy hitter

If x_i^ℓ is a leaf: add i to result R

Else: run DFS($x_{2i-1}^{\ell+1}$) and DFS($x_{2i}^{\ell+1}$)

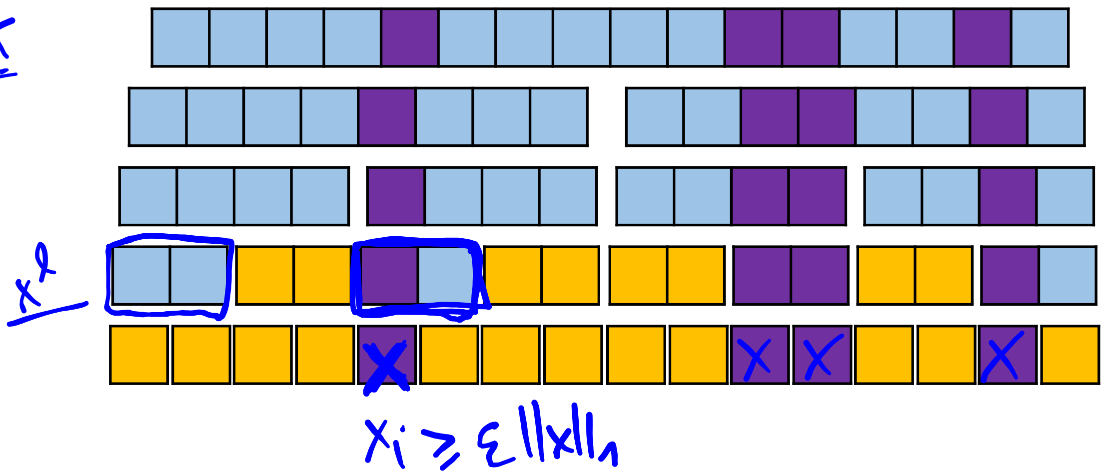
On query():

Same norm on every level: $\left\|x^\ell\right\|_1 = \|x\|_1$ → strict

Every node is less than or equal to its parent

So all heavy hitters on all levels form a **subtree**

There are $O(1/\varepsilon)$ heavy hitters on each level



Heavy Hitters: Dyadic Trick

compute all i with $x_i \geq \varepsilon\|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in **strict** turnstile model

Level ℓ corresponds to vector $x^\ell \in \mathbb{Z}^{2^\ell}$ with

$$x_i^\ell = x_{(i-1)n/2^\ell+1} + \cdots + x_{in/2^\ell}$$

Store a CM++ sketch for every vector x^ℓ

with $\mathbb{P}\left[\left|\tilde{x}_i^\ell - x_i^\ell\right| \geq \frac{\varepsilon}{4}\left\|x^\ell\right\|_1\right] \leq \frac{\delta\varepsilon}{4\log n}$

DFS(x_i^ℓ):

Use CM++ sketch to decide if $x_i^\ell \geq \frac{3}{4}\varepsilon\|x\|_1$

If CM++ says „larger“:

If x_i^ℓ is a leaf: add i to result R

Else: run DFS($x_{2i-1}^{\ell+1}$) and DFS($x_{2i}^{\ell+1}$)

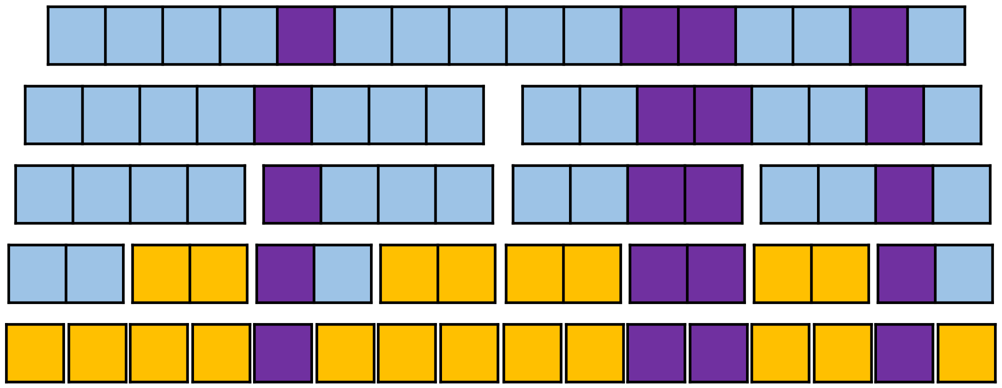
On query():

Assuming correctness of CM++:

CM++ says „larger“ only if $\boxed{x_i^\ell \geq \frac{\varepsilon}{2}\|x\|_1}$

On each level we explore the children of $\leq 2/\varepsilon$ nodes

So $\leq \frac{4}{\varepsilon}\log n$ calls to CM++, so error probability δ



Heavy Hitters: Dyadic Trick

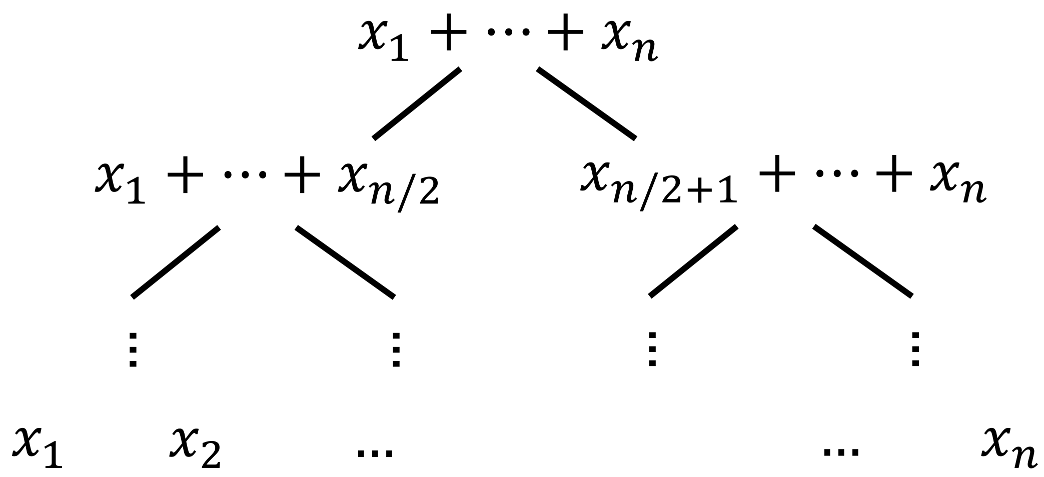
compute all i with $x_i \geq \varepsilon \|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in **strict** turnstile model

Level ℓ corresponds to vector $x^\ell \in \mathbb{Z}^{2^\ell}$ with

$$x_i^\ell = x_{(i-1)n/2^\ell+1} + \cdots + x_{in/2^\ell}$$

Store a CM++ sketch for every vector x^ℓ

with $\mathbb{P}\left[\left|\tilde{x}_i^\ell - x_i^\ell\right| \geq \frac{\varepsilon}{4}\left\|x^\ell\right\|_1\right] \leq \frac{\delta\varepsilon}{4\log n}$

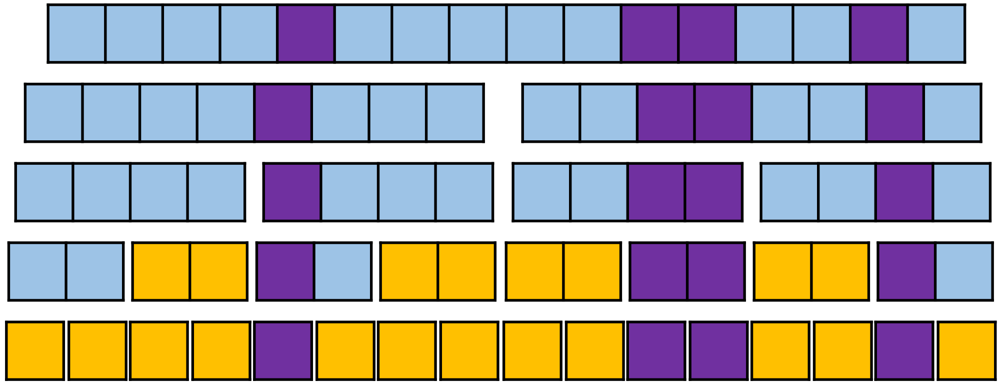


CM++: $O(\frac{1}{\varepsilon} \log \frac{1}{\delta} \cdot \log n)$

#levels

Space: $\underbrace{O(\log n)}_{\text{\#levels}} \cdot O\left(\frac{1}{\varepsilon}\log\left(\frac{\log n}{\delta\varepsilon}\right)\log n\right)$ bits

$$= O\left(\frac{1}{\varepsilon}\log^2 n\left(\log\log n + \log\left(\frac{1}{\delta\varepsilon}\right)\right)\right)$$



Heavy Hitters

Goal: $O(\text{polylog}\, n)$ space

compute all i with $x_i \geq \varepsilon \|x\|_1$ for a vector $x \in \mathbb{Z}^n$ given in **strict** turnstile model

Data structure problem:

maintain vector $x \in \mathbb{Z}^n$

update(i, Δ): $x_i = x_i + \Delta$

query(~~i~~): output a set R s.t.

R contains **all** i with $x_i \geq \varepsilon \|x\|_1$

R contains **no** i with $x_i < \frac{\varepsilon}{2} \|x\|_1$

with failure probability δ

Level ℓ corresponds to vector $x^\ell \in \mathbb{Z}^{2^\ell}$ with

$$x_i^\ell = x_{(i-1)n/2^\ell+1} + \cdots + x_{in/2^\ell}$$

Store a CM++ sketch for every vector x^ℓ

with $\mathbb{P}\left[\left|\tilde{x}_i^\ell - x_i^\ell\right| \geq \frac{\varepsilon}{4} \left\|x^\ell\right\|_1\right] \leq \frac{\delta\varepsilon}{4 \log n}$

CM++: $O(\log 1/\delta)$

Update time: $O\left(\log n \left(\log\log n + \log\left(\frac{1}{\delta\varepsilon}\right)\right)\right)$

Query time: $O\left(\frac{1}{\varepsilon} \log n \left(\log\log n + \log\left(\frac{1}{\delta\varepsilon}\right)\right)\right)$

Space: $O\left(\frac{1}{\varepsilon} \log^2 n \left(\log\log n + \log\left(\frac{1}{\delta\varepsilon}\right)\right)\right)$ bits

More Material

Moment estimation, AMS Sketch:

[Alon, Matias, Szegedy „The space complexity of approximating the ~~d~~frequency moments“ 1999]

Point Query + Heavy Hitters, CountMin Sketch:

[Cormode, Muthukrishnan „An improved data stream summary: the count-min sketch and its applications“ 2005]

– Course Website → Material → Link to Summer School on Streaming by Jelani Nelson

– **Exercise Sheet 1** due on **Friday, May 22**

Nect lecture by Vasileios Nakos on May 28!