# Interval Selection in the Streaming Model

Sergio Cabello, **Pablo Pérez-Lantero**

University of Ljubljana (Slovenia), **Universidad de Santiago, USACH (Chile)**

ADGO 2016

# Introduction

## Interval Selection in the Streaming Model

Given a stream $\mathbb{I}$ of intervals, compute within **one pass** over $\mathbb{I}$ a **maximum subset** of $\mathbb{I}$ of **independent** intervals (of cardinality $\alpha(\mathbb{I})$).

- Data stream model
  - widely used (Data Streams: Alg. & App., Muthukrishnan, 2005)
  - data arrives sequentially (not necessarily sorted)
  - bound in the amount of memory (e.g. polylog)
  - only access data of the past stored in the limited memory
  - $\Rightarrow$ approximate solutions in many cases

# Introduction

## Interval Selection in the Streaming Model

Given a stream $\mathbb{I}$ of intervals, compute within **one pass** over $\mathbb{I}$ a **maximum subset** of $\mathbb{I}$ of **independent** intervals (of cardinality $\alpha(\mathbb{I})$).

- Interval Selection $\equiv$ Maximum Independent Set in Interval Graphs
  - Fundamental optimization problem
  - Greedy algorithm in linear time (once intervals are sorted)
- Interval Selection in Data Stream:
  - 2-approximation in the Data Stream Model with $O(\alpha(\mathbb{I}))$ space:
    Emek et al (ICALP 2012); Cabello & Pérez-Lantero (2015)
  - No $(< 2)$-**approximation** can be obtained in **sublinear space**:
    Emek et al (ICALP 2012)
  - Generalizes the **distinct elements problem**:
    Given a data stream of numbers, identify how many distinct numbers are in the stream (Kane et al, PODS 2010)

# Introduction

## Interval Selection in the Streaming Model

Given a stream $\mathbb{I}$ of intervals, compute within **one pass** over $\mathbb{I}$ a **maximum subset** of $\mathbb{I}$ of **independent** intervals (of cardinality $\alpha(\mathbb{I})$).

We consider the **estimation** of $\alpha(\mathbb{I})$
(assuming that endpoints of intervals are in $[n] = \{1, 2, \ldots, n\}$)

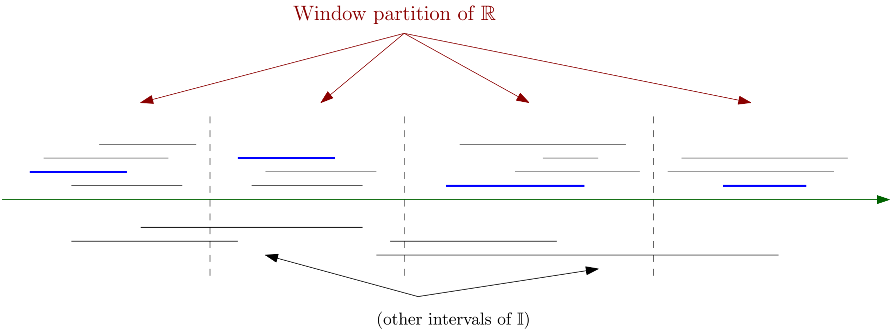
# Our results

1. ($(2+\varepsilon)$-approximation w.h.p.) An algorithm to compute $\hat{\alpha}(\mathbb{I})$ such that:
$$\left(\frac{1}{2}-\varepsilon\right)\alpha(\mathbb{I}) \;\le\; \hat{\alpha}(\mathbb{I}) \;\le\; \alpha(\mathbb{I})$$
with **probability** at least 2/3, in $O(\varepsilon^{-5}\log^6 n)$ space.

2. ($(3/2+\varepsilon)$-approximation w.h.p.) For **same-length** intervals, a computation of $\hat{\alpha}(\mathbb{I})$:
$$\left(\frac{2}{3}-\varepsilon\right)\alpha(\mathbb{I}) \;\le\; \hat{\alpha}(\mathbb{I}) \;\le\; \alpha(\mathbb{I})$$
with **probability** at least 2/3, in $O(\varepsilon^{-2}\log(1/\varepsilon)+\log n)$ space.

3. (Lower bounds) The approximation ratios for estimating $\alpha(\mathbb{I})$ are essentially optimal, if we use $o(n)$ bits of space.

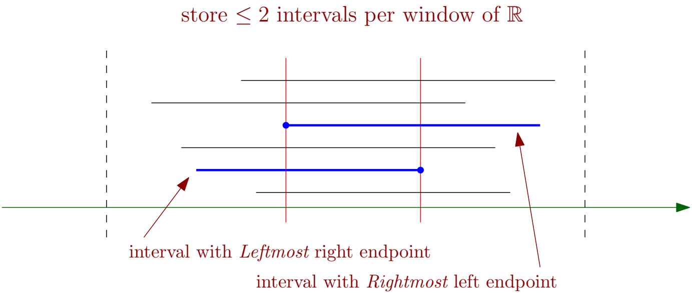
# A 2-approximation in $O(\alpha(\mathbb{I}))$ space
## (Cabello & Pérez-Lantero)




- Maintain a partition of $\mathbb{R}$ into **windows**
- For each window, all intervals from $\mathbb{I}$ contained in it are *pairwise-intertersecting*
- **Fact**: Since in the optimal solution no 2 intervals can fit within the same window, *taking one interval from each window* gives a 2-approximation

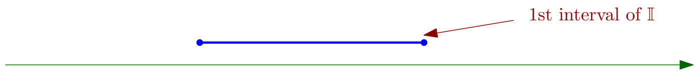
# A 2-approximation in $O(\alpha(\mathbb{I}))$ space

(Cabello & Pérez-Lantero)

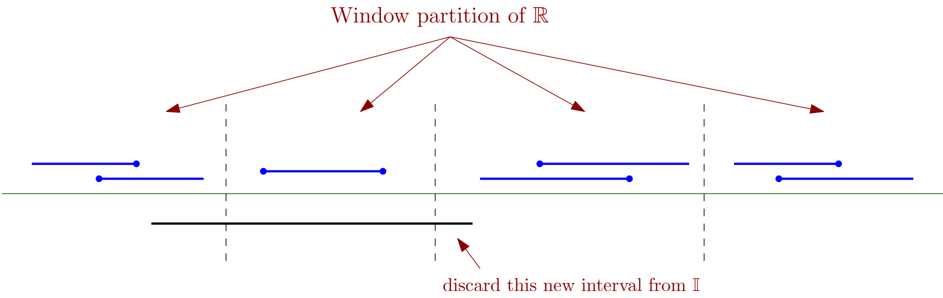


# A 2-approximation in $O(\alpha(\mathbb{I}))$ space
## (Cabello & Pérez-Lantero)

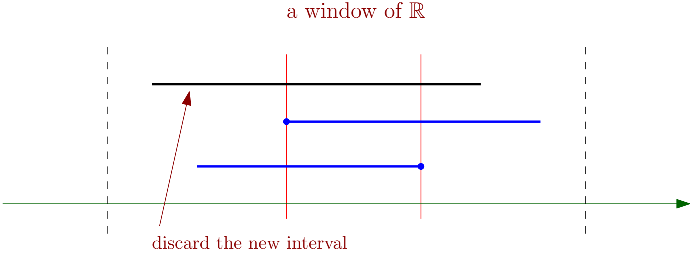
Initialization: one window, i.e. $\mathbb{R}$



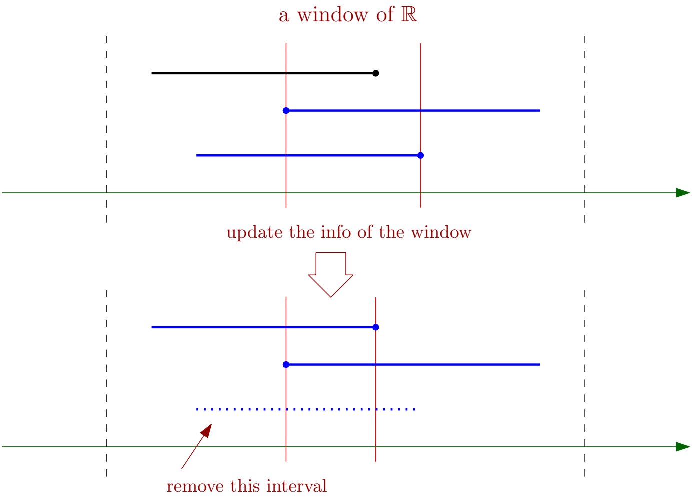
# A 2-approximation in $O(\alpha(\mathbb{I}))$ space
(Cabello & Pérez-Lantero)



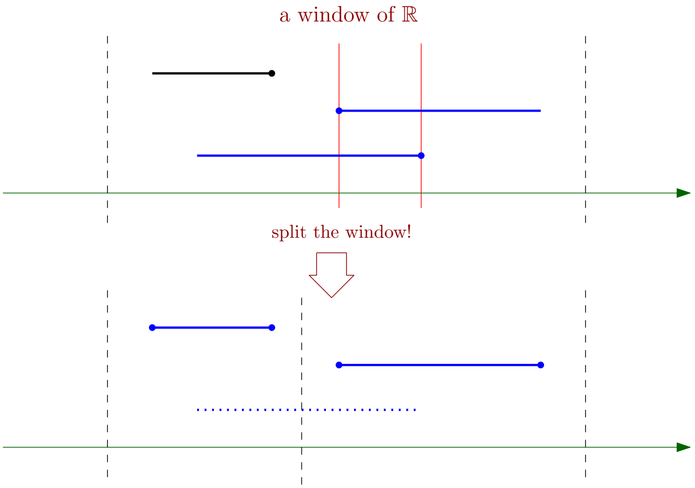
# A 2-approximation in $O(\alpha(\mathbb{I}))$ space
(Cabello & Pérez-Lantero)



# A 2-approximation in $O(\alpha(\mathbb{I}))$ space
(Cabello & Pérez-Lantero)

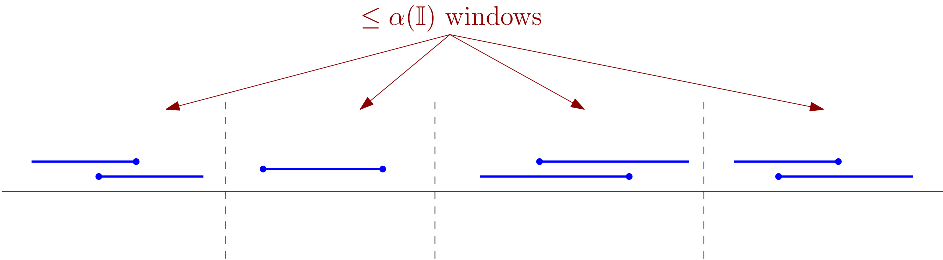


# A 2-approximation in $O(\alpha(\mathbb{I}))$ space
(Cabello & Pérez-Lantero)



# A 2-approximation in $O(\alpha(\mathbb{I}))$ space

(Cabello & Pérez-Lantero)




- the space is within $O(\alpha(\mathbb{I}))$
- each new interval is processed in $O(\log \alpha(\mathbb{I}))$ time

# Our assumptions for the estimation of $\alpha(\mathbb{I})$

1. Endpoints of intervals are in $[n] = \{1, 2, \dots, n\}$
2. A **unit** of memory can store a value from $[n] = \{1, 2, \dots, n\}$

# Sampling techniques

1. Suppose we have a stream $\mathbb{I}$ of numbers in $[n] = \{1, 2, \ldots, n\}$
2. Maintaining the **minimum** over the stream is **easy**
3. To maintain a (uniform) **random** element $s$ over the stream, we **would like** to have a (uniform & computable) **random permutation** $h : [n] \to [n]$:
   - $s$ = first element of $\mathbb{I}$.
   - **for each** new $a \in \mathbb{I}$: **if** $h(a) < h(s)$ **then** $s = a$.
4. The **sampled** element is chosen the **first time** it is seen
5. Problem: there is no compact way to encode a uniform-random permutation
6. Solution: construct $h$ using **hash** functions and **sacrifice** uniformity

# Sampling techniques

A family of permutations $\mathcal{H} = \{h : [n] \to [n]\}$ is $\varepsilon$-***min-wise independent*** if

$$\forall X \subseteq [n], y \in X : \quad \frac{1-\varepsilon}{|X|} \le \Pr_{h \in \mathcal{H}}[h(y) = \min h(X)] \le \frac{1+\varepsilon}{|X|}$$

For $X \subseteq [n]$, choosing $h \in \mathcal{H}$ uniform at random:

$$\arg\min\{h(x) \mid x \in X\}$$ is a **near-uniform** random element of $X$

# Sampling techniques

## Computable family of $\varepsilon$-**min-wise independent** permutations

For every $\varepsilon \in (0, 1/2)$ and $n > 0$, there exists a family $\mathcal{H}(n, \varepsilon) = \{h : [n] \to [n]\}$ of $\varepsilon$-**min-wise independent** permutations such that:

- a **random-uniform** element of $\mathcal{H}(n, \varepsilon)$ can be **chosen** in $O(\log(1/\varepsilon))$ time (**constructive**);
- for $h \in \mathcal{H}(n, \varepsilon)$ and $x, y \in [n]$, we can decide with $O(\log(1/\varepsilon))$ arithmetic operations whether $h(x) < h(y)$ (**computable**)

**Proof:**

Construct $K$-wise independent hash functions $[c \cdot n/\varepsilon] \to [c \cdot n/\varepsilon]$ for $K = \Theta(\log(1/\varepsilon))$ and some constant $c$.

(Indyk, 2001).

# Sampling techniques

How to generate a **near-uniform** random element of $X \subseteq [n] = \{1, 2, \ldots, n\}$?

1. Let $\mathcal{H} = \mathcal{H}(n, \varepsilon)$
2. Choose $h \in \mathcal{H}$ uniformly at random
3. **return** $s = \arg\min\{h(x) \mid x \in X\}$

[Datar and Muthukrishnan (ESA 2002)]

$\forall y \in Y \subseteq X \subseteq [n]$ : (**near-uniform** behavior)

$$\frac{(1-\varepsilon)|Y|}{|X|} \leq \Pr[s \in Y] \leq \frac{(1+\varepsilon)|Y|}{|X|}.$$

$$\frac{1-4\varepsilon}{|Y|} \leq \Pr[y = s \mid s \in Y] \leq \frac{1+4\varepsilon}{|Y|}.$$

# Sampling techniques

How to generate a **near-uniform** random element of $X \subseteq [n] = \{1, 2, \ldots, n\}$?

1. Let $\mathcal{H} = \mathcal{H}(n, \varepsilon)$
2. Choose $h \in \mathcal{H}$ uniformly at random
3. **return** $s = \arg\min\{h(x) \mid x \in X\}$

[Datar and Muthukrishnan (ESA 2002)]

$\forall y \in Y \subseteq X \subseteq [n]$ : (**near-uniform** behavior)

$$\frac{(1-\varepsilon)|Y|}{|X|} \leq \Pr[s \in Y] \leq \frac{(1+\varepsilon)|Y|}{|X|}.$$

$$\frac{1-4\varepsilon}{|Y|} \leq \Pr[y = s \mid s \in Y] \leq \frac{1+4\varepsilon}{|Y|}.$$

# Sampling techniques

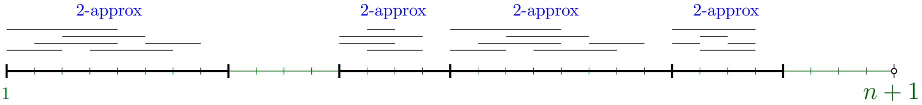
How to maintain a **near-uniform** random interval of the **stream** $\mathbb{I} = I_1, I_2, I_3, \ldots$?

1. Fix an easy-to-compute mapping $b : \mathbb{I} \to [n^2]$, e.g.

$$b([x, y]) \;=\; n(x-1) + y$$

2. Let $\mathcal{H} \;=\; \mathcal{H}(n^2, \varepsilon)$
3. Choose $h \in \mathcal{H}$ uniformly at random
   - $s \;=$ first interval of $\mathbb{I}$.
   - **for each** new interval $a \in \mathbb{I}$: **if** $h \circ b(a) < h \circ b(s)$ **then** $s \;=\; a$.

# Streaming algorithm (general idea)




- Find independent **canonical** segments in the window $[1, n] = [1, n+1)$
- Compute a 2-approximation within each **canonical** segment $S$:

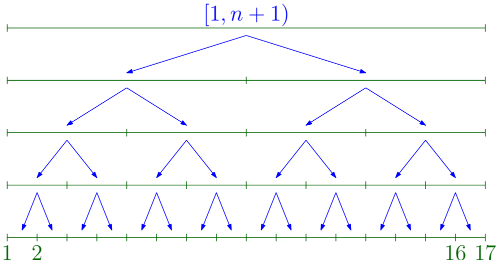
$$\text{in } O\big(\alpha(I \in \mathbb{I} \mid I \subset S)\big) \textbf{ space}$$

- Guarantee that each **canonical** segment $S$ contains **enough** disjoint intervals from $\mathbb{I}$, but **not too many** to save space

**Estimate**

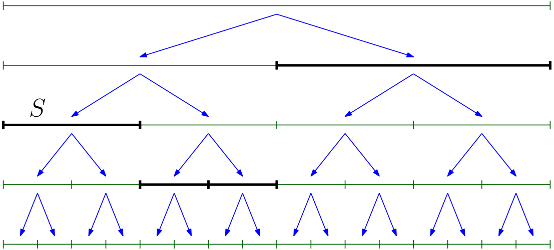
- the number of independent **canonical** segments
- the **average** of the 2-approximations of the segments

# Streaming algorithm (data structure)



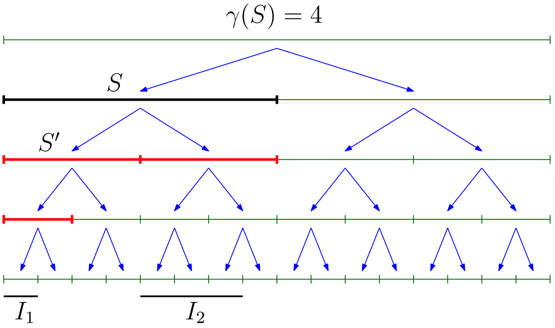

- **Canonical** segments $\mathbb{S}$ form a **segment tree** on $[i, i+1)$, $i \in [n]$
- $\pi(S)$ is the parent of segment $S \in \mathbb{S}$
- $\alpha[S] \ = \ \alpha(\{I \in \mathbb{I} \mid I \subset S\})$ (i.e. $\beta(S)$ in the paper)
- $\hat{\alpha}[S]$ is a 2-approximation of $\alpha[S]$ (i.e. $\hat{\beta}(S)$ in the paper)

# Streaming algorithm (data structure)




- $\hat{\alpha}[S]$ to know if $S$ has **enough**, and **not too many**, disjoint intervals is **not** "Ok"
- It may happen that $\hat{\alpha}[\pi(S)] < \hat{\alpha}[S]$, for some $S \in \mathbb{S}$ (counterintuitive!)
- We define a less-accurate but path-monotone and easy-to-compute estimator $\gamma(S)$:
  - $\gamma(S) \ \leq \ \gamma(\pi(S))$    (**path-monotone**)
  - $\alpha[S] \ \leq \ \gamma(S) \ \leq \ \alpha[S] \cdot \lceil \log n \rceil$   ($O(\log n)$-**approximation**)

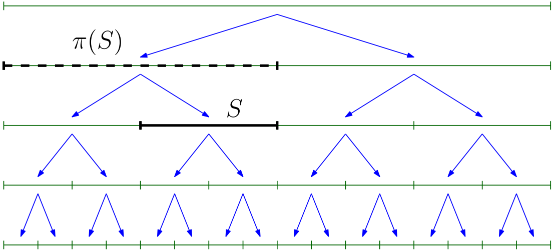
# Streaming algorithm (data structure)




$\gamma(S)$ is the (containment) number of **canonical** sub-intervals of $S$ containing an $I \in \mathbb{I}$:

- $\gamma(S) \ \le \ \gamma(\pi(S))$ **(path-monotone)**
- $\alpha[S] \ \le \ \gamma(S) \ \le \ \alpha[S] \cdot \lceil \log n \rceil$ ($O(\log n)$-**approximation**)

# Streaming algorithm (data structure)



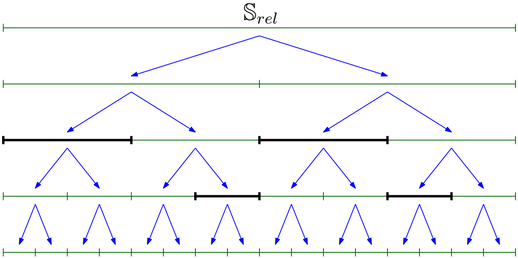

$S \in \mathbb{S}$ is ***relevant*** if

(i) $1 \leq \gamma(S) < 2\varepsilon^{-1}\lceil \log n \rceil^2$ (**not too many** disjoint intervals in $S$)

(ii) $\gamma(\pi(S)) \geq 2\varepsilon^{-1}\lceil \log n \rceil^2$ (**enough** disjoint intervals in $S$)

- $\mathbb{S}_{rel} \subset \mathbb{S}$ is the set of **relevant** segments, $N_{rel} = |\mathbb{S}_{rel}|$
- the **relevant** segments $\mathbb{S}_{rel}$ are **independent** (by definition)

# Streaming algorithm (data structure)

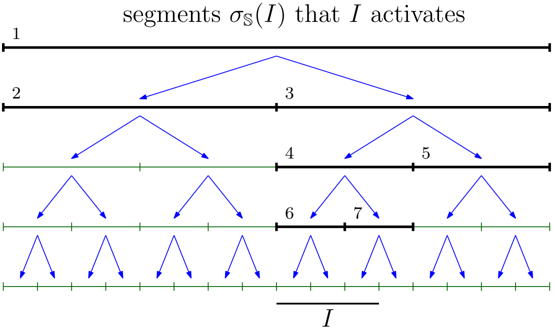



$$\left(\frac{1}{2} - \varepsilon\right) \alpha(\mathbb{I}) \; \le \; \sum_{S \in \mathbb{S}_{rel}} \hat{\alpha}[S] \; \le \; \alpha(\mathbb{I})$$

**Precise goal:** Estimate

- the number of **relevant** segments
- the **average** of the 2-approximations of the **relevant** segments

# Streaming algorithm (data structure)




$S \in \mathbb{S}$ is ***active*** if its *parent* $\pi(S)$ contains some $I \in \mathbb{I}$ (or $S = [1, n+1)$)

$N_{act}$ = number of **active** segments

$\sigma_{\mathbb{S}}(I)$ = stream of segments that $I$ activates

# Streaming algorithm (data structure)

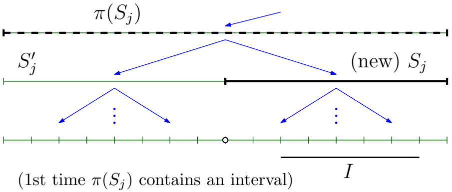
Maintaining a **near-uniform** random **active** segment $S_j$ in the **new** stream

$$\sigma \;=\; \sigma_{\mathbb{S}}(I_1), \sigma_{\mathbb{S}}(I_2), \sigma_{\mathbb{S}}(I_3), \sigma_{\mathbb{S}}(I_4), \dots \qquad O(\log n)\text{-times longer}$$

- Choose $h_j \in \mathcal{H}(n^2, \varepsilon)$ uniform at random and maintain the **active** segment

$$S_j \;=\; \arg\min\left\{ h_j(b(S)) \mid S \in \sigma \right\}$$

- If $S_j$ **changes**: $\gamma(S_j) \leftarrow \mathbf{1}$ if $I \subset S_j$ ($\mathbf{0}$ i.c.c.), $\gamma(\pi(S_j)) \leftarrow \mathbf{1} + \gamma(S_j) + \gamma(S_j')$, the part of $\sigma$ following $S_j$ has the information to compute $\gamma(\pi(S_j))$, $\gamma(S_j)$, and $\tilde{\alpha}[S_j]$




- By computing each of $\gamma(\pi(S_j))$ and $\gamma(S_j)$ **up to** $2\varepsilon^{-1}\lceil \log n \rceil^2$, we can **decide** at the end of $\mathbb{I}$ whether **the final** $S_j$ is **relevant**, in $O(\varepsilon^{-1} \log^2 n)$ space!

# Streaming algorithm (idea)

Given $\mathbb{I} = I_1, I_2, I_3, I_4, \ldots$, within **one pass** over the **new** $O(\log n)$-times-longer stream

$$\sigma \;=\; \sigma_{\mathbb{S}}(I_1), \sigma_{\mathbb{S}}(I_2), \sigma_{\mathbb{S}}(I_3), \sigma_{\mathbb{S}}(I_4), \ldots$$

1. Compute an estimator $\hat{N}_{act}$ of $N_{act}$
2. Compute an estimator $\hat{N}_{rel}$ of $N_{rel}$ (estimate $N_{rel}/N_{act}$, multiply by $\hat{N}_{act}$)
3. Compute an estimator $\hat{\rho}$ of

$$\rho \;=\; \left( \sum_{S \in \mathbb{S}_{rel}} \hat{\alpha}[S] \right) / N_{rel}$$

4. **return** $\hat{\alpha} = \hat{N}_{rel} \cdot \hat{\rho}$
5. Show that

$$\left(\frac{1}{2} - \varepsilon\right) \alpha(\mathbb{I}) \;\leq\; \hat{\alpha} \;\leq\; \alpha(\mathbb{I})$$

with probability at least $2/3$

1. **Goal**: Estimate the number $N_{act}$ of **distinct elements** in $\sigma$
2. Compute $\hat{N}_{act}$ using $O(\varepsilon^{-2} + \log|\mathbb{S}|) = O(\varepsilon^{-2} + \log n)$ space, which satisfies:

$$\Pr\left[(1-\varepsilon)N_{act} \le \hat{N}_{act} \le (1+\varepsilon)\cdot N_{act}\right] \ \ge \ \frac{11}{12}$$

(Kane et al, PODS 2010)

1. Sample $k = \Theta(\varepsilon^{-3} \log^2 n)$ **active** segments and count how many are **relevant**:
   - Maintain near-uniform random **active** segments $S_1, S_2, \ldots, S_k \in \sigma$
   - Count $X = |\{j \mid S_j \text{ is } \textbf{relevant}\}|$ for the final $S_1, S_2, \ldots, S_k$
   - Estimate $N_{rel}/N_{act}$ with $X/k$
   - **return** $\hat{N}_{rel} = \hat{N}_{act} \cdot \left(\frac{X}{k}\right)$

Analysis:

- $p = \Pr[S_j \text{ is } \textbf{relevant}] \in \left[\frac{(1-\varepsilon)N_{rel}}{N_{act}}, \frac{(1+\varepsilon)N_{rel}}{N_{act}}\right], \;\; p \geq 12/(k\varepsilon^2)$
- $X$ is the sum of $k$ i.i.d. random $\{0,1\}$-variables: $\mathbb{E}[X] = kp$
- $\Pr\Big[|X/k - p| \geq \varepsilon p\Big] \leq 1/12$ (Chebyshev's inequality)
- $\Big[|X/k - p| \leq \varepsilon p\Big]$ **AND** $\Big[|\hat{N}_{act} - N_{act}| \leq \varepsilon N_{act}\Big] \Longrightarrow \Big[|\hat{N}_{rel} - N_{rel}| \leq \varepsilon N_{rel}\Big]$

$$\Pr\Big[(1-\varepsilon)N_{rel} \leq \hat{N}_{rel} \leq (1+\varepsilon)\cdot N_{rel}\Big] \;\geq\; 10/12$$

1. Set $k = \Theta(\varepsilon^{-3}\log^2 n)$ and $k_0 = k\cdot\Theta(\varepsilon^{-1}\log^2 n) = \Theta(\varepsilon^{-4}\log^4 n) > k$
2. Maintain $k_0$ near-uniform random **active** segments $S_1, S_2, \ldots, S_{k_0} \in \sigma$:
   $\gamma(S_j)$, $\gamma(\pi(S_j))$, and $\hat{\alpha}[S_j]$ for each $j \in [1..k_0]$ in $O(\varepsilon^{-1}\log^2 n)$ space
3. For $X = \left|\{j \mid S_j \text{ is } \textbf{relevant}\}\right|$ and $p = \Pr[S_j \text{ is } \textbf{relevant}]$:

$$\mathbb{E}[X] = k_0 p, \quad \Pr\Big[|X - k_0 p| \geq k_0 p/2\Big] \leq \frac{1}{12}, \quad \text{and} \quad (1/2)k_0 p \geq k$$

4. $X \geq k$ with probability at least $11/12$
5. $S_1, S_2, \ldots, S_k$ are the **first** $k$ **relevant** segments of $S_1, S_2, \ldots, S_{k_0}$ (w.l.o.g.)
6. Compute $\hat{\rho} = \left(\sum_{j=1}^{k} \hat{\alpha}[S_j]\right)/k$, and using $1 \leq \hat{\alpha}[S_j] \leq \gamma(S_j) < 2\varepsilon^{-1}\lceil\log n\rceil^2$ and $Y_1 = \hat{\alpha}[S_1], Y_2 = \hat{\alpha}[S_2], \ldots, Y_k = \hat{\alpha}[S_k]$ are i.i.d. random variables:

$$\mathbb{E}[Y_j] \in \Big[(1-4\varepsilon)\rho, (1+4\varepsilon)\rho\Big] \quad \text{and} \quad \Pr\Big[|\hat{\rho} - \mathbb{E}[Y_j]| \geq \varepsilon\rho\Big] \leq \frac{1}{12}.$$

7. $\Pr\Big[(1-\varepsilon)\rho \;\leq\; \hat{\rho} \;\leq\; (1+\varepsilon)\rho\Big] \geq 10/12$

# Putting things together ..

With probability at least

$$1 - \frac{2}{12} - \frac{2}{12} \;=\; \frac{2}{3}$$

we have the events

$$\Big[ |N_{rel} - \hat{N}_{rel}| \le \varepsilon \cdot N_{rel} \Big] \quad \text{and} \quad \Big[ |\rho - \hat{\rho}| \le \varepsilon \rho \Big]$$

then, for $\hat{\alpha} = \hat{N}_{rel} \cdot \hat{\rho}$

$$\Pr\left[ \left( \frac{1}{2} - \varepsilon \right) \cdot \alpha(\mathbb{I}) \;\le\; \hat{\alpha} \;\le\; \alpha(\mathbb{I}) \right] \;\ge\; \frac{2}{3}.$$

# Lower bounds

Consider the estimation of $\alpha(\mathbb{I})$ for **same-length** intervals, $c > 0$

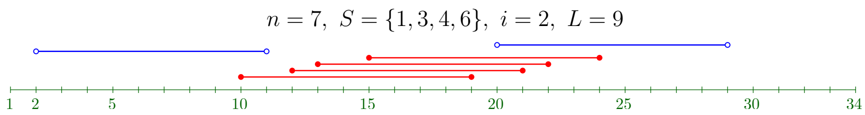
There is no algorithm that uses $o(n)$ bits of memory and computes an estimate $\hat{\alpha}$:

$$\Pr\left[\left(\frac{2}{3} + c\right)\alpha(\mathbb{I}) \le \hat{\alpha} \le \alpha(\mathbb{I})\right] \ \ge \ \frac{2}{3}$$

**Reduction** from the **one-way communication** of $\textsc{Index}(S, i)$: (Jayram et al, 2008)

- Alice knows a set $S \subseteq [n]$ and sends a message encoding $S$ to Bob
- Bob knows $i \in [n]$ and should determine from the message of Alice whether $i \in S$
- **Fact**: Alice's message must have $\Omega(n)$ bits in the worst case in order to Bob's answer is correct with probability $> 1/2$, say $\ge 2/3$.

# Reducing an instance of $\textsc{Index}(S, i)$




- Use intervals with endpoints in $[5n]$ for simplicity, and set $L = n + 2$
- Define the **streams** of intervals

$$\sigma_1(S) = [L + j, 2L + j] \text{ for } j \in S,\ \ \sigma_2(i) = (i, L + i), (2L + i, 3L + i)$$

- $\mathbb{I} = \sigma_1(S)\sigma_2(i)$, where $\alpha(\mathbb{I}) \in \{2, 3\}$ and $\alpha(\mathbb{I}) = 3$ iff $\textsc{Index}(S, i) = 1$
- **Algorithm** to estimate $\alpha(\mathbb{I})$:
  - Alice **simulates** the algorithm on $\sigma_1(S)$ and sends to Bob a **message** that encodes the **state of the memory** at the end.
  - Bob **continues** the simulation on the last two items of $\sigma_2(i)$: **return** 1 if $\hat{\alpha} > 2$, and 0 if $\hat{\alpha} \le 2$.
- $\Pr\left[(2/3 + c)\, \alpha(\mathbb{I}) \le \hat{\alpha} \le \alpha(\mathbb{I})\right] \ \ge\ 2/3 \ \Rightarrow\ \Pr\left[\text{Bob'answer is correct}\right] \ge 2/3$
- Alice's message (i.e. **space** of the algorithm) **cannot** be $o(n)$ bits

# Open problems

- We used the **cash register** model (intervals appear only). It is open to consider the **turnstile** model in which intervals can both appear and disappear.
- Approximate Maximum Independent Sets (MIS) of streaming ranges in the plane: **rectangles** and **squares**.
- Estimate the cardinalities of such MIS.

# The end

Thanks :)