CS6713: Scalable Algorithms for Data Analysis

Fahad Panolan

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

27-Aug-2022

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)

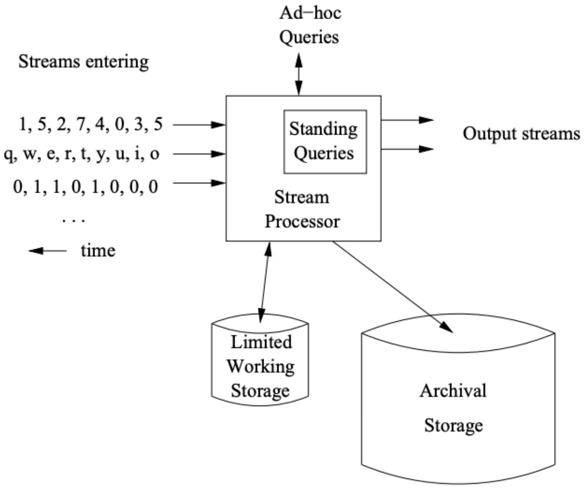
Streaming Model

- 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]

Finding Majority Element

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

Finding Majority Element

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

Algorithm:

- 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.

Heavy Hitters Problem and Misra-Gries Algorithm

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

Heavy Hitters Problem and Misra-Gries Algorithm

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

- Initialize $D[1, \ldots, k] \leftarrow 0$ and $S[1, \ldots, k] = Null$
- **For** $j = 1$ to m
  - **If** $A[j] = S[r]$ for some r, **then** $D[r] \leftarrow D[r] + 1$.
  - **Else If** $S[r] = Null$ for some r, **then** $D[r] \leftarrow 1$ and $S[r] \leftarrow A[j]$.
  - **Else:** for all $\ell \in [k]$, $D[\ell] \leftarrow D[\ell] - 1$.
  - Remove elements from S whose counter values are 0
- Verify whether elements S have frequency at least m/k.

Analysis

Space usage $O(k \log m)$ bits.

Analysis

Space usage $O(k \log m)$ bits.

- For each $r \in [k]$, $\widehat{f}_{S[r]} = D[r]$.
- For all other elements i, set $\hat{f}_i = 0$.

Analysis

Space usage $O(k \log m)$ bits.

- For each $r \in [k]$, $\widehat{f}_{S[r]} = D[r]$.
- For all other elements i, set $\hat{f}_i = 0$.

Theorem

For each element i in A: $f_i - \frac{m}{k+1} \leq \hat{f}_i \leq f_i$.

Analysis

Space usage $O(k \log m)$ bits.

- For each $r \in [k]$, $\widehat{f}_{S[r]} = D[r]$.
- For all other elements i, set $\hat{f}_i = 0$.

Theorem

For each element i in A: $f_i - \frac{m}{k+1} \leq \hat{f}_i \leq f_i$.

Corollary

Any item with $f_i > \frac{m}{k}$ is in D at the end of the algorithm.

Proof of Correctness

Theorem

For each element i *in* A*:* $f_i - \frac{m}{k+1} \leq \hat{f}_i \leq f_i$.

Proof of Correctness

Theorem

For each element i in A: $f_i - \frac{m}{k+1} \leq \hat{f}_i \leq f_i$.

Easy to see: $\hat{f}_i \leq f_i$. Why?

Proof of Correctness

Theorem

For each element i in A: $f_i - \frac{m}{k+1} \leq \hat{f}_i \leq f_i$.

Easy to see: $\hat{f}_i \leq f_i$. Why?

Alternative view of algorithm:

- Assume $A \subseteq \{1, \ldots, n\}$ for some n and let $a_j = A[j]$
- Maintain counts $C[i]$ for each $i \in \{1, \ldots, n\}$ (initialized to 0). Only k are non-zero at any time.
- During the jth iteration
  - **If** $C[a_j] > 0$ then increment $C[a_j]$ by one.
  - **ElseIf:** less than k positive counters then set $C[a_j] = 1$
  - **Else:** decrement all positive counters (exactly k of them)
- Output $\hat{f}_i = C[i]$ for each i

Proof of Correctness

Want to show: $f_i - \hat{f}_i \leq \frac{m}{(k+1)}$

Proof of Correctness

Want to show: $f_i - \hat{f}_i \leq \frac{m}{(k+1)}$

- Suppose we have ℓ occurrences of k counters being decremented.

Proof of Correctness

Want to show: $f_i - \hat{f}_i \leq \frac{m}{(k+1)}$

- Suppose we have ℓ occurrences of k counters being decremented. Then $\ell k + \ell \leq m$ which implies $\ell \leq \frac{m}{(k+1)}$.

- Consider $\alpha = (f_i - \hat{f}_i)$ during the execution of the algorithms. Initially 0. Initially $\ell = 0$. How big α can be?

Proof of Correctness

Want to show: $f_i - \hat{f}_i \leq \frac{m}{(k+1)}$

- Suppose we have ℓ occurrences of k counters being decremented. Then $\ell k + \ell \leq m$ which implies $\ell \leq \frac{m}{(k+1)}$.
- Consider $\alpha = (f_i - \hat{f}_i)$ during the execution of the algorithms. Initially 0. Initially $\ell = 0$. How big α can be?
    - If $a_j = i$ and $C[i]$ is incremented α stays same

Proof of Correctness

Want to show: $f_i - \hat{f}_i \leq \frac{m}{(k+1)}$

- Suppose we have ℓ occurrences of k counters being decremented. Then $\ell k + \ell \leq m$ which implies $\ell \leq \frac{m}{(k+1)}$.
- Consider $\alpha = (f_i - \hat{f}_i)$ during the execution of the algorithms. Initially 0. Initially $\ell = 0$. How big α can be?
  - If $a_j = i$ and $C[i]$ is incremented α stays same
  - If $a_j = i$ and $C[i]$ is not incremented then α increases by one and k counters decremented — charge to ℓ

Proof of Correctness

Want to show: $f_i - \hat{f}_i \leq \frac{m}{(k+1)}$

- Suppose we have ℓ occurrences of k counters being decremented. Then $\ell k + \ell \leq m$ which implies $\ell \leq \frac{m}{(k+1)}$.

- Consider $\alpha = (f_i - \hat{f}_i)$ during the execution of the algorithms. Initially 0. Initially $\ell = 0$. How big α can be?

  - If $a_j = i$ and $C[i]$ is incremented α stays same
  - If $a_j = i$ and $C[i]$ is not incremented then α increases by one and k counters decremented — charge to ℓ
  - If $a_j \neq i$ and $C[i]$ is decremented. Then α increases by 1 it is because $C[i]$ is decremented — charge to ℓ

Proof of Correctness

Want to show: $f_i - \hat{f}_i \leq \frac{m}{(k+1)}$

- Suppose we have ℓ occurrences of k counters being decremented. Then $\ell k + \ell \leq m$ which implies $\ell \leq \frac{m}{(k+1)}$.

- Consider $\alpha = (f_i - \hat{f}_i)$ during the execution of the algorithms. Initially 0. Initially $\ell = 0$. How big α can be?

  - If $a_j = i$ and $C[i]$ is incremented α stays same
  - If $a_j = i$ and $C[i]$ is not incremented then α increases by one and k counters decremented — charge to ℓ
  - If $a_j \neq i$ and $C[i]$ is decremented. Then α increases by 1 it is because $C[i]$ is decremented — charge to ℓ

- Hence total number of times α increases is at most ℓ.

Thank You.