Data Streams: Random Order & Multiple Passes

2009 Barbados Workshop on Computational Complexity

Andrew McGregor

Introduction

Random Order Streams:

- Average case analysis: data is worst-case but order is random.
- Lower bounds are more useful than in the adversarial case.
- Streams ordered randomly: e.g., space-efficient sampling

Introduction

Random Order Streams:

- Average case analysis: data is worst-case but order is random.
- Lower bounds are more useful than in the adversarial case.
- Streams ordered randomly: e.g., space-efficient sampling

Multiple Pass Streams:

- How much extra power do you get with a few extra passes?
- With external data, it's easier to access data sequentially.

Pass-Space Trade-Offs

Problem

Given a stream of n values from $[n]$, what's smallest value that doesn't appear in stream? You have p passes over the data.

Pass-Space Trade-Offs

Problem

Given a stream of n values from $[n]$, what's smallest value that doesn't appear in stream? You have p passes over the data.

- Version 1: All values appear exactly once except for the missing value.

$$\tilde{\Theta}(1)$$

Pass-Space Trade-Offs

Problem

Given a stream of n values from $[n]$, what's smallest value that doesn't appear in stream? You have p passes over the data.

- Version 1: All values appear exactly once except for the missing value.

$$\tilde{\Theta}(1)$$

- Version 2: All values less than smallest missing value appear exactly once

$$\tilde{\Theta}(n^{1/p})$$

Pass-Space Trade-Offs

Problem

Given a stream of n values from $[n]$, what's smallest value that doesn't appear in stream? You have p passes over the data.

- Version 1: All values appear exactly once except for the missing value.

$$\tilde{\Theta}(1)$$

- Version 2: All values less than smallest missing value appear exactly once

$$\tilde{\Theta}(n^{1/p})$$

- Version 3: General problem,

$$\tilde{\Theta}(n/p)$$

Pass-Space Trade-Offs

Problem

Given a stream of n values from $[n]$, what's smallest value that doesn't appear in stream? You have p passes over the data.

- Version 1: All values appear exactly once except for the missing value.

$$\tilde{\Theta}(1)$$

- Version 2: All values less than smallest missing value appear exactly once

$$\tilde{\Theta}(n^{1/p})$$

- Version 3: General problem,

$$\tilde{\Theta}(n/p)$$

Other trade-offs: Find length k increasing sequence given it exists: $\tilde{\Theta}(k^{1+1/(2^p-1)})$ [Liben-Nowell et al. '06, Guha, McGregor '08]

Random Order Streams

Problem

Given m *values from* $[n]$*, find median in* $\text{polylog}(m, n)$ *space.*

Random Order Streams

Problem

Given m values from $[n]$, find median in $\text{polylog}(m, n)$ *space.*

Approximate Median (i.e., one with rank $m/2 \pm t$) in One Pass:

- Adversarial: $\tilde{\Theta}(m)$-approx [Greenwald, Khanna '01]
- Random: $\tilde{O}(m^{1/2})$-approx [Guha, McGregor '06]

Random Order Streams

Problem

Given m values from $[n]$*, find median in* $\text{polylog}(m, n)$ *space.*

Approximate Median (i.e., one with rank $m/2 \pm t$) in One Pass:

- Adversarial: $\tilde{\Theta}(m)$-approx [Greenwald, Khanna '01]
- Random: $\tilde{O}(m^{1/2})$-approx [Guha, McGregor '06]

Exact Median in Multiple Passes

- Adversarial: $\Theta(\log m/\log\log m)$ pass [Munro, Paterson '78, Guha, McGregor '07]
- Random: $\Theta(\log\log m)$ pass [Guha, McGregor '06, Chakrabarti, Jayram, Patrascu '08, Chakrabarti, Cormode, McGregor '08]

Selection

- Adversarial Order
- Random Order

Frequency Moments

Hamming Distance

Outline

Selection
- Adversarial Order
- Random Order

Frequency Moments

Hamming Distance

Outline

- Selection
  - Adversarial Order
  - Random Order
- Frequency Moments
- Hamming Distance

Algorithms for Median in Adversarial-Order Stream

Theorem (Adversarial Order)

Can find element of rank $m/2 \pm \epsilon m$ in one pass and $\tilde{O}(\epsilon^{-1})$ space.
Can find median in $O(\log m / \log \log m)$ passes and $\tilde{O}(1)$ space.

Algorithms for Median in Adversarial-Order Stream

Theorem (Adversarial Order)

Can find element of rank $m/2 \pm \epsilon m$ in one pass and $\tilde{O}(\epsilon^{-1})$ space.
Can find median in $O(\log m / \log\log m)$ passes and $\tilde{O}(1)$ space.

- Already seen one pass result:
  - Can find elements with rank $i\epsilon m \pm \epsilon m$ for $i \in [\epsilon^{-1}]$

Algorithms for Median in Adversarial-Order Stream

Theorem (Adversarial Order)

Can find element of rank $m/2 \pm \epsilon m$ in one pass and $\tilde{O}(\epsilon^{-1})$ space.
Can find median in $O(\log m / \log \log m)$ passes and $\tilde{O}(1)$ space.

- Already seen one pass result:
  - Can find elements with rank $i\epsilon m \pm \epsilon m$ for $i \in [\epsilon^{-1}]$
- For multiple-pass result:

Algorithms for Median in Adversarial-Order Stream

Theorem (Adversarial Order)

Can find element of rank $m/2 \pm \epsilon m$ in one pass and $\tilde{O}(\epsilon^{-1})$ space. Can find median in $O(\log m / \log\log m)$ passes and $\tilde{O}(1)$ space.

- Already seen one pass result:
  - Can find elements with rank $i\epsilon m \pm \epsilon m$ for $i \in [\epsilon^{-1}]$
- For multiple-pass result:
  - In pass 1, use one pass alg. with $\epsilon = \frac{1}{\log m}$ to find a and b s.t.

$$\text{rank}(a) = \frac{m}{2} - \frac{2m}{\log m} \pm \frac{m}{\log m} \text{ and } \text{rank}(b) = \frac{m}{2} + \frac{2m}{\log m} \pm \frac{m}{\log m}$$

Algorithms for Median in Adversarial-Order Stream

Theorem (Adversarial Order)

Can find element of rank $m/2 \pm \epsilon m$ in one pass and $\tilde{O}(\epsilon^{-1})$ space.
Can find median in $O(\log m / \log \log m)$ passes and $\tilde{O}(1)$ space.

- Already seen one pass result:
  - Can find elements with rank $i\epsilon m \pm \epsilon m$ for $i \in [\epsilon^{-1}]$
- For multiple-pass result:
  - In pass 1, use one pass alg. with $\epsilon = \frac{1}{\log m}$ to find a and b s.t.

$$\text{rank}(a) = \frac{m}{2} - \frac{2m}{\log m} \pm \frac{m}{\log m} \text{ and } \text{rank}(b) = \frac{m}{2} + \frac{2m}{\log m} \pm \frac{m}{\log m}$$

  - In pass 2, compute $\text{rank}(a)$ and $\text{rank}(b)$

Algorithms for Median in Adversarial-Order Stream

Theorem (Adversarial Order)

Can find element of rank $m/2 \pm \epsilon m$ in one pass and $\tilde{O}(\epsilon^{-1})$ space.
Can find median in $O(\log m / \log\log m)$ passes and $\tilde{O}(1)$ space.

- Already seen one pass result:
  - Can find elements with rank $i\epsilon m \pm \epsilon m$ for $i \in [\epsilon^{-1}]$
- For multiple-pass result:
  - In pass 1, use one pass alg. with $\epsilon = \frac{1}{\log m}$ to find a and b s.t.

    $$\text{rank}(a) = \frac{m}{2} - \frac{2m}{\log m} \pm \frac{m}{\log m} \text{ and } \text{rank}(b) = \frac{m}{2} + \frac{2m}{\log m} \pm \frac{m}{\log m}$$

  - In pass 2, compute $\text{rank}(a)$ and $\text{rank}(b)$
  - Recurse on elements in the range (a, b).

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^{\delta}$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^{\delta}$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^{\delta}$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$
- Alice constructs $A = \{2i + x_i : i \in [t]\}$

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^\delta$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$
- Alice constructs $A = \{2i + x_i : i \in [t]\}$
- Bob constructs $B = \{t - j$ copies of $0, j - 1$ copies of $2t + 2\}$

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^{\delta}$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$
- Alice constructs $A = \{2i + x_i : i \in [t]\}$
- Bob constructs $B = \{t - j$ copies of $0, j - 1$ copies of $2t + 2\}$
- Median of the $2t - 1$ values is $2j + x_j$

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^{\delta}$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$
- Alice constructs $A = \{2i + x_i : i \in [t]\}$
- Bob constructs $B = \{t - j \text{ copies of } 0, j - 1 \text{ copies of } 2t + 2\}$
- Median of the $2t - 1$ values is $2j + x_j$
- $\therefore$ Exact median requires $\Omega(t) = \Omega(m)$ space.

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^{\delta}$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$
- Alice constructs $A = \{2i + x_i : i \in [t]\}$
- Bob constructs $B = \{t - j \text{ copies of } 0, j - 1 \text{ copies of } 2t + 2\}$
- Median of the $2t - 1$ values is $2j + x_j$
- $\therefore$ Exact median requires $\Omega(t) = \Omega(m)$ space.
- For approximate result, duplicate each element $2m^{\delta} + 1$ times.

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^{\delta}$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$
- Alice constructs $A = \{2i + x_i : i \in [t]\}$
- Bob constructs $B = \{t - j \text{ copies of } 0, j - 1 \text{ copies of } 2t + 2\}$
- Median of the $2t - 1$ values is $2j + x_j$
- $\therefore$ Exact median requires $\Omega(t) = \Omega(m)$ space.
- For approximate result, duplicate each element $2m^{\delta} + 1$ times.
- $\therefore$ Approx median requires $\Omega(t) = \Omega(m/m^{\delta})$ space.

One Pass Lower Bound

Theorem

Finding $m/2 \pm m^\delta$ rank element in 1 pass requires $\Omega(m^{1-\delta})$ space.

- $\textsc{Index}$ Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$
- Alice constructs $A = \{2i + x_i : i \in [t]\}$
- Bob constructs $B = \{t - j \text{ copies of } 0, j - 1 \text{ copies of } 2t + 2\}$
- Median of the $2t - 1$ values is $2j + x_j$
- $\therefore$ Exact median requires $\Omega(t) = \Omega(m)$ space.
- For approximate result, duplicate each element $2m^\delta + 1$ times.
- $\therefore$ Approx median requires $\Omega(t) = \Omega(m/m^\delta)$ space.

Exercise

Prove an algorithm that doesn't know m in advance requires $\Omega(m)$ space to find median even when the data comes in sorted order.

Two Pass Lower Bound

Theorem

Finding median in 2 passes requires $\Omega(m^{1/2})$ *space.*

Two Pass Lower Bound

Theorem

Finding median in 2 passes requires $\Omega(m^{1/2})$ *space.*

- "2-level INDEX" Reduction: Alice has $x^1, \ldots, x^t \in \{0,1\}^t$, Bob has $y \in [t]^t$, Charlie has $i \in [t]$. To determine x^i_j where $j = y_i$ after two rounds, requires $\Omega(t)$ bits of communication.
  [Nisan, Widgerson '91]

Two Pass Lower Bound

Theorem

Finding median in 2 passes requires $\Omega(m^{1/2})$ space.

- "2-level INDEX" Reduction: Alice has $x^1, \ldots, x^t \in \{0,1\}^t$, Bob has $y \in [t]^t$, Charlie has $i \in [t]$. To determine x^i_j where $j = y_i$ after two rounds, requires $\Omega(t)$ bits of communication. [Nisan, Widgerson '91]
- For $j \in [t]$, appropriate players construct

$$A_i = \{2j + x^i_j : i \in [t]\} + o_i \text{ where } o_i = B(i-1)$$

$$B_i = \{t - y_i \text{ copies of } 0 \text{ and } y_i - 1 \text{ copies of } B\} + o_i$$

$$C = \{t - i \text{ copies of } 0 \text{ and } i - 1 \text{ copies of } Bo_t\}$$

Two Pass Lower Bound

Theorem

Finding median in 2 passes requires $\Omega(m^{1/2})$ space.

- "2-level INDEX" Reduction: Alice has $x^1, \ldots, x^t \in \{0,1\}^t$, Bob has $y \in [t]^t$, Charlie has $i \in [t]$. To determine x^i_j where $j = y_i$ after two rounds, requires $\Omega(t)$ bits of communication. [Nisan, Widgerson '91]
- For $j \in [t]$, appropriate players construct

$$A_i = \{2j + x^i_j : i \in [t]\} + o_i \text{ where } o_i = B(i-1)$$

$$B_i = \{t - y_i \text{ copies of } 0 \text{ and } y_i - 1 \text{ copies of } B\} + o_i$$

$$C = \{t - i \text{ copies of } 0 \text{ and } i - 1 \text{ copies of } Bo_t\}$$

- Median of the $O(t^2)$ values is $o_i + 2j + x^i_j$ where $j = y_i$

Two Pass Lower Bound

Theorem

Finding median in 2 passes requires $\Omega(m^{1/2})$ space.

- "2-level INDEX" Reduction: Alice has $x^1, \ldots, x^t \in \{0,1\}^t$, Bob has $y \in [t]^t$, Charlie has $i \in [t]$. To determine x^i_j where $j = y_i$ after two rounds, requires $\Omega(t)$ bits of communication.
  [Nisan, Widgerson '91]
- For $j \in [t]$, appropriate players construct

$$A_i = \{2j + x^i_j : i \in [t]\} + o_i \text{ where } o_i = B(i-1)$$

$$B_i = \{t - y_i \text{ copies of } 0 \text{ and } y_i - 1 \text{ copies of } B\} + o_i$$

$$C = \{t - i \text{ copies of } 0 \text{ and } i - 1 \text{ copies of } Bo_t\}$$

- Median of the $O(t^2)$ values is $o_i + 2j + x^i_j$ where $j = y_i$
- $\therefore$ Exact median requires $\Omega(t) = \Omega(m^{1/2})$ space.

Outline

Selection

Adversarial Order

Random Order

Frequency Moments

Hamming Distance

Random Order Algorithms

Theorem

Can find element of rank $m/2 \pm \tilde{O}(\sqrt{m})$ *in one pass and* $\tilde{O}(1)$ *space. Can find median in* $O(\log \log m)$ *passes and* $\tilde{O}(1)$ *space.*

Random Order Algorithms

Theorem

Can find element of rank $m/2 \pm \tilde{O}(\sqrt{m})$ *in one pass and* $\tilde{O}(1)$ *space. Can find median in* $O(\log \log m)$ *passes and* $\tilde{O}(1)$ *space.*

- One pass result:

Random Order Algorithms

Theorem

Can find element of rank $m/2 \pm \tilde{O}(\sqrt{m})$ in one pass and $\tilde{O}(1)$ space. Can find median in $O(\log \log m)$ passes and $\tilde{O}(1)$ space.

- One pass result:
  - Split stream into $O(\log m)$ segments of length $O(m/\log m)$

Random Order Algorithms

Theorem

Can find element of rank $m/2 \pm \tilde{O}(\sqrt{m})$ in one pass and $\tilde{O}(1)$ space. Can find median in $O(\log \log m)$ passes and $\tilde{O}(1)$ space.

- One pass result:
  - Split stream into $O(\log m)$ segments of length $O(m/\log m)$
  - At start of i-th segment: we think $\text{rank}(a_i) < m/2 < \text{rank}(b_i)$.

Random Order Algorithms

Theorem

Can find element of rank $m/2 \pm \tilde{O}(\sqrt{m})$ in one pass and $\tilde{O}(1)$ space. Can find median in $O(\log \log m)$ passes and $\tilde{O}(1)$ space.

- One pass result:
  - Split stream into $O(\log m)$ segments of length $O(m/\log m)$
  - At start of i-th segment: we think $\text{rank}(a_i) < m/2 < \text{rank}(b_i)$.
  - Let c be first element in segment with $a_i < c < b_i$

Random Order Algorithms

Theorem

Can find element of rank $m/2 \pm \tilde{O}(\sqrt{m})$ in one pass and $\tilde{O}(1)$ space. Can find median in $O(\log\log m)$ passes and $\tilde{O}(1)$ space.

- One pass result:
  - Split stream into $O(\log m)$ segments of length $O(m/\log m)$
  - At start of i-th segment: we think $\text{rank}(a_i) < m/2 < \text{rank}(b_i)$.
  - Let c be first element in segment with $a_i < c < b_i$
  - In rest of segment, estimate $\text{rank}(c)$ by $\tilde{r}$

Random Order Algorithms

Theorem

Can find element of rank $m/2 \pm \tilde{O}(\sqrt{m})$ in one pass and $\tilde{O}(1)$ space. Can find median in $O(\log \log m)$ passes and $\tilde{O}(1)$ space.

- One pass result:
  - Split stream into $O(\log m)$ segments of length $O(m/\log m)$
  - At start of i-th segment: we think $\text{rank}(a_i) < m/2 < \text{rank}(b_i)$.
  - Let c be first element in segment with $a_i < c < b_i$
  - In rest of segment, estimate $\text{rank}(c)$ by $\tilde{r}$
  - If $\tilde{r} = m/2 \pm \tilde{O}(\sqrt{m})$ return $\tilde{r}$, otherwise:

$$(a_{i+1}, b_{i+1}) = \begin{cases} (a_i, c) & \text{if } \tilde{r} > m/2 \\ (c, b_i) & \text{if } \tilde{r} < m/2 \end{cases}$$

Random Order Algorithms

Theorem

Can find element of rank $m/2 \pm \tilde{O}(\sqrt{m})$ in one pass and $\tilde{O}(1)$ space. Can find median in $O(\log \log m)$ passes and $\tilde{O}(1)$ space.

- One pass result:
  - Split stream into $O(\log m)$ segments of length $O(m/\log m)$
  - At start of i-th segment: we think $\text{rank}(a_i) < m/2 < \text{rank}(b_i)$.
  - Let c be first element in segment with $a_i < c < b_i$
  - In rest of segment, estimate $\text{rank}(c)$ by $\tilde{r}$
  - If $\tilde{r} = m/2 \pm \tilde{O}(\sqrt{m})$ return $\tilde{r}$, otherwise:

$$(a_{i+1}, b_{i+1}) = \begin{cases} (a_i, c) & \text{if } \tilde{r} > m/2 \\ (c, b_i) & \text{if } \tilde{r} < m/2 \end{cases}$$

- For multiple-pass result: Recurse with care!

Random Order One Pass Lower Bound

Theorem

Finding median in 1 pass requires $\Omega(m^{1/2})$ *space.*

Random Order One Pass Lower Bound

Theorem

Finding median in 1 pass requires $\Omega(m^{1/2})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$. Solving problem requires $\Omega(t)$ even when $x \in_R \{0,1\}^t$.

Random Order One Pass Lower Bound

Theorem

Finding median in 1 pass requires $\Omega(m^{1/2})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$. Solving problem requires $\Omega(t)$ even when $x \in_R \{0,1\}^t$.
- For some constant $c > 0$, define:

$$A = \{2i + x_i : i \in [t]\}$$

$$B = \{ct^2 + t - j \text{ copies of } 0 \text{ and } ct^2 + j - 1 \text{ copies of } 2t + 2\}$$

Random Order One Pass Lower Bound

Theorem

Finding median in 1 pass requires $\Omega(m^{1/2})$ *space.*

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$. Solving problem requires $\Omega(t)$ even when $x \in_R \{0,1\}^t$.
- For some constant $c > 0$, define:

$$A = \{2i + x_i : i \in [t]\}$$

$$B = \{ct^2 + t - j \text{ copies of } 0 \text{ and } ct^2 + j - 1 \text{ copies of } 2t + 2\}$$

- Alice and Bob simulate algorithm on random permutation of $A \cup B$. Alice determines 1st half and Bob determines 2nd half:

Random Order One Pass Lower Bound

Theorem

Finding median in 1 pass requires $\Omega(m^{1/2})$ *space.*

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$. Solving problem requires $\Omega(t)$ even when $x \in_R \{0,1\}^t$.
- For some constant $c > 0$, define:

$$A = \{2i + x_i : i \in [t]\}$$

$$B = \{ct^2 + t - j \text{ copies of } 0 \text{ and } ct^2 + j - 1 \text{ copies of } 2t + 2\}$$

- Alice and Bob simulate algorithm on random permutation of $A \cup B$. Alice determines 1st half and Bob determines 2nd half:
  - Alice assumes $j = t/2$: Bob "fixes" the balance.

Random Order One Pass Lower Bound

Theorem

Finding median in 1 pass requires $\Omega(m^{1/2})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$. Solving problem requires $\Omega(t)$ even when $x \in_R \{0,1\}^t$.
- For some constant $c > 0$, define:

$$A = \{2i + x_i : i \in [t]\}$$

$$B = \{ct^2 + t - j \text{ copies of } 0 \text{ and } ct^2 + j - 1 \text{ copies of } 2t + 2\}$$

- Alice and Bob simulate algorithm on random permutation of $A \cup B$. Alice determines 1st half and Bob determines 2nd half:
  - Alice assumes $j = t/2$: Bob "fixes" the balance.
  - Bob guesses values of x_i if $2i + x_i$ appears in his half.

Random Order One Pass Lower Bound

Theorem

Finding median in 1 pass requires $\Omega(m^{1/2})$ space.

- INDEX Reduction: Alice has $x \in \{0,1\}^t$, Bob has $j \in [t]$. Solving problem requires $\Omega(t)$ even when $x \in_R \{0,1\}^t$.
- For some constant $c > 0$, define:

$$A = \{2i + x_i : i \in [t]\}$$

$$B = \{ct^2 + t - j \text{ copies of } 0 \text{ and } ct^2 + j - 1 \text{ copies of } 2t + 2\}$$

- Alice and Bob simulate algorithm on random permutation of $A \cup B$. Alice determines 1st half and Bob determines 2nd half:
  - Alice assumes $j = t/2$: Bob "fixes" the balance.
  - Bob guesses values of x_i if $2i + x_i$ appears in his half.
- Choosing large c ensures ordering is sufficiently random.

Outline

Selection
Adversarial Order
Random Order

Frequency Moments

Hamming Distance

Frequency Moments

Problem

Given m elements from $[n]$, find $(1+\epsilon)$ approx for $F_k = \sum_{i \in [n]} f_i^k$ with probability $1-\delta$ where f_i is the frequency of item i.

Frequency Moments

Problem

Given m elements from $[n]$, find $(1+\epsilon)$ approx for $F_k = \sum_{i \in [n]} f_i^k$ with probability $1-\delta$ where f_i is the frequency of item i.

Theorem (Chakrabarti et al. '03, Indyk, Woodruff '05)

$\tilde{\Theta}_{\epsilon,\delta}(n^{1-2/k})$ *space when stream is in adversarial order.*

Frequency Moments

Problem

Given m elements from $[n]$, find $(1+\epsilon)$ approx for $F_k = \sum_{i \in [n]} f_i^k$ with probability $1-\delta$ where f_i is the frequency of item i.

Theorem (Chakrabarti et al. '03, Indyk, Woodruff '05)

$\tilde{\Theta}_{\epsilon,\delta}(n^{1-2/k})$ *space when stream is in adversarial order.*

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-2.5/k})$ *space necessary when the stream is in random order.*

Frequency Moments

Problem

Given m elements from $[n]$, find $(1+\epsilon)$ approx for $F_k = \sum_{i \in [n]} f_i^k$ with probability $1-\delta$ where f_i is the frequency of item i.

Theorem (Chakrabarti et al. '03, Indyk, Woodruff '05)

$\tilde{\Theta}_{\epsilon,\delta}(n^{1-2/k})$ *space when stream is in adversarial order.*

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-2.5/k})$ *space necessary when the stream is in random order.*

Rumor has it that that this has been tightened to $\Omega(n^{1-2/k})\ldots$

Adversarial Order Lower Bound

- t-DISJ Reduction: t sets $S_1, \ldots, S_t \subset [n]$ of size n/t. Are sets pairwise-disjoint or does there exists common element?

Adversarial Order Lower Bound

- t-DISJ Reduction: t sets $S_1, \ldots, S_t \subset [n]$ of size n/t. Are sets pairwise-disjoint or does there exists common element?
- If i-th player has S_i, t-DISJ requires $\tilde{\Omega}(n/t)$ communication.
  [Bar-Yosseff et al. '02, Chakrabarti et al. '03]

Adversarial Order Lower Bound

- t-DISJ Reduction: t sets $S_1, \ldots, S_t \subset [n]$ of size n/t. Are sets pairwise-disjoint or does there exists common element?
- If i-th player has S_i, t-DISJ requires $\tilde{\Omega}(n/t)$ communication. [Bar-Yosseff et al. '02, Chakrabarti et al. '03]
- Let $S = \cup_{i \in [t]} S_i$. If $t^k > 2n$,

$$(F_k(S) \leq n) \Rightarrow (t\text{-}\mathrm{DISJ}(S) = \text{“disjoint”})$$

$$(F_k(S) \geq 2n) \Rightarrow (t\text{-}\mathrm{DISJ}(S) = \text{“common element”})$$

Adversarial Order Lower Bound

- t-DISJ Reduction: t sets $S_1, \ldots, S_t \subset [n]$ of size n/t. Are sets pairwise-disjoint or does there exists common element?
- If i-th player has S_i, t-DISJ requires $\tilde{\Omega}(n/t)$ communication. [Bar-Yosseff et al. '02, Chakrabarti et al. '03]
- Let $S = \cup_{i \in [t]} S_i$. If $t^k > 2n$,

$$(F_k(S) \leq n) \Rightarrow (t\text{-}\mathrm{DISJ}(S) = \text{“disjoint”})$$

$$(F_k(S) \geq 2n) \Rightarrow (t\text{-}\mathrm{DISJ}(S) = \text{“common element”})$$

- An 1-pass, s-space algorithm that 2-approximates F_k gives a ts-space algorithm that solves $(2n)^{1/k}$-DISJ

Random Order Lower Bound

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-3/k})$ *space necessary for random order stream.*

Random Order Lower Bound

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-3/k})$ *space necessary for random order stream.*

- t-DISJ Reduction: $S_1, \ldots, S_t \subset [n']$ of size n'/t.

Random Order Lower Bound

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-3/k})$ *space necessary for random order stream.*

- t-DISJ Reduction: $S_1, \ldots, S_t \subset [n']$ of size n'/t.
- Using public random bits, players pick random stream S from $[2n]^n$, random map $f : [n] \to [n]$, and random permutations π_i

Random Order Lower Bound

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-3/k})$ *space necessary for random order stream.*

- t-DISJ Reduction: $S_1, \ldots, S_t \subset [n']$ of size n'/t.
- Using public random bits, players pick random stream S from $[2n]^n$, random map $f : [n] \to [n]$, and random permutations π_i

Random Order Lower Bound

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-3/k})$ *space necessary for random order stream.*

- t-DISJ Reduction: $S_1, \ldots, S_t \subset [n']$ of size n'/t.
- Using public random bits, players pick random stream S from $[2n]^n$, random map $f : [n] \to [n]$, and random permutations π_i
- Player i computes string $\sigma(f(S_i))$

Random Order Lower Bound

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-3/k})$ *space necessary for random order stream.*

- t-DISJ Reduction: $S_1, \ldots, S_t \subset [n']$ of size n'/t.
- Using public random bits, players pick random stream S from $[2n]^n$, random map $f : [n] \to [n]$, and random permutations π_i
- Player i computes string $\sigma(f(S_i))$
- Players embed the strings in S at random locations:
  - If embedding of two strings overlap, abort algorithm.
  - Probability of aborting is sufficiently small if $n' = n^{1-1/k}$

Random Order Lower Bound

Theorem (Andoni et al. '08)

$\tilde{\Omega}(n^{1-3/k})$ *space necessary for random order stream.*

- t-DISJ Reduction: $S_1, \ldots, S_t \subset [n']$ of size n'/t.
- Using public random bits, players pick random stream S from $[2n]^n$, random map $f : [n] \to [n]$, and random permutations π_i
- Player i computes string $\sigma(f(S_i))$
- Players embed the strings in S at random locations:
  - If embedding of two strings overlap, abort algorithm.
  - Probability of aborting is sufficiently small if $n' = n^{1-1/k}$

Extending ideas, gives $\tilde{\Omega}(n^{1-2/k})$.

Outline

Selection

Adversarial Order

Random Order

Frequency Moments

Hamming Distance

Hamming Distance Lower Bound

Problem

Alice knows $x \in \{0,1\}^n$ and Bob knows $y \in \{0,1\}^n$. Want to estimate hamming distance up to $\pm o(\sqrt{n})$ with probability $9/10$.

Hamming Distance Lower Bound

Problem

Alice knows $x \in \{0,1\}^n$ *and Bob knows* $y \in \{0,1\}^n$. *Want to estimate hamming distance up to* $\pm o(\sqrt{n})$ *with probability* $9/10$.

Theorem (Woodruff 2004, Jayram et al. 2008)

Any one-way protocol requires $\Omega(n)$ *bits of communication.*

Hamming Distance Lower Bound

Problem

Alice knows $x \in \{0,1\}^n$ and Bob knows $y \in \{0,1\}^n$. Want to estimate hamming distance up to $\pm o(\sqrt{n})$ with probability $9/10$.

Theorem (Woodruff 2004, Jayram et al. 2008)

Any one-way protocol requires $\Omega(n)$ bits of communication.

Theorem (Brody, Chakrabarti last week)

Any $O(1)$-round protocol requires $\Omega(n)$ bits of communication.

Hamming Distance Lower Bound

Problem

Alice knows $x \in \{0,1\}^n$ and Bob knows $y \in \{0,1\}^n$. Want to estimate hamming distance up to $\pm o(\sqrt{n})$ with probability $9/10$.

Theorem (Woodruff 2004, Jayram et al. 2008)

Any one-way protocol requires $\Omega(n)$ bits of communication.

Theorem (Brody, Chakrabarti last week)

Any $O(1)$-round protocol requires $\Omega(n)$ bits of communication.

Corollary

Any $O(1)$-pass algorithm that $(1+\epsilon)$ approximates F_0 or F_2 requires $\Omega(\epsilon^{-2})$ space.

One-Pass Lower Bound (1/2)

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.

One-Pass Lower Bound (1/2)

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^n$ using public random bits.

One-Pass Lower Bound (1/2)

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^n$ using public random bits.
- Alice computes $\mathrm{sn}(r.z)$ and Bob computes $\mathrm{sn}(r_j)$

One-Pass Lower Bound (1/2)

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^n$ using public random bits.
- Alice computes $\mathrm{sn}(r.z)$ and Bob computes $\mathrm{sn}(r_j)$
- Claim: For some constant $c > 0$,

$$\mathbb{P}\left[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\right] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

One-Pass Lower Bound (1/2)

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^n$ using public random bits.
- Alice computes $\mathrm{sn}(r.z)$ and Bob computes $\mathrm{sn}(r_j)$
- Claim: For some constant $c > 0$,

$$\mathbb{P}\left[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\right] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- Repeat $n = O(t)$ times to construct

$$x_i = I[\mathrm{sn}(r.z) = +] \quad \text{and} \quad y_i = I[\mathrm{sn}(r_j) = +]$$

One-Pass Lower Bound (1/2)

- Reduction from INDEX problem: Alice knows $z \in \{0,1\}^t$ and Bob knows $j \in [t]$. Let's assume $|z| = t/2$ and this is odd.
- Alice and Bob pick $r \in_R \{-1,1\}^n$ using public random bits.
- Alice computes $\mathrm{sn}(r.z)$ and Bob computes $\mathrm{sn}(r_j)$
- Claim: For some constant $c > 0$,

$$\mathbb{P}\left[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\right] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- Repeat $n = O(t)$ times to construct

$$x_i = I[\mathrm{sn}(r.z) = +] \quad \text{and} \quad y_i = I[\mathrm{sn}(r_j) = +]$$

- With probability $9/10$, for some constants $c_1 < c_2$,

$$z_j = 0 \Rightarrow \Delta(x,y) \geq n/2 - c_1\sqrt{n}$$

$$z_j = 1 \Rightarrow \Delta(x,y) \leq n/2 - c_2\sqrt{n}$$

One-Pass Lower Bound (2/2)

Claim

For some constant $c > 0$,

$$\mathbb{P}\left[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\right] = \begin{cases} 1/2 & \textit{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \textit{if } z_j = 1 \end{cases}$$

One-Pass Lower Bound (2/2)

Claim

For some constant $c > 0$,

$$\mathbb{P}\left[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\right] = \begin{cases} 1/2 & \textit{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \textit{if } z_j = 1 \end{cases}$$

- If $z_j = 0$ then $\mathrm{sn}(r.z)$ and $\mathrm{sn}(r_j)$ are independent.

One-Pass Lower Bound (2/2)

Claim

For some constant $c > 0$,

$$\mathbb{P}[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)] = \begin{cases} 1/2 & \textit{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \textit{if } z_j = 1 \end{cases}$$

- If $z_j = 0$ then $\mathrm{sn}(r.z)$ and $\mathrm{sn}(r_j)$ are independent.
- If $z_j = 1$, let $s = r.z - r_j$, $A = \{\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\}$:

One-Pass Lower Bound (2/2)

Claim

For some constant $c > 0$,

$$\mathbb{P}\left[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\right] = \begin{cases} 1/2 & \text{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \text{if } z_j = 1 \end{cases}$$

- If $z_j = 0$ then $\mathrm{sn}(r.z)$ and $\mathrm{sn}(r_j)$ are independent.
- If $z_j = 1$, let $s = r.z - r_j$, $A = \{\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\}$:

$$\mathbb{P}[A] = \mathbb{P}[A|s = 0]\,\mathbb{P}[s = 0] + \mathbb{P}[A|s \neq 0]\,\mathbb{P}[s \neq 0]$$

One-Pass Lower Bound (2/2)

Claim

For some constant $c > 0$,

$$\mathbb{P}\left[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\right] = \begin{cases} 1/2 & \textit{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \textit{if } z_j = 1 \end{cases}$$

- If $z_j = 0$ then $\mathrm{sn}(r.z)$ and $\mathrm{sn}(r_j)$ are independent.
- If $z_j = 1$, let $s = r.z - r_j$, $A = \{\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\}$:

$$\mathbb{P}[A] = \mathbb{P}[A|s = 0]\,\mathbb{P}[s = 0] + \mathbb{P}[A|s \neq 0]\,\mathbb{P}[s \neq 0]$$

$$\mathbb{P}[A|s = 0] = 1 \text{ and } \mathbb{P}[A|s \neq 0] = 1/2$$

One-Pass Lower Bound (2/2)

Claim

For some constant $c > 0$,

$$\mathbb{P}\left[\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\right] = \begin{cases} 1/2 & \textit{if } z_j = 0 \\ 1/2 + c/\sqrt{t} & \textit{if } z_j = 1 \end{cases}$$

- If $z_j = 0$ then $\mathrm{sn}(r.z)$ and $\mathrm{sn}(r_j)$ are independent.
- If $z_j = 1$, let $s = r.z - r_j$, $A = \{\mathrm{sn}(r.z) = \mathrm{sn}(r_j)\}$:

$$\mathbb{P}[A] = \mathbb{P}[A|s = 0]\,\mathbb{P}[s = 0] + \mathbb{P}[A|s \neq 0]\,\mathbb{P}[s \neq 0]$$

$$\mathbb{P}[A|s = 0] = 1 \text{ and } \mathbb{P}[A|s \neq 0] = 1/2$$

$$\mathbb{P}[s = 0] = 2c/\sqrt{n} \text{ for some constant } c > 0$$

Summary: We looked at some nice problems, our curiousity is piqued, and now we want to start finding more problems to solve.

Thanks!