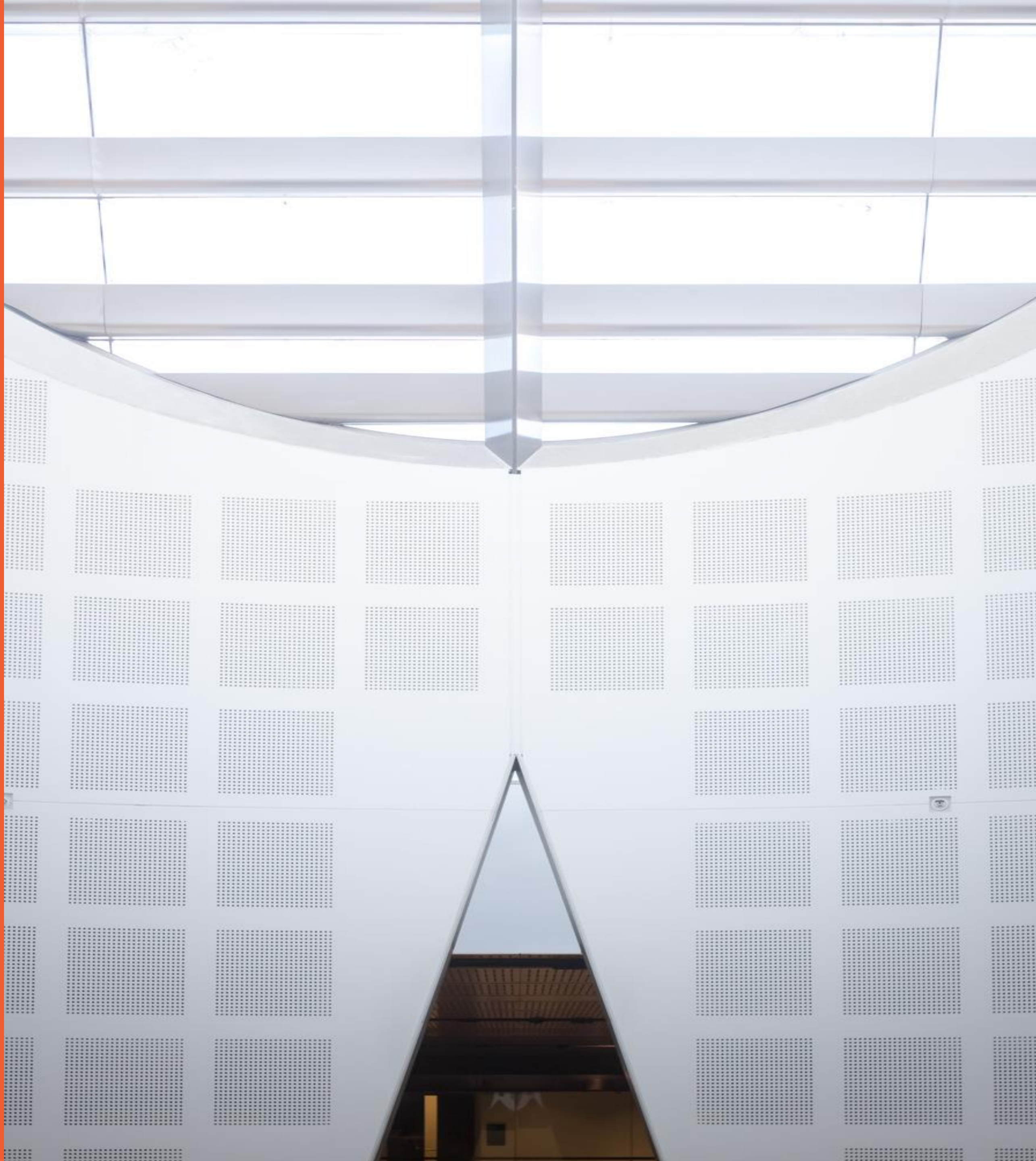
COMPx270: Randomised and Advanced Algorithms
Lecture 8: Streaming and Sketching I

Clément Canonne

School of Computer Science

THE UNIVERSITY OF SYDNEY



Some housekeeping

- A2 due this Friday
- Office Hours (OH) tomorrow, 2:30-3:30pm in J12 402 (+Zoom)
- Preliminary marks for A1 released this evening on Gradescope

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(1,2)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(2,4)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(1,2)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(4,5)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(4,5)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(3,4)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(3,6)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(1,4)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(4,6)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(3,5)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(3,4)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(4,5)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(3,4)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(3,6)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

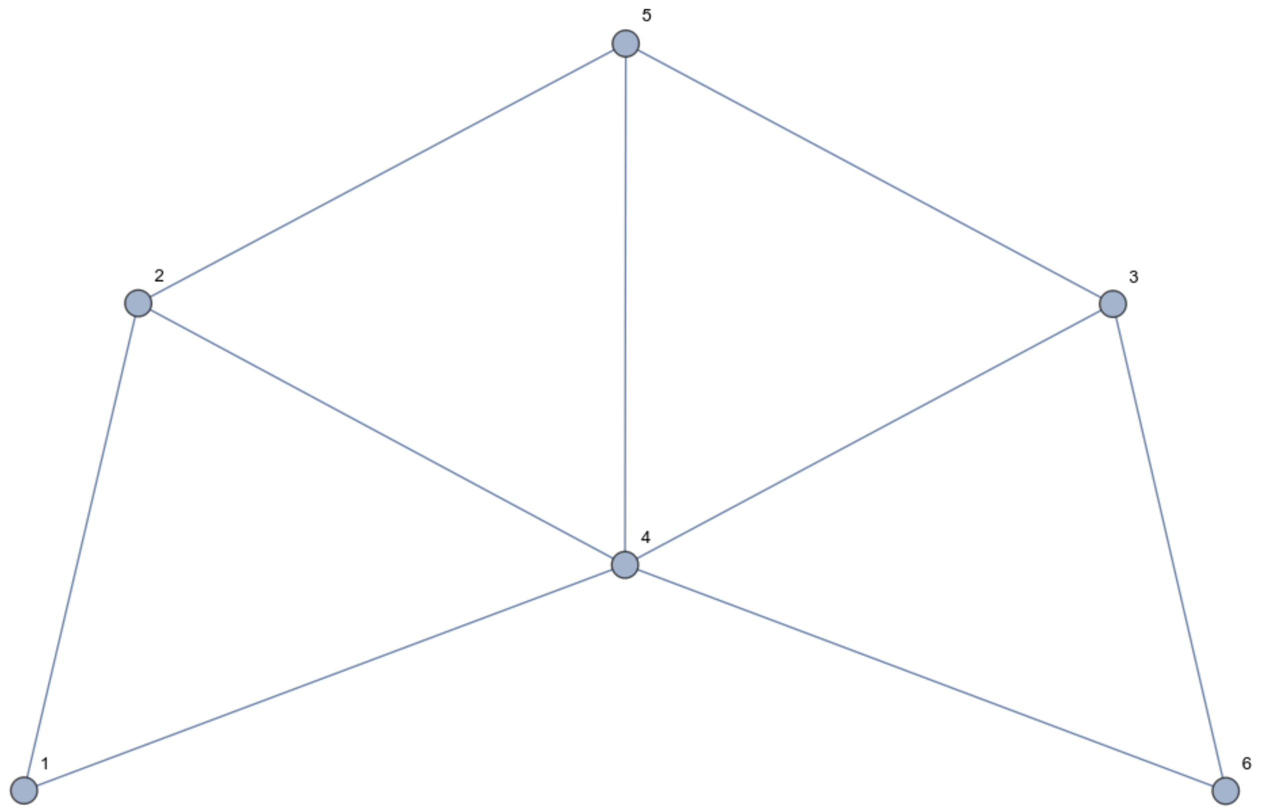
(1,4)

A question

You have a graph, coming one edge at a time, with possible duplicates, and no paper to write anything done, only your memory. What is its average degree?

(4,6)

A question (an answer)



Streaming algorithms: what? (1/3)

→ Low memory: cannot store the whole input.

→ Input comes as a "stream": sequence σ of length m

$$\sigma = (a_1, \ldots, a_m), \quad a_i \in \mathcal{X} \text{ of size } n$$

(could store this in $O(m \log n)$ or $O(n \log m)$ bits → NOT ALLOWED)

Worst-case order (arbitrary)

→ p-pass algorithms: get to see σ p times

For us, $p = 1$ (unless "I say so")

→ "cash register" model: don't remove parts of the input

SPACE: $o(\min(n, m))$

HOPE: $O(\log n + \log m)$

VERY GOOD: $\mathrm{polylog}(n, m)$

Streaming algorithms: what? (2/3)

- Randomised
- Approximate:

  ↓

  Want to compute some value $v \geq 0$

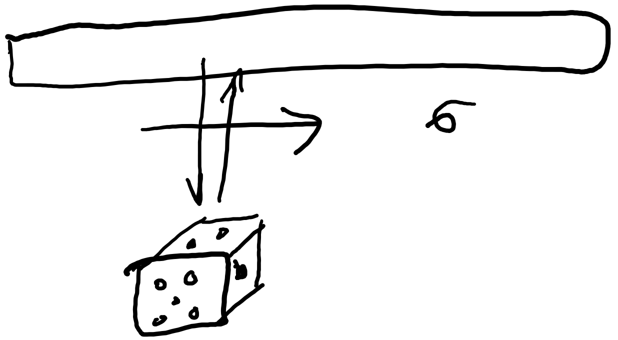
<u>Multiplicative</u>: $\Pr[\,|\hat{v} - v| \leq \varepsilon v\,] \geq 1 - \delta$ (A)

<u>Additive</u>: $\Pr[\,|\hat{v} - v| \leq \varepsilon\,] \geq 1 - \delta$

Streaming algorithms: what? (3/3)

- MAJORITY (HEAVY HITTERS)
- COUNTING
- DISTINCT ELEMENTS → "F_0"

RANDOMNESS



$\sigma = (a_1, \dots, a_m) \in [n]^m$

$\forall j \in [n] \quad f_j = \#(\text{times } j \text{ appears}) = \sum_{i=1}^{m} \mathbb{1}_{a_i = j}$ "frequency" of j → $\vec{f} = (f_1, \dots, f_n)$

$0 \leq f_j \leq m, \quad \underbrace{\sum_{j=1}^{n} f_j}_{\|\vec{f}\|_1} = m$

First example: Majority

MAJ: "Is there an element $i^* \in [n]$ s.t. $f_{i^*} \geq \frac{m}{2}$?" (at most 2)

ε-HH: "All elem'ts $i \in [n]$ s.t. $f_i \geq \varepsilon m$" → at most $\frac{1}{\varepsilon}$

Want to solve this in one pass.

We'll see how in two passes, deterministically.

First example: Majority (Frequency Estimation)

MISRA-GRIES : returns $\hat{f}_1, \ldots, \hat{f}_n$

st. $f_j - \varepsilon m \le \hat{f}_j \le f_j \quad \forall j$

one pass.

$(\varepsilon = \frac{1}{2}$ for MAJORITY$)$

If $\hat{f}_j \ge \frac{m}{2} \rightarrow$ candidate

+ second pass to check

2-pass alg for MAJ

First example: the Misra-Gries algorithm (1/3)

$A \leftarrow n$ zeroes $\quad \left(k = \frac{1}{\varepsilon}\right)$

At step $i \in [m]$: get a_i

- If $A[a_i] > 0$
  - $A[a_i] \mathrel{+}= 1$
- If $A[a_i] = 0$ but (# of non-zeroes of A) $< k-1$
  - $A[a_i] = 1$
- If $A[a_i] = 0$ but ( ) $= k-1$
  - For all j st $A[j] > 0$:
    - $A[j] = A[j] - 1$

End: return all j's st $A[j] > 0$

If implemented smartly (not w/ an array, but a BST)

space:

$\leq k\, O(\log n + \log m)$

$= O(k \log nm)$

$= O\left(\frac{\log nm}{\varepsilon}\right)$

First example: the Misra-Gries algorithm, alternative view (2/3)

$$\hat{f}_j \leq f_j \quad \forall j \quad \text{"easy"}$$

$$\hat{f}_j \geq f_j - \varepsilon m \;?$$

($\uparrow$ $\varepsilon = \frac{1}{k}$)

Every time I "decrement" this is for $k \neq$ elements from the stream so far

First example: the Misra-Gries algorithm (3/3)

Theorem 39. *The* Misra-Gries *algorithm is a* deterministic *one-pass algorithm which, for any given parameter $\varepsilon \in (0,1]$, provides $\hat{f}_1, \dots, \hat{f}_n$ of all element frequencies such that*

$$f_j - \varepsilon m \leq \hat{f}_j \leq f_j, \qquad j \in [n]$$

with space complexity $s = O(\log(mn)/\varepsilon)$. (In particular, it can be used to solve the Majority *problem in two passes.)*

Second example: Approximate Counting

$n = 2$ $\quad a_i \in \{0,1\}$ $\quad$ Want. $d = \sum_{i=1}^{n} a_i$

- $O(\log m)$ exact is trivial
- 2-estimate : $O(\log \log m)$ space

Morris

Algo.

$x \leftarrow 0$

Step i : observe a_i

If $a_i = 1$: increment x w.p. $\frac{1}{2^x}$

End: output $2^x - 1$

At the end

$E[2^x - 1]$?

$Var[2^x]$?

$C_i = 2^x$ at step i

$E[C]$

$Var[C]$

Second example: Approximate Counting and the Morris Counter

Alt. view:

$C \leftarrow 1$

At step i:

If $a_i = 1$ $\quad C \leftarrow 2C$ w/ $\frac{1}{C}$

Return $C-1$

(☆)

If $a_i = 1$

$$\mathbb{E}[C_{i+1} \mid C_i] = \frac{1}{C_i} 2C_i + \left(1 - \frac{1}{C_i}\right) C_i$$

$$= 2 + C_i - 1$$

$$= C_i + 1$$

✓

$$\mathbb{E}[C] = \mathbb{E}[C_m] \underset{☆}{=} \sum_{i=1}^{m} a_i + 1 = d + 1$$

1: original value

$$\mathbb{E}[C_m] = \mathbb{E}[\mathbb{E}[C_m \mid C_{m-1}]] = \mathbb{E}[C_{m-1}] + a_m$$

E ✓

Var ?

Second example: the Morris Counter (~~1/~~3)

$\mathbb{E}[C_m^2]$?   Compute $\mathbb{E}[C_i^2]$.

$$\mathbb{E}[C_{i+1}^2] = \mathbb{E}\left[\mathbb{E}[C_{i+1}^2 \mid C_i]\right]$$

$$= \mathbb{E}\left[\frac{1}{C_i}(2C_i)^2 + \left(1 - \frac{1}{C_i}\right)C_i^2\right]$$

$$= \mathbb{E}[4C_i + C_i^2 - C_i] = \mathbb{E}[C_i^2 + 3C_i]$$

$$= \mathbb{E}[C_i^2] + 3\,\mathbb{E}[C_i]$$

↖ use previous slide:

$$= \mathbb{E}[C_i^2] + 3\left(\sum_{j=1}^{i} a_j + 1\right)$$

previous: $\mathbb{E}[C_m]^2$

↓

$(\ldots)$

$$\mathbb{E}[C_m^2] = 1 + \frac{3d(d+1)}{2}$$

$\longrightarrow \mathrm{Var}\, C_m = \mathbb{E}\, C_m^2 - (d+1)^2 = \frac{d(d-1)}{2}$

Second example: the Morris Counter (2/3)

$\mathbb{E}[C_m] = d+1$ ✓

$\mathrm{Var}[C_m] = \frac{d(d-1)}{2} = \Theta(d^2)$ ✗

Bad

Chebyshev gives

$C_m \approx d+1 \pm \underbrace{\Theta\left(\sqrt{\mathrm{Var}\, C_m}\right)}_{\Theta(d)}$

vacuous guarantee.

Doom?

Second example: the Morris Counter (3/3)

Two options:

① Amplify?

~~Median trick~~

loss error ← | Reduce variance: mean of k indept counters

~~+ Chebyshev~~

<u>then</u> median trick

↳ better proba.

<u>Median-of-means</u>

Second example: the Morris Counter, Median-of-Means

Theorem 40. *The medians-of-means version of the* MORRIS COUNTER *is a* randomised *one-pass algorithm which, for any given parameters* $\varepsilon, \delta \in (0,1]$, *provides an estimate* $\widehat{d}$ *of the number* d *of non-zero elements of the stream such that*

$$\Pr\left[(1-\varepsilon)d \leq \widehat{d} \leq (1+\varepsilon)d \right] \geq 1-\delta$$

with space complexity

$$s = O\left(\frac{\log\log m}{\varepsilon^2} \cdot \log\frac{1}{\delta} \right)$$

that is, doubly logarithmic *in* m.

Did we need to do that?

No

Second example: the Morris Counter, careful version (1/3)

Increment $C \leftarrow 2C$ w/p $\frac{1}{C}$

$\downarrow$

Instead $C \leftarrow (1+\beta)C$ w/p ~~p~~ p?

When $a_i = 1$

$$\mathbb{E}[C_{i+1} \mid C_i] = (1+\beta)C_i \cdot p + C_i(1-p)$$

$$= C_i(\beta p + 1)$$

$$= C_i + 1$$

($\uparrow$ want)

Second example: the Morris Counter, careful version (2/3)

Second example: the Morris Counter, careful version (3/3)

Theorem 41. *The "careful" version of* Morris Counter *is a* randomised *one-pass algorithm which, for any given parameters* $\varepsilon, \delta \in (0, 1]$, *provides an estimate* $\widehat{d}$ *of the number* d *of non-zero elements of the stream such that*

$$\Pr\left[(1-\varepsilon)d \leq \widehat{d} \leq (1+\varepsilon)d \right] \geq 1 - \delta$$

with space complexity

$$s = O\left(\log\log m + \log\frac{1}{\varepsilon} + \log\frac{1}{\delta} \right)$$

that is, doubly logarithmic in m and logarithmic in $1/\varepsilon$.

Third example: Distinct Elements

Third example: Distinct Elements, the Tidemark (AMS) algorithm (1/4)

Third example: Distinct Elements, the Tidemark (AMS) algorithm (2/4)

Third example: Distinct Elements, the Tidemark (AMS) algorithm (3/4)

Third example: Distinct Elements, the Tidemark (AMS) algorithm (4/4)

Third example: Distinct Elements, the Tidemark (AMS) algorithm (5/4?)

Theorem 42. *The (median trick version of the)* TIDEMARK *(AMS) algorithm is a* randomised *one-pass algorithm which, for any given parameter* $\delta \in (0,1]$, *provides an estimate* $\widehat{d}$ *of the number* d *of distinct elements of the stream such that, for some absolute constant* $C > 0$,

$$\Pr\left[\frac{1}{C} \cdot d \leq \widehat{d} \leq C \cdot d\right] \geq 1 - \delta$$

with space complexity

$$s = O\left(\log n \cdot \log \frac{1}{\delta}\right).$$

Can we do better?

Third example: Distinct Elements, the BJKST algorithm (1/4)

Third example: Distinct Elements, the BJKST algorithm (2/4)

Third example: Distinct Elements, the BJKST algorithm (4/4)

Theorem 43. *The (median trick version of the)* BJKST *algorithm is a* randomised *one-pass algorithm which, for any given parameters* $\varepsilon, \delta \in (0,1]$*, provides an estimate* $\widehat{d}$ *of the number* d *of distinct elements of the stream such that, for some absolute constant* $C > 0$*,*

$$\Pr\left[(1-\varepsilon) \cdot d \leq \widehat{d} \leq (1+\varepsilon) d \right] \geq 1 - \delta$$

with space complexity

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

… Can we do better?