CS6713: Scalable Algorithms for Data Analysis

Fahad Panolan and Rameshwar Pratap

Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India

20-Aug-2022

The course

- Contents:
  - Streaming algorithms (Frequency moments)
  - Sketching
  - Dimension reduction
  - Graph streaming/sketching
- Prerequisite: Undergraduate algorithms, Basics of probability and linear algebra

The course

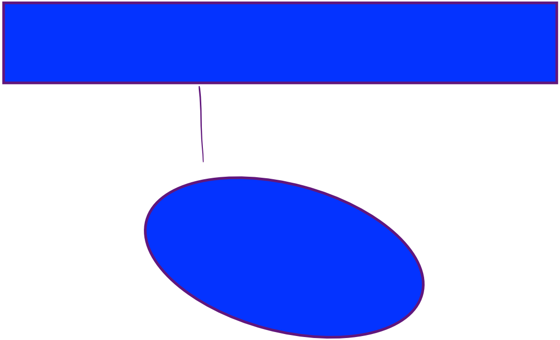
- Contents:
  - Streaming algorithms (Frequency moments)
  - Sketching
  - Dimension reduction
  - Graph streaming/sketching
- Prerequisite: Undergraduate algorithms, Basics of probability and linear algebra
- Evaluation
  - 35% : Theory assignments
  - 30% : Coding assignments
  - 35% : Mini project

Reference

- Data Stream algorithms, Lecture Notes, Amit Chakrabarti, 2020
- For basics of probability and randomized algorithms : Probability and Computing (2nd Edition) by Mitzenmacher and Upfal

Streaming Algorithms

Classical Algorithms: Random Access Model (RAM)



Classical Algorithms: Random Access Model (RAM)

Streaming Model

- The input consists of m objects/items/tokens $e_1, e_2, \ldots, e_m$ that are seen one by one by the algorithm.
- The algorithm has "limited" memory say for B tokens where $B < m$ (often $B << m$) and hence cannot store all the input
- Want to compute interesting functions over input

Classical Algorithms: Random Access Model (RAM)

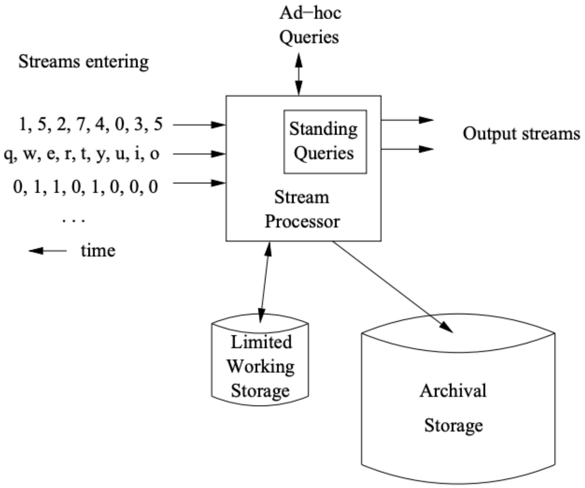
Streaming Model

- The input consists of m objects/items/tokens $e_1, e_2, \ldots, e_m$ that are seen one by one by the algorithm.
- The algorithm has "limited" memory say for B tokens where $B < m$ (often $B << m$) and hence cannot store all the input
- Want to compute interesting functions over input

Some examples:

- Each token is a number from $[n]$
- High-speed network switch: tokens are packets with source, destination IP addresses and message contents.
- Each token is an edge in graph (graph streams)
- Each token is a point in some feature space
- Each token is a row/column of a matrix

A data stream management system



[“Mining of Massive Data Sets” by Leskovec, Rajaraman, Ullman]

Streaming model: motivation/connections

- Very large but slow storage (tape, slow disk) that is suited for sequential access and fast main memory. Read data in one (or more) passes from slow medium.

- Scenarios such as network switches, sensors etc where huge amount of data is flying by and cannot be stored (due to cost or privacy/legal reasons) but one wants only high-level statistics.

- Distributed computing. Data stored in multiple machines. Cannot send all data to central location. Streaming algorithms can simulate a class of algorithms that exchange small amount of data.

Finding Majority Element

Given an array A of m integers, output an element that occurs more than $m/2$ times in A?

1,2,3,4,….,m/2-1,m/2-1,….

$O(m)$ $B << m$

1,2,3,…..m/2-2,1,1,1,1

5 2 6 3 5 1 5 4 5

Finding Majority Element

2, 3,3,1,2,1,1,1,1,5,6,6,6

Given an array A of m integers, output an element that occurs more than $m/2$ times in A?

Algorithm: 2, 3,3,1,2,1,1,1,1,5,1

- Initialize $c \leftarrow 0$ and $s = Null$
- **For** $i = 1$ to m
  - **If** $A[i] = s$, **then** $c \leftarrow c + 1$.
  - **If** $A[i] \neq s$ and $c > 0$, **then** $c \leftarrow c - 1$.
  - **If** $A[i] \neq s$ and $c = 0$, **then** $c \leftarrow 1$ and $s \leftarrow A[i]$.
- Check whether s is indeed the majority element and output accordingly.

i=1, s=2, c=1
i=2, s=2, c=0
i=3, s=3, c=1
i=4, s=3, c=0
i=5, s=2, c=1
i=6, s=2, c=0
i=7, s=1, c=1
i=8, s=1, c=2
i=9, s=1, c=3
i=10, s=1, c=2
i=11, s=1, c=3

2, 3,3,1,2,1,1,1,1,5,6 i=11,s=1,c=1

Analysis

Definition: For each element q in the array A, let f_q be the frequency of q.

Claim: If there is a majority element q, then algorithm outputs $s = q$ and $c \geq f_q - m/2$.

Analysis

Definition: For each element q in the array A, let f_q be the frequency of q.

Claim: If there is a majority element q, then algorithm outputs $s = q$ and $c \geq f_q - m/2$.

Proof:

- I_q: No. of times we increment c when we see q.
- D_q: No. of times we decrement c when we see q.
- I_0: No. of times we increment c when we see an element $\neq q$
- D_0: No. of times we decrement c when we see an element $\neq q$

Analysis

Definition: For each element q in the array A, let f_q be the frequency of q.

Claim: If there is a majority element q, then algorithm outputs $s = q$ and $c \geq f_q - m/2$.

Proof:

- I_q: No. of times we increment c when we see q.
- D_q: No. of times we decrement c when we see q.
- I_0: No. of times we increment c when we see an element $\neq q$
- D_0: No. of times we decrement c when we see an element $\neq q$

$$c = I_q + I_0 - (D_q + D_0)$$

Analysis

Definition: For each element q in the array A, let f_q be the frequency of q.

Claim: If there is a majority element q, then algorithm outputs $s = q$ and $c \geq f_q - m/2$.

Proof:

- I_q: No. of times we increment c when we see q.
- D_q: No. of times we decrement c when we see q.
- I_0: No. of times we increment c when we see an element $\neq q$
- D_0: No. of times we decrement c when we see an element $\neq q$

$$\begin{aligned} c &= I_q + I_0 - (D_q + D_0) \\ &\geq I_q + I_0 - \frac{m}{2} \end{aligned}$$

Analysis

Definition: For each element q in the array A, let f_q be the frequency of q.

Claim: If there is a majority element q, then algorithm outputs $s = q$ and $c \geq f_q - m/2$.

Proof:

- I_q: No. of times we increment c when we see q.
- D_q: No. of times we decrement c when we see q.
- I_0: No. of times we increment c when we see an element $\neq q$
- D_0: No. of times we decrement c when we see an element $\neq q$

$$\begin{aligned} c &= I_q + I_0 - (D_q + D_0) \\ &\geq I_q + I_0 - \frac{m}{2} \\ &\geq I_q + D_q - \frac{m}{2} \end{aligned}$$

Analysis

Definition: For each element q in the array A, let f_q be the frequency of q.

Claim: If there is a majority element q, then algorithm outputs $s = q$ and $c \geq f_q - m/2$.

Proof:

- I_q: No. of times we increment c when we see q.
- D_q: No. of times we decrement c when we see q.
- I_0: No. of times we increment c when we see an element $\neq q$
- D_0: No. of times we decrement c when we see an element $\neq q$

$$
\begin{aligned}
c &= I_q + I_0 - (D_q + D_0) \\
&\geq I_q + I_0 - \frac{m}{2} \\
&\geq I_q + D_q - \frac{m}{2} \overset{=}{\geq} f_q - \frac{m}{2}
\end{aligned}
$$

Heavy Hitters Problem and Misra-Gries Algorithm

Heavy Hitters Problem: Find all elements i such that $f_i > m/k$.

Thank You.