Applied Algorithms

CSCI-B505 / INFO-I500

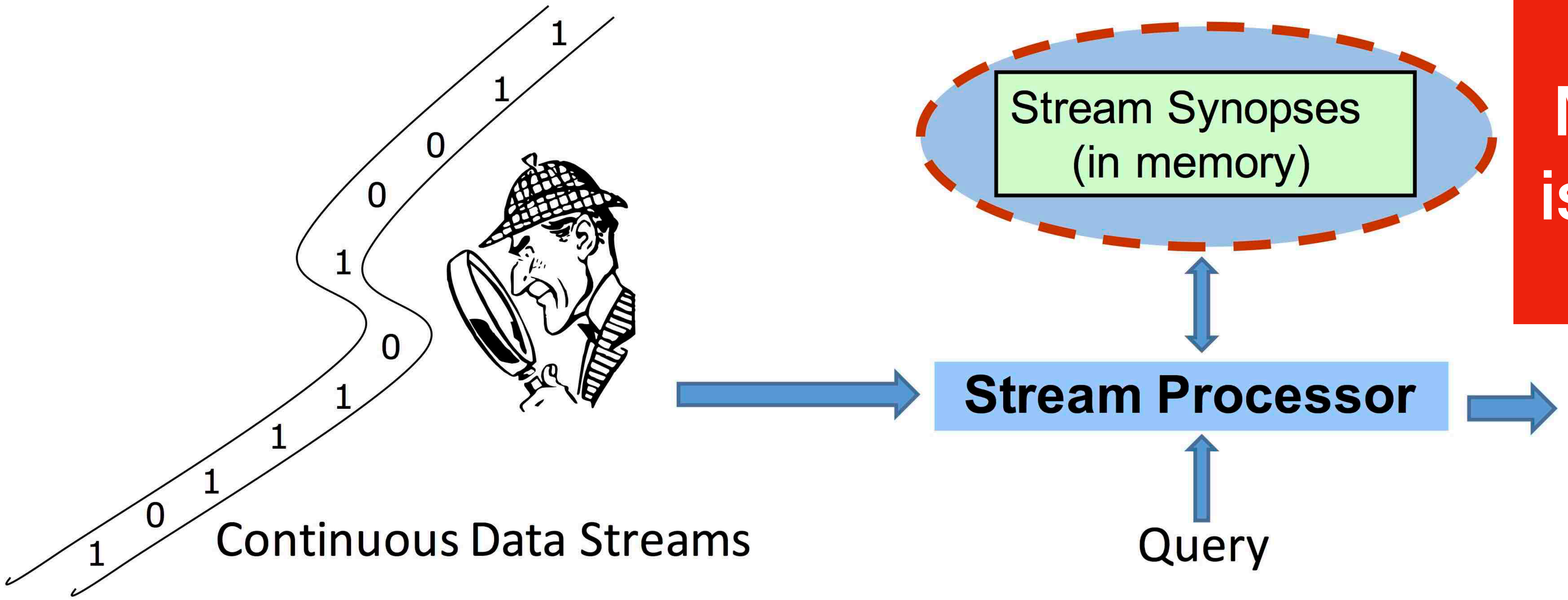
Lecture 23.

Algorithms on Streaming Data

M. Oguzhan Kulekci

- Streaming Data Model
- Random selection
- Finding heavy-hitters (Misra-Gries algorithm)

Streaming Model



Challenge:
Monitor real-world data that is streaming fast and huge in volume. Many examples…

$$x_0, x_1, x_2, \ldots$$

- Compute on-the-fly (only one-pass allowed)
- Limited memory allowance
- Should be fast
- Errors are unavoidable in general
  - Estimations with well control of error

Warm-up…

$$x_0, x_1, x_2, \ldots$$

$$\sum_{i=0}^{n} x_i \qquad \frac{1}{n}\sum_{i=0}^{n} x_i$$

$$max\{x_0, x_1, x_2, \ldots\}$$

$$min\{x_0, x_1, x_2, \ldots\}$$

- Sum, average, maximum, minimum are trivial cases
- The variance (σ^2) seems a bit tricky…

$$\sigma^2 = \frac{1}{n(n-1)}\left(n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right)$$

Two variables, one for the sum and one for the sum squares would be enough, but that may cause **numerical problems**

$M_1 = x_1$ and $S_1 = 0$

$M_k = M_{k-1} + (x_k - M_{k-1})/k$

$S_k = S_{k-1} + (x_k - M_{k-1})*(x_k - M_k)$

A better way is to maintain M_i and S_i values, and compute variance with $\sigma^2 = S_k/(k-1)$

Random selection problem

Reservoir Sampling: How can we select a random item among $x_1, x_2, \ldots x_n$ for any n such that probability of selecting the kth item is $1/n$.

Method: Sample the current item x_i with the correct probability p_i . What should be that p_i ?

$$x_1, x_2, x_3, \ldots, \boxed{x_i} \longrightarrow p_i = \frac{1}{i}$$

Why $p_i = 1/i$ works ?

Random selection problem

What is the probability that x_i will be selected among $x_1, x_2, \ldots, x_n$ at the end ?

1. Whatever happened is not important on previous $x_1, x_2, \ldots, x_{i-1}$
2. When x_i appeared it should get selected, which happens with probability $p_i = 1/i$.
3. All the remaining item $x_{i+1}, x_{i+2}, \ldots, x_n$ should NOT be selected after x_i

$$\frac{1}{i} \qquad 1 - \frac{1}{i+1} = \frac{i}{i+1}$$

$$x_1, x_2, x_3, \ldots, \boxed{x_i}, \boxed{x_{i+1}}, x_{i+2}, \ldots x_n$$

$$\frac{1}{i} \cdot \frac{i}{i+1} \cdot \frac{i+1}{i+2} \cdot \frac{i+2}{i+3} \cdots \cdot \frac{n-2}{n-1} \cdot \frac{n-1}{n} = \frac{\mathbf{1}}{\mathbf{n}}$$

Random selection problem

Select one element RANDOMLY from the streaming sequence S. Use the R values which are random values in range [1..100] for the probability generation such that, **If** $R_i \leq 100 \cdot p_i$ **perform sampling**, else proceed with the next item, where p_i is the required probability.

	1	2	3	4	5	6	7	8	9	10
S	9	7	10	2	5	4	8	6	3	4
Sampled	**9**	9	9	**2**	**5**	5	5	**6**	**3**	3
R	-	65	45	**21**	**15**	87	95	**10**	**2**	78

$21 < 100 \cdot \frac{1}{4}$ $15 < 100 \cdot \frac{1}{5}$ $10 < 100 \cdot \frac{1}{8}$ $2 < 100 \cdot \frac{1}{9}$

Random selection problem

Reservoir Sampling:

How can we select $\mathbf{k}$ random items among $x_1, x_2, \ldots x_n$ for any n ?

- Maintain a collection of k items selected.
- Place the first k items $x_1, x_2, \ldots, x_k$ directly to this collection.
- At each arrival of the x_i, $\;i > k$, decide whether to sample it with probability p_i.
- If positive, then we randomly replace one of the items in the collection with x_k

$$\boxed{s_1, s_2, s_3, \ldots, s_k}$$

What should be the p_i?

$$x_1, x_2, x_3, \ldots, \boxed{x_i} \longrightarrow$$

With probability p_i, , replace a random item from the current selection

Random selection problem

$$\boxed{s_1, s_2, s_3, \ldots, s_k}$$

$$x_1, x_2, x_3, \ldots, \boxed{x_i} \longrightarrow p_i = \frac{k}{i}$$

How can x_i appear in the selected items at the end ?

- What happened in $x_1, x_2, \ldots, x_{i-1}$ is not important
- The x_i should have been selected with p_i
- All the remaining items $x_{i+1}, x_{i+2}, \ldots, x_n$ are either not selected or not replaced with the x_i in the collection.

$$\frac{k}{i} \cdot \left(\left(1 - \frac{k}{i+1}\right) + \frac{k}{i+1} \cdot \frac{k-1}{k}\right) \cdot \ldots = \frac{k}{i} \cdot \frac{i}{i+1} \cdot \frac{i+1}{i+2} \cdots = \frac{k}{n}$$

Random selection problem

Select 3 items randomly from the streaming sequence S. Use R for probability assignment accordingly.

	1	2	3	4	5	6	7	8	9	10
S	9	7	10	2	5	4	8	6	3	4
Sampled-1	9	9	9	9	9	9	**8**	8	8	8
Sampled-2		7	7	**2**	2	2	2	**6**	6	6
Sampled-3			10	10	10	10	10	10	10	**4**
R	-	-	-	**21**	75	67	**35**	**17**	**73**	**15**
Replace	-	-	-	**2**	1	3	**1**	**2**	2	**3**

$$21 < 100 \cdot \frac{3}{4}$$

$$35 < 100 \cdot \frac{3}{7}$$

$$3 < 100 \cdot \frac{3}{10}$$

$$17 < 100 \cdot \frac{3}{8}$$

Finding heavy hitters

- The ϕ-HeavyHitters are those who appeared more than $\phi \cdot n$ times on the data stream $x_1, x_2, \ldots, x_n$

  For example: On S= {3,4,3,5,4,7,3,2,2,3}, the 0.3-HH is 3, 0.2-HHs are 2,3,4

- Find 0.3-HH keywords on a search engine, those keywords that appeared more than 0.3n times for any n value at the time of query.

- We can maintain frequency counters for each symbol and return the ones that comply with the query. The answer will be precise. However, we need large number of counters and managing them will be a mess when alphabet is large.

- **Would it be possible to allow a controlled error and achieve in less space ?**

Finding heavy hitters

- The Misra-Gries Algorithm
  - Returns all items that appeared more than $\phi \cdot n$
  - Never returns an item that appeared less than $(\phi - \epsilon) \cdot n$
  - Some of the items that appeared more than $(\phi - \epsilon) \cdot n$ but less than $\phi \cdot n$ may still be reported in the ϕ-HH list.

- While achieving those, it uses $O(1/\epsilon)$-space. So smaller the allowed error, larger the space, and vice versa

Finding heavy hitters

Algorithm 0.1: $\textsc{Misra-Gries}(k)$

$$
\begin{array}{l}
n \leftarrow 0; T \leftarrow \emptyset;\\
\textbf{for each } i :\\
\quad \textbf{do} \begin{cases}
n \leftarrow n+1;\\
\textbf{if } i \in T\\
\quad \textbf{then } c_i \leftarrow c_i + 1;\\
\quad \textbf{else if } |T| < k-1\\
\quad \textbf{then } \begin{cases} T \leftarrow T \cup \{i\};\\ c_i \leftarrow 1; \end{cases}\\
\quad \textbf{else for all } j \in T\\
\quad \textbf{do } \begin{cases} c_j \leftarrow c_j - 1;\\ \textbf{if } c_j = 0\\ \quad \textbf{then } T \leftarrow T \backslash \{j\}; \end{cases}
\end{cases}
\end{array}
$$

- Maintain **at most** $1/\epsilon$ number of counters.
- Each symbol is devoted to an **active** counter.
- When we need more counters at a point, and had already achieved maximum count, we decrease all counters by one and wipe out the ones that dropped to zero.
- At the end report the active counters and their symbols.

Finding heavy hitters

$$S = 1,1,2,3,4,5,1,1,1,5,3,3,1,1,2 \qquad |S| = 15,\ \ \epsilon = 0.25$$

$C_1 = 1$

$C_1 = 2$

$C_1 = 2$

$C_2 = 1$

$C_3 = 1$

$C_4 = 1$

Only 4 counters allowed, so we can't initiate c_5

$C_1 = 2$

$C_2 = 1$

$C_3 = 1$

$C_4 = 1$

Decrease all by 1

$C_1 = 1$

$C_2 = 0$

$C_3 = 0$

$C_4 = 0$

Remove counters with 0

$C_1 = 1$

$C_2 = 0$

$C_3 = 0$

$C_4 = 0$

Finding heavy hitters

$$S = 1,1,2,3,4,5,1,1,1,5,3,3,1,1,2 \qquad |S| = 15,\ \epsilon = 0.25$$

$C_1 = 6 \qquad C_5 = 1 \qquad C_3 = 2 \qquad C_2 = 1$

Return 0.35-HH values.

- (0.35-0.25)=0.1
- 0.1 x 15 = 1.5
- Return all counters whose value is larger than 1.5
- Answer C_1 and C_3, symbols 1 and 3.
  - Actually they appeared 7 and 3 times.
  - 1 is a correct 0.35-HH, where 3 is from the gray area

Why Misra-Gries Work ?

- When a counter decrement is required, we loose $1/\epsilon$ count on the counters plus the current item, which makes $(1/\epsilon + 1)$ total loss in count.
- We have n items, so the number of decrement cannot exceed

$$\leq n/(1 + 1/\epsilon) = \frac{\epsilon \cdot n}{1 + \epsilon} < \epsilon \cdot n$$

- This means on each counter, maximum error is $< \epsilon \cdot n$
- Since we return all counters whose value is $\geq (\phi - \epsilon) \cdot n$, we guarantee to report the ones that appeared $\phi \cdot n$ times, and not to return the ones that appeared less than $(\phi - \epsilon) \cdot n$ times.
- In between, is the gray area. Depending on the positions they may or may not be returned.

Reading assignment

- https://www.cs.dartmouth.edu/~ac/Teach/data-streams-lecnotes.pdf
- Check internet resources to learn more about the reservoir sampling, Misra-Gries, majority selection, frequent item detection, heavy-hitter detection…