Optimal Sampling Algorithms for Frequency Estimation in Distributed Data

Zengfeng Huang Ke Yi Yunhao Liu

HKUST

Guihai Chen

Shanghai Jiaotong University

INFOCOM 2011

Preliminaries

- Massive data
  - Impractical or impossible to store in a single machine

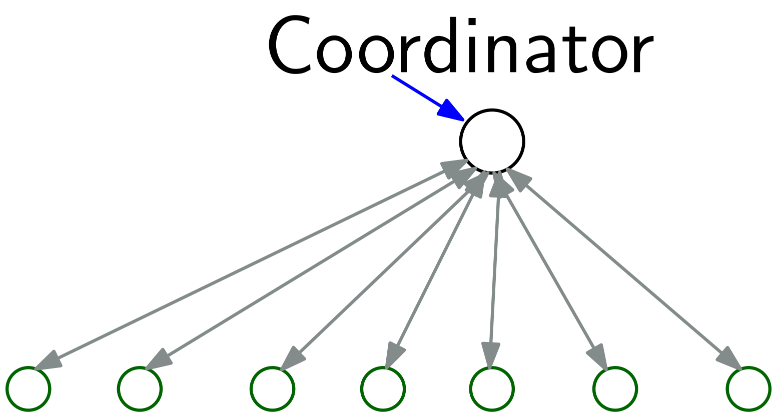
Preliminaries

- Massive data
  - Impractical or impossible to store in a single machine
- Large distributed system
  - Sensor networks, distributed databases, data centers, etc.

Preliminaries

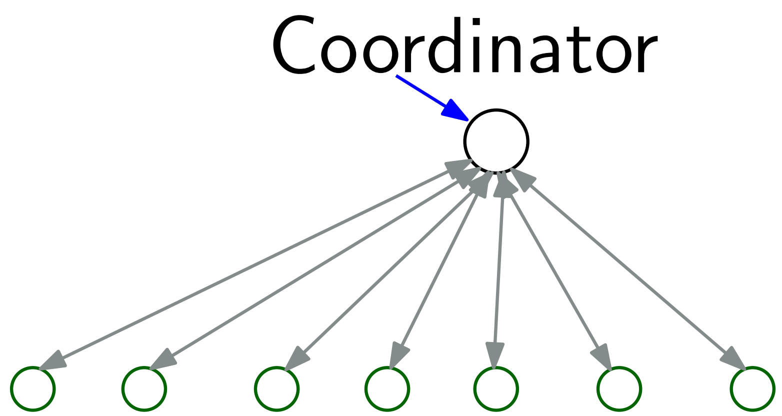
- Massive data
  - Impractical or impossible to store in a single machine
- Large distributed system
  - Sensor networks, distributed databases, data centers, etc.
- Communication bandwidth: most valuable resource

Preliminaries



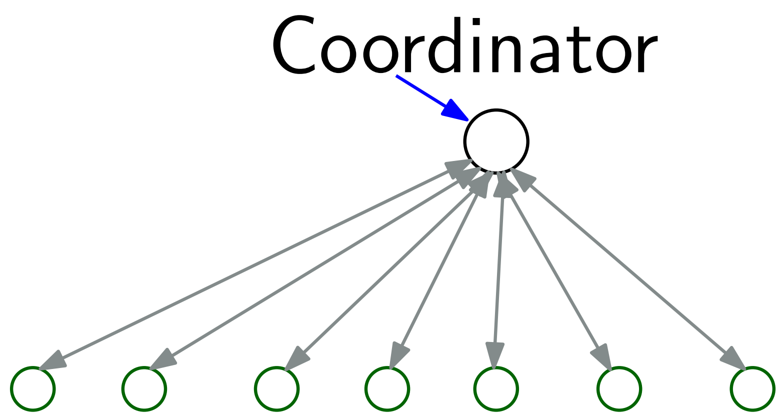
- Model
  - Coordinator
    - To computing some function of the Data

Preliminaries



- Model
  - Coordinator
    - To computing some function of the Data
  - Data distributed on n nodes
    - Nodes communicate with the Coordinator

Preliminaries



- Model
  - Coordinator
    - To computing some function of the Data
  - Data distributed on n nodes
    - Nodes communicate with the Coordinator
  - Communication-efficiently

Preliminaries

- Frequency Estimation
  - Input: Multiset S of N items drawn from the universe $[1 \dots u]$

Preliminaries

- Frequency Estimation
  - Input: Multiset S of N items drawn from the universe $[1 \ldots u]$
  - Each node $j \in [n]$ holds a subset of S

Preliminaries

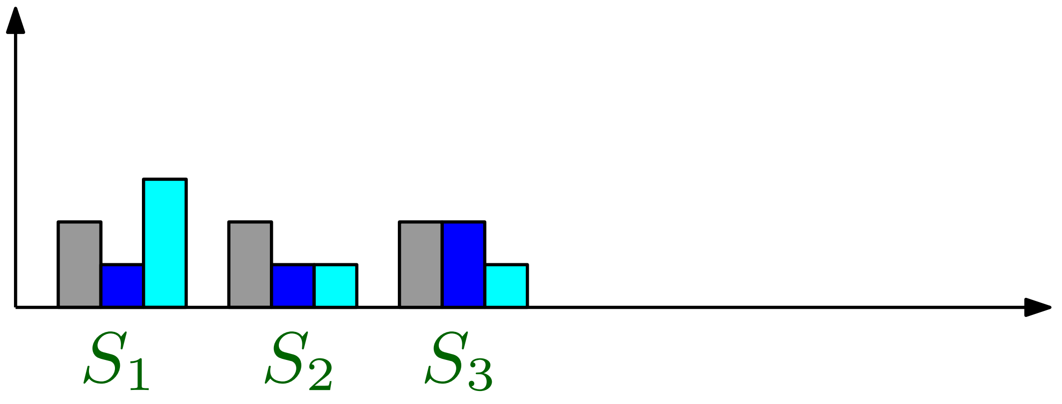
- Frequency Estimation
  - Input: Multiset S of N items drawn from the universe $[1 \ldots u]$
  - Each node $j \in [n]$ holds a subset of S
    - For any item $i \in [u]$
      x_{ij}: total number of i's in node j (local count)
      $y_i = \sum_{j=1}^{n} x_{ij}$ (global count)

Preliminaries

- Frequency Estimation
  - Input: Multiset S of N items drawn from the universe $[1 \ldots u]$
  - Each node $j \in [n]$ holds a subset of S
    - For any item $i \in [u]$
      x_{ij}: total number of i's in node j (local count)
      $y_i = \sum_{j=1}^{n} x_{ij}$ (global count)
  - Compute y_i for each i

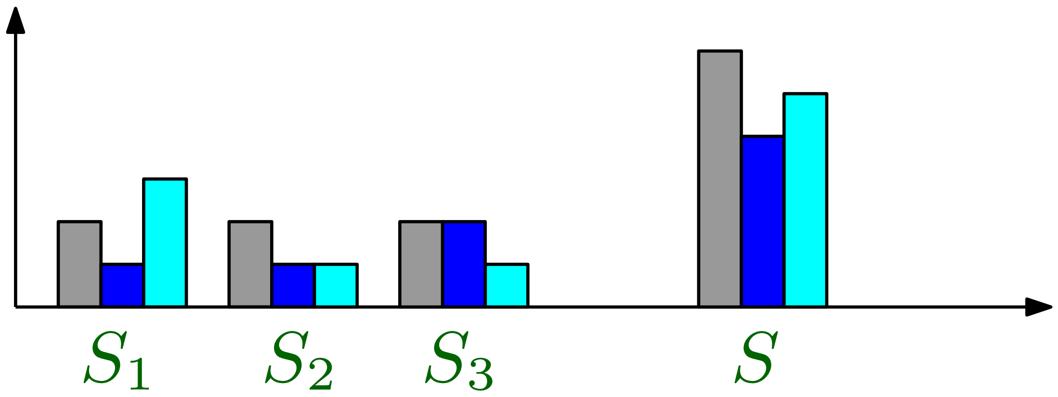
Preliminaries

- Frequency Estimation
  - Input: Multiset S of N items drawn from the universe $[1 \ldots u]$
  - Each node $j \in [n]$ holds a subset of S
    - For any item $i \in [u]$
      x_{ij}: total number of i's in node j (local count)
      $y_i = \sum_{j=1}^{n} x_{ij}$ (global count)
  - Compute y_i for each i



Preliminaries

- Frequency Estimation
  - Input: Multiset S of N items drawn from the universe $[1 \ldots u]$
  - Each node $j \in [n]$ holds a subset of S
    - For any item $i \in [u]$
      x_{ij}: total number of i's in node j (local count)
      $y_i = \sum_{j=1}^{n} x_{ij}$ (global count)
  - Compute y_i for each i



Distributed Algorithm

- Compute exactly: send everything
- Approximate each y_i within addtive error ϵN

Distributed Algorithm

- Compute exactly: send everything
- Approximate each y_i within addtive error ϵN

- Sketch: Each node computes a sketch of its own data and sends it to the coordinator.
  - *Count-min, Space saving, etc.*

Distributed Algorithm

- Compute exactly: send everything
- Approximate each y_i within addtive error ϵN

- Sketch: Each node computes a sketch of its own data and sends it to the coordinator.
  - *Count-min, Space saving, etc.*
  - Sketch size: $O(1/\varepsilon)$

Distributed Algorithm

- Compute exactly: send everything
- Approximate each y_i within addtive error ϵN

- Sketch: Each node computes a sketch of its own data and sends it to the coordinator.
  - *Count-min, Space saving, etc.*
  - Sketch size: $O(1/\varepsilon)$
  - Communication cost: $O(n/\varepsilon)$

Distributed Algorithm

- Compute exactly: send everything
- Approximate each y_i within addtive error ϵN

- Sketch: Each node computes a sketch of its own data and sends it to the coordinator.
  - *Count-min, Space saving, etc.*
  - Sketch size: $O(1/\varepsilon)$
  - Communication cost: $O(n/\varepsilon)$

- Random sampling

Distributed Algorithm

- Compute exactly: send everything
- Approximate each y_i within addtive error ϵN

- Sketch: Each node computes a sketch of its own data and sends it to the coordinator.
  - *Count-min, Space saving, etc.*
  - Sketch size: $O(1/\varepsilon)$
  - Communication cost: $O(n/\varepsilon)$

- Random sampling
  - Uniformly randomly sample a subset of size $O(1/\varepsilon^2)$

Distributed Algorithm

- Compute exactly: send everything
- Approximate each y_i within addtive error ϵN

- Sketch: Each node computes a sketch of its own data and sends it to the coordinator.
  - *Count-min, Space saving, etc.*
  - Sketch size: $O(1/\varepsilon)$
  - Communication cost: $O(n/\varepsilon)$

- Random sampling
  - Uniformly randomly sample a subset of size $O(1/\varepsilon^2)$
  - Communication cost: $O(n + 1/\varepsilon^2)$

Distributed Algorithm

- Our result: $O(n + \frac{\sqrt{n}}{\varepsilon})$

Distributed Algorithm

- Our result: $O(n + \frac{\sqrt{n}}{\varepsilon})$
  - Strictly better than $O(\frac{n}{\varepsilon})$ and $O(n + \frac{1}{\varepsilon^2})$

Distributed Algorithm

- Our result: $O(n + \frac{\sqrt{n}}{\varepsilon})$
  - Strictly better than $O(\frac{n}{\varepsilon})$ and $O(n + \frac{1}{\varepsilon^2})$
- We assume $n \leq \frac{1}{\varepsilon^2}$
  - $n \approx \frac{1}{\varepsilon}$ in practice. example: $n = 1000$ and $\varepsilon = 0.001$.
  - Not theoretically interesting: if $n > \frac{1}{\varepsilon^2}$, the cost is dominated by n, and $\Omega(n)$ is a lower bound.

HT estimator [Horvitz and Thompson 56]

x_{ij}: local count of i at node j

- Sample each item randomly. if i is sampled, sends (i, x_{ij})
- The probability is a function of x_{ij}
  - Let $g : \mathbb{N} \rightarrow [0, 1]$ be the sampling function

HT estimator [Horvitz and Thompson 56]

x_{ij}: local count of i at node j

- Sample each item randomly. if i is sampled, sends (i, x_{ij})
- The probability is a function of x_{ij}
  - Let $g : \mathbb{N} \rightarrow [0,1]$ be the sampling function

HT estimator for x_{ij}:

$$Y_{i,j} = \frac{x_{i,j}}{g(x_{i,j})} \text{ if it is sampled, otherwise } 0$$

HT estimator [Horvitz and Thompson 56]

x_{ij}: local count of i at node j

- Sample each item randomly. if i is sampled, sends (i, x_{ij})
- The probability is a function of x_{ij}
  - Let $g : \mathbb{N} \rightarrow [0, 1]$ be the sampling function

HT estimator for x_{ij}:

$$Y_{i,j} = \frac{x_{i,j}}{g(x_{i,j})} \text{ if it is sampled, otherwise } 0$$

Estimator for y_i:

$$Y_i = Y_{i,1} + \cdots + Y_{i,n}$$

Variance of the HT estimator

Variance of the HT estimator

$$\mathrm{Var}[Y_{i,j}] = (\frac{x_{i,j}}{g(x_{i,j})} - x_{i,j})^2 g(x_{i,j}) + (x_{i,j})^2(1 - g(x_{i,j}))$$

$$= \frac{x_{i,j}^2(1-g(x_{i,j}))}{g(x_{i,j})}$$

Variance of the HT estimator

$$\mathrm{Var}[Y_{i,j}] = (\tfrac{x_{i,j}}{g(x_{i,j})} - x_{i,j})^2 g(x_{i,j}) + (x_{i,j})^2(1 - g(x_{i,j}))$$

$$= \tfrac{x_{i,j}^2(1-g(x_{i,j}))}{g(x_{i,j})}$$

Estimator for item i

$$Y_i = Y_{i,1} + \cdots + Y_{i,n}$$

$$\mathrm{Var}[Y_i] = \textstyle\sum_{j=1}^n \mathrm{Var}[Y_{ij}] = \sum_{j=1}^n \tfrac{x_{i,j}^2(1-g(x_{i,j}))}{g(x_{i,j})}$$

Sampling Function

- Question: What sampling function $g(x)$ should we use

Sampling Function

- Question: What sampling function $g(x)$ should we use
  - Accuracy: standard deviation less than εN

Sampling Function

- Question: What sampling function $g(x)$ should we use
  - Accuracy: standard deviation less than εN

    A function is valid, if $\mathrm{Var}[Y_i] \leq (\epsilon N)^2$ for all item i

Sampling Function

- Question: What sampling function $g(x)$ should we use
  - Accuracy: standard deviation less than εN

    A function is valid, if $\mathrm{Var}[Y_i] \leq (\epsilon N)^2$ for all item i
  - Communication cost: $\sum_{i,j} g(x_{ij})$

Sampling Function

- Question: What sampling function $g(x)$ should we use
  - Accuracy: standard deviation less than εN

    A function is valid, if $\mathrm{Var}[Y_i] \leq (\epsilon N)^2$ for all item i
  - Communication cost: $\sum_{i,j} g(x_{ij})$

Optimal valid $g(x)$?

A worst case optimal Sampling Function

- $g_1(x) = \frac{\sqrt{n}}{\varepsilon N} x$ ($g_1(x) = 1$ if $\frac{\sqrt{n}}{\varepsilon N} x > 1$)

A worst case optimal Sampling Function

- $g_1(x) = \frac{\sqrt{n}}{\varepsilon N} x$ ($g_1(x) = 1$ if $\frac{\sqrt{n}}{\varepsilon N} x > 1$)

$$
\begin{aligned}
\mathrm{Var}[Y_i] &= \sum_{j=1}^{n} \frac{x_{i,j}^2 (1 - x_{i,j}\sqrt{n}/\varepsilon N)}{x_{i,j}\sqrt{n}/\varepsilon N} \\
&\le \frac{\varepsilon N}{\sqrt{n}} y_i - \frac{1}{n} y_i^2 \\
&= -\left(\frac{y_i}{\sqrt{n}} - \frac{\varepsilon N}{2}\right)^2 + \frac{(\varepsilon N)^2}{4} \le \frac{1}{4}(\varepsilon N)^2.
\end{aligned}
$$

A worst case optimal Sampling Function

- $g_1(x) = \frac{\sqrt{n}}{\varepsilon N} x$ ($g_1(x) = 1$ if $\frac{\sqrt{n}}{\varepsilon N} x > 1$)

$$\begin{aligned}
\mathrm{Var}[Y_i] &= \sum_{j=1}^{n} \frac{x_{i,j}^2 (1 - x_{i,j}\sqrt{n}/\varepsilon N)}{x_{i,j}\sqrt{n}/\varepsilon N} \\
&\leq \frac{\varepsilon N}{\sqrt{n}} y_i - \frac{1}{n} y_i^2 \\
&= -\left(\frac{y_i}{\sqrt{n}} - \frac{\varepsilon N}{2}\right)^2 + \frac{(\varepsilon N)^2}{4} \leq \frac{1}{4}(\varepsilon N)^2.
\end{aligned}$$

- Communication cost: $\sum_{i,j} g_1(x_{ij}) = O(\frac{\sqrt{n}}{\varepsilon})$

A worst case optimal Sampling Function

- *Theorem: any valid sampling function has cost* $\Omega(\sqrt{n}/\varepsilon)$ *on some input.*

A worst case optimal Sampling Function

- *Theorem: any valid sampling function has cost $\Omega(\sqrt{n}/\varepsilon)$ on some input.*

Hard Input:

$$y_i = \varepsilon\sqrt{n}N \leq N \ (n \leq \frac{1}{\varepsilon^2}) \text{ for } 1 \leq i \leq \frac{1}{\varepsilon\sqrt{n}}$$

$$x_{i,1} = x_{i,2} = \cdots = x_{i,n} = \frac{\varepsilon N}{\sqrt{n}}$$

A worst case optimal Sampling Function

- *Theorem: any valid sampling function has cost $\Omega(\sqrt{n}/\varepsilon)$ on some input.*

Hard Input:

$$y_i = \varepsilon\sqrt{n}N \leq N \; (n \leq \tfrac{1}{\varepsilon^2}) \text{ for } 1 \leq i \leq \frac{1}{\varepsilon\sqrt{n}}$$

$$x_{i,1} = x_{i,2} = \cdots = x_{i,n} = \frac{\varepsilon N}{\sqrt{n}}$$

The total number of local counts is $\frac{\sqrt{n}}{\varepsilon}$

A worst case optimal Sampling Function

$$
\begin{aligned}
\mathrm{Var}[Y_i] &= \sum_{j=1}^{n} \frac{x_{i,j}^2 (1 - g(x_{i,j}))}{g(x_{i,j})} \\
&= \sum_{j=1}^{n} \frac{(\varepsilon N)^2 / n \cdot (1 - g(x_{i,j}))}{g(x_{i,j})} \\
&= \frac{(\varepsilon N)^2 \cdot (1 - g(x_{i,j}))}{g(x_{i,j})}
\end{aligned}
$$

A worst case optimal Sampling Function

$$
\begin{aligned}
\mathrm{Var}[Y_i] &= \sum_{j=1}^{n} \frac{x_{i,j}^2 (1 - g(x_{i,j}))}{g(x_{i,j})} \\
&= \sum_{j=1}^{n} \frac{(\varepsilon N)^2 / n \cdot (1 - g(x_{i,j}))}{g(x_{i,j})} \\
&= \frac{(\varepsilon N)^2 \cdot (1 - g(x_{i,j}))}{g(x_{i,j})}
\end{aligned}
$$

- $\mathrm{Var}[Y_i] \leq (\varepsilon N)^2 \rightarrow g(x_{i,j}) \geq \frac{1}{2}$
- Cost: $\sum_{i,j} g(x_{i,j}) = \sqrt{n}/\varepsilon \cdot \frac{1}{2} = \Omega(\sqrt{n}/\varepsilon)$

Instance Optimal

- $g_2(x) = (g_1(x))^2 = \frac{n}{(\varepsilon N)^2} x^2$

Instance Optimal

- $g_2(x) = (g_1(x))^2 = \frac{n}{(\varepsilon N)^2} x^2$

  $\mathrm{Var}[Y_i] \leq (\varepsilon N)^2$ for g_2

Instance Optimal

- $g_2(x) = (g_1(x))^2 = \frac{n}{(\varepsilon N)^2} x^2$

  $\mathrm{Var}[Y_i] \leq (\varepsilon N)^2$ for g_2

- $g_1(x) \geq g_2(x)$

  g_2 is better than g_1 in terms of communication cost
  g_1 is too accurate for some input

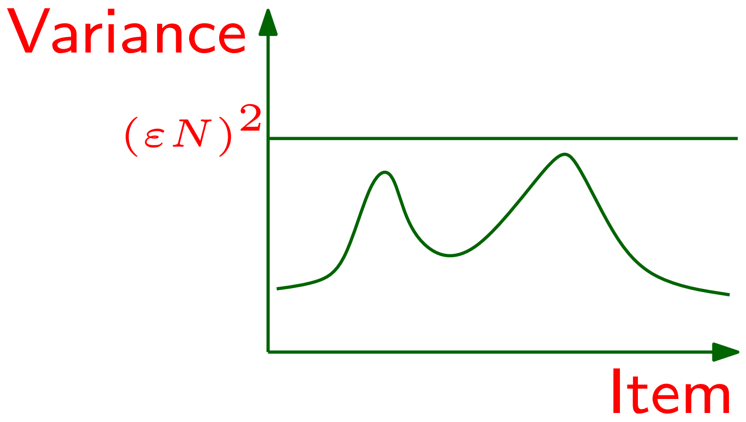
Instance Optimal

- $g_2(x) = (g_1(x))^2 = \frac{n}{(\varepsilon N)^2} x^2$

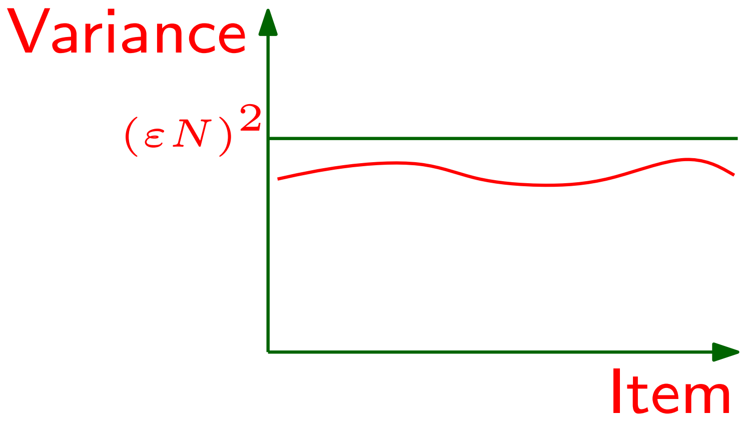
  $\mathrm{Var}[Y_i] \le (\varepsilon N)^2$ for g_2

- $g_1(x) \ge g_2(x)$

  g_2 is better than g_1 in terms of communication cost
  g_1 is too accurate for some input



g_1



g_2

Instance Optimal

- On input $I : \{x_{i,j}\}$, $opt(I) = \sum_{i,j} g_2(x_{i,j})$

Instance Optimal

- On input $I : \{x_{i,j}\}$, $opt(I) = \sum_{i,j} g_2(x_{i,j})$

- $g_2(x)$ is Instance Optimal:

  On input $I : \{x_{i,j}\}$, any valid sampling function $g(x)$ must have cost $\Omega(opt(I))$.

Instance Optimal

- Claim: for any valid function g and any input I, $g(x_{i,j}) \geq \frac{1}{2} g_2(x_{i,j})$ for all $x_{i,j}$ in I.

Instance Optimal

- Claim: for any valid function g and any input I, $g(x_{i,j}) \geq \frac{1}{2} g_2(x_{i,j})$ for all $x_{i,j}$ in I.

- Prove by contradiction

  If $g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j})$ for some $x_{i,j}$

  Exist I', s.t. the variance of g on I' is greater than $(\varepsilon N)^2$

Instance Optimal

- Claim: for any valid function g and any input I, $g(x_{i,j}) \geq \frac{1}{2} g_2(x_{i,j})$ for all $x_{i,j}$ in I.

- Prove by contradiction

  If $g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j})$ for some $x_{i,j}$

  Exist I', s.t. the variance of g on I' is greater than $(\varepsilon N)^2$

  Contradiction!

Instance Optimal

- High level idea

Instance Optimal

- High level idea

$$Y_i = Y_{i,1} + \cdots + Y_{i,n}$$

- $\mathrm{Var}[Y_i] \leq (\varepsilon N)^2$

Instance Optimal

- High level idea

$$Y_i = Y_{i,1} + \cdots + Y_{i,n}$$

- $\mathrm{Var}[Y_i] \leq (\varepsilon N)^2$

The best we can do: $\mathrm{Var}[Y_{i,j}] \leq \frac{(\varepsilon N)^2}{n}$

Otherwise, I': $x'_{i,j} = x_{i,j}$ for all $1 \leq j \leq n$

$$\mathrm{Var}[Y_i] > (\varepsilon N)^2$$

Instance Optimal

- Assumption: $g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j})$

Instance Optimal

- Assumption: $g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j})$

If $x_{i,j} \leq \frac{\varepsilon N}{\sqrt{n}}$

I': $x'_{i,j} = x_{i,j}, 1 \leq j \leq n$ and $y_i = \varepsilon N \sqrt{n} \leq N$

Set other local count, s.t. $\sum_{i,j} x'_{i,j} = N$

Instance Optimal

- Assumption: $g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j})$

If $x_{i,j} \leq \frac{\varepsilon N}{\sqrt{n}}$

I': $x'_{i,j} = x_{i,j}, 1 \leq j \leq n$ and $y_i = \varepsilon N \sqrt{n} \leq N$

Set other local count, s.t. $\sum_{i,j} x'_{i,j} = N$

$$g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j}) \leq \frac{x_{i,j}^2 n}{2(\varepsilon N)^2}$$

$$\mathrm{Var}[Y_i] > n \left(\frac{2(\varepsilon N)^2}{n} - \left(\frac{\varepsilon N}{\sqrt{n}} \right)^2 \right) = (\varepsilon N)^2$$

Instance Optimal

- Assumption: $g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j})$

If $x_{i,j} > \frac{\varepsilon N}{\sqrt{n}} \geq \varepsilon^2 N, g_2(x_{i,j}) = 1$

Instance Optimal

- Assumption: $g(x_{i,j}) < \frac{1}{2}g_2(x_{i,j})$

If $x_{i,j} > \frac{\varepsilon N}{\sqrt{n}} \geq \varepsilon^2 N, g_2(x_{i,j}) = 1$

I': $x'_{i,j} = x_{i,j}, 1 \leq j \leq m, m = \min\{N/x_{i,j}, n\}$

Set other local count, s.t. $\sum_{i,j} x'_{i,j} = N$

Instance Optimal

- Assumption: $g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j})$

If $x_{i,j} > \frac{\varepsilon N}{\sqrt{n}} \geq \varepsilon^2 N, g_2(x_{i,j}) = 1$

I': $x'_{i,j} = x_{i,j}, 1 \leq j \leq m, m = \min\{N/x_{i,j}, n\}$

Set other local count, s.t. $\sum_{i,j} x'_{i,j} = N$

$m x_{i,j}^2 > (\varepsilon N)^2$

$g(x_{i,j}) < \frac{1}{2} g_2(x_{i,j}) = \frac{1}{2}$

$$\mathrm{Var}[Y_i] = m x_{i,j}^2 \left(\frac{1}{g(x_{i,j})} - 1 \right) > (\varepsilon N)^2 \left(\frac{1}{g(x_{i,j})} - 1 \right) = (\varepsilon N)^2$$

Bit Level

- Send an (item, count) pair if sampled

  Cost: $O(\frac{\sqrt{n}}{\varepsilon})(\log u + \log N)$ bits

Bit Level

- Send an (item, count) pair if sampled

  Cost: $O(\frac{\sqrt{n}}{\varepsilon})(\log u + \log N)$ bits

- Reduce to $O(\frac{\sqrt{n}}{\varepsilon})$ bits

Bit Level

- Send an (item, count) pair if sampled

  Cost: $O(\frac{\sqrt{n}}{\varepsilon})(\log u + \log N)$ bits

- Reduce to $O(\frac{\sqrt{n}}{\varepsilon})$ bits

- Bloom Filter

Bit Level

- Bloom Filter

Bit Level

- Bloom Filter
  - Data structure for membership queries with false positive error.

Bit Level

- Bloom Filter
  - Data structure for membership queries with false positive error.
  - $O(\log 1/q)$ bits per item, with false positive probability q.

Bit Level

- $g_1(x) = \frac{\sqrt{n}}{\epsilon N} x$

Bit Level

- $g_1(x) = \frac{\sqrt{n}}{\epsilon N} x$

  $Y_{i,j} = \frac{x}{g_1(x)}$ if $x_{i,j}$ is sampled, otherwise 0

Bit Level

- $g_1(x) = \frac{\sqrt{n}}{\epsilon N} x$

  $Y_{i,j} = \frac{x}{g_1(x)}$ if $x_{i,j}$ is sampled, otherwise 0

- Easy case: $x_{i,j} \leq \frac{\varepsilon N}{\sqrt{n}}$ for all i, j

Bit Level

- $g_1(x) = \frac{\sqrt{n}}{\epsilon N} x$

  $Y_{i,j} = \frac{x}{g_1(x)}$ if $x_{i,j}$ is sampled, otherwise 0

- Easy case: $x_{i,j} \le \frac{\varepsilon N}{\sqrt{n}}$ for all i, j

  $Y_{i,j}$ is either 0 or $\frac{\varepsilon N}{\sqrt{n}}$

  Encode the sampled items in Bloom Filters.

Bit Level

- Query the n Bloom filters.

  Let Z_i be the number of Bloom filter asserts the existence of i

Bit Level

- Query the n Bloom filters.

  Let Z_i be the number of Bloom filter asserts the existence of i

  $\mathbf{E}[Z_i] = x + (n - x)q$, x is the exact number

  $Y_i = \frac{\varepsilon N}{\sqrt{n}} \cdot \frac{Z_i - nq}{1-q}$

Bit Level

- Query the n Bloom filters.

  Let Z_i be the number of Bloom filter asserts the existence of i

  $\mathbf{E}[Z_i] = x + (n-x)q$, x is the exact number

  $Y_i = \frac{\varepsilon N}{\sqrt{n}} \cdot \frac{Z_i - nq}{1-q}$

- $\mathbf{E}[Y_i] = y_i$; $\mathrm{Var}[Y_i] \le \frac{(\varepsilon N)^2}{4(1-q)^2}$

Bit Level

- Query the n Bloom filters.

  Let Z_i be the number of Bloom filter asserts the existence of i

  $\mathbf{E}[Z_i] = x + (n-x)q$, x is the exact number

  $Y_i = \frac{\varepsilon N}{\sqrt{n}} \cdot \frac{Z_i - nq}{1-q}$

- $\mathbf{E}[Y_i] = y_i$; $\mathrm{Var}[Y_i] \leq \frac{(\varepsilon N)^2}{4(1-q)^2}$

- Set q to be a constant $\rightarrow O(1)$ bits per sampled item

  $O(\frac{\sqrt{n}}{\varepsilon})$ bits of communication

Bit Level

- When $x_{i,j} > \frac{\varepsilon N}{\sqrt{n}}$, $g_1(x_{i,j}) = 1$

Bit Level

- When $x_{i,j} > \frac{\varepsilon N}{\sqrt{n}}$, $g_1(x_{i,j}) = 1$

  $Y_{i,j}$ is either 0 or $x_{i,j}$

Bit Level

- When $x_{i,j} > \frac{\varepsilon N}{\sqrt{n}}$, $g_1(x_{i,j}) = 1$

  $Y_{i,j}$ is either 0 or $x_{i,j}$

- $x_{i,j} = a_{i,j}\frac{\varepsilon N}{\sqrt{n}} + b_{i,j}$, $a_{i,j} \leq \frac{\sqrt{n}}{\varepsilon}, b_{i,j} < \frac{\varepsilon N}{\sqrt{n}}$

  $y_i = \frac{\varepsilon N}{\sqrt{n}} \sum_{j=1}^{n} a_{i,j} + \sum_{j=1}^{n} b_{i,j}$

Bit Level

- When $x_{i,j} > \frac{\varepsilon N}{\sqrt{n}}$, $g_1(x_{i,j}) = 1$

  $Y_{i,j}$ is either 0 or $x_{i,j}$

- $x_{i,j} = a_{i,j}\frac{\varepsilon N}{\sqrt{n}} + b_{i,j}$, $a_{i,j} \leq \frac{\sqrt{n}}{\varepsilon}, b_{i,j} < \frac{\varepsilon N}{\sqrt{n}}$

  $y_i = \frac{\varepsilon N}{\sqrt{n}} \sum_{j=1}^{n} a_{i,j} + \sum_{j=1}^{n} b_{i,j}$

- Estimate $\sum_{j=1}^{k} b_{i,j}$ as before

  Encode each bit of the binary form of $a_{i,j}$ separately

Bit Level

- Each node j uses multiple bloom filters

Bit Level

- Each node j uses multiple bloom filters
- Let $a[r]$ be the r-th rightmost bit of a. The rth bloom filter B_r encodes the items

$$\{i|a_{i,j}[r] = 1\}$$

Bit Level

- Each node j uses multiple bloom filters
- Let $a[r]$ be the r-th rightmost bit of a. The rth bloom filter B_r encodes the items

$$\{i | a_{i,j}[r] = 1\}$$

$$a_{1,j} = 101,\ a_{2,j} = 011,\ a_{3,j} = 111$$

$$B_0 = \{1, 2, 3\},\ B_1 = \{2, 3\},\ B_2 = \{1, 3\}$$

Bit Level

- Each node j uses multiple bloom filters
- Let $a[r]$ be the r-th rightmost bit of a. The rth bloom filter B_r encodes the items

$$\{i|a_{i,j}[r] = 1\}$$

$$a_{1,j} = 101,\ a_{2,j} = 011,\ a_{3,j} = 111$$

$$B_0 = \{1, 2, 3\},\ B_1 = \{2, 3\},\ B_2 = \{1, 3\}$$

- Enough to set the false positive rate for B_r to be $1/2^r$

Bit Level

- Each node j uses multiple bloom filters
- Let $a[r]$ be the r-th rightmost bit of a. The rth bloom filter B_r encodes the items

$$\{i | a_{i,j}[r] = 1\}$$

$$a_{1,j} = 101,\ a_{2,j} = 011,\ a_{3,j} = 111$$

$$B_0 = \{1, 2, 3\},\ B_1 = \{2, 3\},\ B_2 = \{1, 3\}$$

- Enough to set the false positive rate for B_r to be $1/2^r$

  Each 1-bit at position r costs $\log 2^r = r$ bits

Bit Level

- Each node j uses multiple bloom filters
- Let $a[r]$ be the r-th rightmost bit of a. The rth bloom filter B_r encodes the items

$$\{i | a_{i,j}[r] = 1\}$$

$$a_{1,j} = 101,\ a_{2,j} = 011,\ a_{3,j} = 111$$

$$B_0 = \{1,2,3\},\ B_1 = \{2,3\},\ B_2 = \{1,3\}$$

- Enough to set the false positive rate for B_r to be $1/2^r$

  Each 1-bit at position r costs $\log 2^r = r$ bits

  Each 1-bit at position r represents $2^r \cdot \frac{\varepsilon N}{\sqrt{n}}$ items

Bit Level

- Each node j uses multiple bloom filters
- Let $a[r]$ be the r-th rightmost bit of a. The rth bloom filter B_r encodes the items

$$\{i | a_{i,j}[r] = 1\}$$

$$a_{1,j} = 101,\ a_{2,j} = 011,\ a_{3,j} = 111$$

$$B_0 = \{1,2,3\},\ B_1 = \{2,3\},\ B_2 = \{1,3\}$$

- Enough to set the false positive rate for B_r to be $1/2^r$

  Each 1-bit at position r costs $\log 2^r = r$ bits

  Each 1-bit at position r represents $2^r \cdot \frac{\varepsilon N}{\sqrt{n}}$ items

  Each copy of $\frac{\varepsilon N}{\sqrt{n}}$ items costs $\frac{r}{2^r} \le O(1)$ bits

Bit Level

- Each node j uses multiple bloom filters
- Let $a[r]$ be the r-th rightmost bit of a. The rth bloom filter B_r encodes the items

$$\{i | a_{i,j}[r] = 1\}$$

$$a_{1,j} = 101,\ a_{2,j} = 011,\ a_{3,j} = 111$$

$$B_0 = \{1,2,3\},\ B_1 = \{2,3\},\ B_2 = \{1,3\}$$

- Enough to set the false positive rate for B_r to be $1/2^r$

  Each 1-bit at position r costs $\log 2^r = r$ bits

  Each 1-bit at position r represents $2^r \cdot \frac{\varepsilon N}{\sqrt{n}}$ items

  Each copy of $\frac{\varepsilon N}{\sqrt{n}}$ items costs $\frac{r}{2^r} \leq O(1)$ bits

  Total cost is at most $O(\frac{\sqrt{n}}{\varepsilon})$ bits

Final Remarks

- More general sampling models
  different $g_{i,j}$ for each $x_{i,j}$

Final Remarks

- More general sampling models
  different $g_{i,j}$ for each $x_{i,j}$

- General communication model

The End

THANK YOU

Q and A