Tight Lower Bound
for Comparison-Based Quantile Summaries

Based on joint work with Graham Cormode (Warwick)

8 April 2020

University of Warwick

Pavel Veseley
Overview of the talk

Streaming Model

Big Data Algorithms

Quantiles & Distributions

0.5

median

0

Tight Lower Bound for Quantile Summaries

Pavel Veselý

Overview of the talk
Motivation: Monitoring Latencies of Web Requests

- Millions of observations
  - no need to store all observed latencies

- How does the distribution look like?
- What is the median latency?
  - Average latency too high due to ±2% of very high latencies
Motivation: Monitoring Latencies of Web Requests

- No need to store all observed latencies

Millions of observations

Motivation: Monitoring Latencies of Web Requests

What is the median latency?

How does the distribution look like?

- No need to store all observed latencies

Millions of Observations

Motivation: Monitoring Latencies of Web Requests

- Average latency too high due to ~2% of very high latencies

What is the median latency?

How does the distribution look like?

- Millions of observations
  - no need to store all observed latencies

Pavel Veselý

Tight Lower Bound for Quantile Summaries

Streaming Model

Motivation: monitoring latencies of requests

Streaming Model = one pass over data & limited memory

Streaming algorithm
- receives data in a stream
- uses memory sublinear in $N$ = stream length
- at the end, computes the answer

Challenges:
- $N$ very large & not known
- Data independent
- Stream ordered arbitrarily
- No random access to data

Main objective:
- space
- How to summarize the input?
Streaming Model

Motivation: monitoring latencies of requests

Streaming model = one pass over data & limited memory

Challenges:
- $N$ very large & not known
- Data independent
- Stream ordered arbitrarily
- No random access to data

Main objective:
- Space
  - How to summarize the input?
Streaming Model

Motivation: monitoring latencies of requests

Streaming model = one pass over data & limited memory

- Receives data in a stream, item by item
- Uses memory sublinear in $N = \text{stream length}$
- At the end, computes the answer

Challenges:
- $N$ very large & not known
- Data independent
- Stream ordered arbitrarily
- No random access to data

Main objective: space

How to summarize the input?
Streaming Model

Motivation: monitoring latencies of requests

Streaming model = one pass over data \& limited memory

- Streaming algorithm
  - receives data in a stream, item by item
  - uses memory sublinear in \( N \), stream length
  - at the end, computes the answer

Challenges:
- \( N \) very large \& not known
- Data independent
- Stream ordered arbitrarily
- No random access to data

Pavel Veselý
Tight Lower Bound for Quantile Summaries
Streaming Model

Motivation: monitoring latencies of requests

Streaming model = one pass over data & limited memory

Streaming algorithm

- receives data in a stream, item by item
- uses memory sublinear in $N = \text{stream length}$
- at the end, computes the answer

Challenges:

- $N$ very large & not known
- Data independent
- Stream ordered arbitrarily
- No random access to data

Main objective: space
Streaming Model

Motivation: monitoring latencies of requests

Streaming model = one pass over data & limited memory

Main objective: space

- No random access to data
- Stream ordered arbitrarily
- Data independent

Challenges:
- Very large $N$ not known
- At the end, computes the answer
- Uses memory sublinear in $N = \text{stream length}$
- Receives data in a stream, item by item

Streaming algorithm

How to summarize the input?
Selection Problem & Streaming

• Input: stream of \( N \) numbers
• Goal: find the \( k \)-th smallest
  
  • e.g.: the median, 99th percentile
  
  • \( O(N) \) time offline algorithm [Blum et al., '73]

Streaming restrictions:
• just one pass over the data
• limited memory: \( o(N) \)

No streaming algorithm for exact selection \( \Theta(N) \) space needed for median [Munro & Paterson '80, Guha & McGregor '07]

What about finding an approximate median?

Pavel Veselý
Tight Lower Bound for Quantile Summaries
Selection Problem & Streaming

• Input: stream of $N$ numbers
• Goal: find the $k$-th smallest
  e.g.: the median, 99th percentile

• $O(N)$ time offline algorithm [Blum et al., ’73]

• Streaming Restrictions:
  • Just one pass over the data
  • Limited memory: $o(N)$

• Goal: find the $k$-th smallest
  e.g.: the median, 99th percentile
  
• Input: stream of $N$ numbers

What about finding an approximate median?
Selection Problem & Streaming

- Input: stream of $N$ numbers
- Goal: find the $k$-th smallest
  - e.g.: the median, 99th percentile

$O(N)$ time offline algorithm [Blum et al., '73]

Streaming restrictions:
- limited memory: $o(N)$
- just one pass over the data

No streaming algorithm for exact selection

$\Omega(N)$ space needed to find the median

[$\mu$unro & Paterson, '80, Chara & McGregor, '07]

Selection Problem & Streaming

Pavel Veselý
Selection Problem & Streaming

Input: stream of $N$ numbers
Goal: find the $k$-th smallest
- e.g.: the median, 99th percentile

O($N$) time algorithm [Blum et al., '73]

No streaming algorithm for exact selection

$\Omega(N)$ space needed to find the median

Pavel Veseley
Tight Lower Bound for Quantile Summaries

What about finding an approximate median?

[Munro & Paterson, '80, Cukha & McGregor, '07]

No streaming algorithm for exact selection

- Limited memory: $o(N)$
- Just one pass over the data
- Streaming restrictions:

  - $O(N)$ time offline algorithm [Blum et al., '73]
  - e.g.: the median, 99th percentile
  - Goal: find the $k$-th smallest
  - Input: stream of $N$ numbers

Selection Problem & Streaming
How to define an approximate median?

-quantile = $d \cdot N$th smallest element

Median = .5-quantile

Quartiles = .25, .5, and .75-quantiles

Percentiles = .01, .02, . . . , .99-quantiles

-approximate $0$-quantile = any $0$-quantile for $0 = [0, 1]$

.01-approximate medians are .49- and .51-quantiles (and items in between)

-approximate selection:

query $k$-th smallest:
return $k$0-th smallest for $k0 = k \pm$
How to define an approximate median?

\[-\text{quantile} = \frac{d}{N} \cdot \text{\(n\)th smallest element}\]

\[\text{Median} = \frac{1}{2}-\text{quantile}\]

Quartiles = .25, .5, and .75-quantiles

Percentiles = .01, .02, . . . , .99-quantiles

How to define an approximate medians?

\[0.01-\text{approximate medians are} \quad .49-\text{and} \quad .51-\text{quantiles}\]

\[\text{approximate selection:} \quad \text{\(k\)-th smallest} \quad \Rightarrow \quad \text{return} \quad k\]

Median = \(0.5\)-quantile

\[\phi = \exists \phi \in [0, 1]\]

\[N \cdot \phi \Rightarrow \text{\(N\)-th smallest element}\]

Tight Lower Bound for Quantile Summaries

Pavel Veselý
How to define an approximate median?

\[ \text{-quantile} = d \cdot \frac{N}{n} \text{-th smallest element} \]

- Median = 0.5-quantile
- Quartiles = 0.25, 0.5, and 0.75-quantiles
- Percentiles = 0.01, 0.02, ..., 0.99-quantiles

Approximate selection:

- Query \( k \)-th smallest element:
  - Return \( k \) \( \frac{N}{n} \)-th smallest for \( k \) \( \frac{N}{n} \) = \( k \pm \frac{N}{n} \)

Note: sort data & select \( \frac{N}{n} \) items
Approximate Median & Quantiles

How to define an approximate median?

\[ \approx\cdot N \in \text{median for } \approx N \cdot \text{quantile} = \text{any-quantile} \]

- Quantiles = 0.25, 0.5, and 0.75-quantiles
- Percentiles = 0.01, 0.02, ..., 0.99-quantiles
- 01-approximate medians are 49- and 51-quantiles (and items in between)

Approximate Median & Quantiles
Approximate Median & Quantiles

- **How to define an approximate median?**

\[ q \] query \( k \)-th smallest \( k = N \) \( \frac{k}{2} \)th smallest for \( k \rightleftharpoons N \) \( \frac{k}{2} \)

- **Approximate selection:**

\[ \phi \] \( \epsilon \)-approximate selection:

\( \epsilon \)-approximate medians are \( 49 \)- and \( 51 \)-quantiles (and items in between)

\[ \left[ 3 + \phi, 3 - \phi \right] = \phi \] any \( \phi \)-quantile for \( \phi \)

- **Percentiles =** 01, 02, 03, 04, \( 99 \)-quantiles

- **Quartiles =** 25, 50, 75-quantiles

- **Median =** 50-quantile

(How to define an approximate median?)

Sorted data

Sorted data

25-quantile

 Median

75-quantile

Approximate Median & Quantiles
Approximate Median & Quantiles

How to define an approximate median?

\[ \text{approximate } \phi \text{-quantile} = \text{any } \phi \text{-quantile for } \phi = \left[ 0, 1 \right] \]

- 0.01-approximate medians are 0.49- and 0.51-quantiles (and items in between)
- 0.1-approximate medians are 0.49- and 0.51-quantiles
- Percentiles are 0.01, 0.02, \ldots, 0.99-quantiles
- Quartiles are 0.25, 0.5, and 0.75-quantiles
- Median = 0.5-quantile

Approximate Median & Quantiles
Approximate Quantile Summaries

Data structure with two operations:

- **Update**($x$):
  
  $x = \text{new item from the stream}$

- **Quantile Query**($Q$):
  
  For $Q \in [0, 1]$, return $\approx \text{quantile}$

Additional operations:

- **Rank Query**($x$):
  
  For item $x$, determine its rank in the order of the input

- **Merge of two quantile summaries**
  
  Preserve space bounds, while maintaining accuracy

Quantile summaries for:

- Approximating distributions
- Equi-depth histograms
- Streaming Bin Packing

Bottom line: Finding $\approx$-approximate median in data streams
Data structure with two operations:

- **Update**: \( x \rightarrow n \) with item \( x \) from the stream
- **Quantile Query** (for \( 0 < \phi < 1 \)): Return \( \phi \)-quantile

Additional operations:
- **Rank Query**: For item \( x \), determine its rank in the order of input
- **Merge of two quantile summaries**: Preserve space bounds, while maintaining accuracy

Quantile summaries are used in streaming algorithms for:

- Approximating distributions
- Equi-depth histograms
- Streaming Bin Packing

Bottom line: Finding \( \varepsilon \)-approximate median in data streams
Approximate Quantile Summaries

Data structure with two operations:

1. \( \text{Update}(x) \): \( x = \text{new item from the stream} \)
2. \( \text{Quantile Query}(\phi) \): For \( \phi \in [0, 1] \), return \( \phi \)-approximate quantiles

Additional operations:

1. \( \text{Rank Query}(x) \): For item \( x \), determine its rank = position in the ordering of the input

Bottom line: Finding \( \epsilon \)-approximate median in data streams
Approximate Quantile Summaries

Data structure with two operations:

- \textbf{Update}(x): x = \text{new item from the stream}
- \textbf{Quantile Query}(\phi): \text{for } \phi \in [0,1], \text{return } \phi\text{-quantile}

Additional operations:

- \textbf{Rank}(x): \text{for item } x, \text{determine its rank = position in the ordering of the input}
- \textbf{Merge} of two quantile summaries
- \text{Preserve space bounds, while maintaining accuracy}

Bottom line: Finding \( \varepsilon \)-approximate median in data streams
Approximate Quantile Summaries

Data structure with two operations:

- \text{Update}(x)\: x = \text{new item from the stream}
- \text{Quantile Query}(\phi)\: \phi \in [0, 1], \text{return } \phi\text{-approximate-quantile for } x

Additional operations:

- \text{Rank Query}(x)\: \text{for item } x\text{, determine its rank in the ordering of the input}
- \text{Merge of two quantile summaries}\: \text{preserve space bounds, while maintaining accuracy}
- \text{Quantile summaries streaming algorithms for:}
  - Approximating distributions
  - Equi-depth histograms
  - Streaming Bin Packing

Bottom line: Finding $\epsilon$-approximate median in data streams
Approximate Quantile Summaries

Data structure with two operations:

- Update\( (x) \): \( x \) = new item from the stream
- Quantile\( \phi \)\-Query\( (x) \): For \( \phi \) \epsilon [0, 1], return \( \phi \)-

Additional operations:

- Rank\( \) Query\( (x) \):
- Quantile\( \phi \)\-Query\( (x) \):
- Merge of two quantile summaries
- Preserve space bounds, while maintaining accuracy
- For item \( x \), determine its rank = position in the ordering of the input

Quantile summaries \rightleftharpoons \) streaming algorithms for:

- Approximating distributions
- Equi-depth histograms
- Streaming Bin Packing
- Guarding histograms
- Approximating distributions

Quantile summaries

Bottom line: Finding \( \varepsilon \)-approximate median in data streams
Approximate Median & Quantiles: Streaming Algorithms

State-of-the-art results

\[ \# \text{of stored items} \approx \ln N \]

\[ \# \text{of stored items} \approx \ln M \]

Many more papers: Munro & Paterson '80, Manku et al. '98, Manku et al. '99, Hung & Ting '10, Agarwal et al. '12, Wang et al. '13, Felber & Ostrovsky '15, . . .
Approximate Median & Quantiles: Streaming Algorithms

State-of-the-art results

\[ \text{space} \propto \# \text{of stored items} \]

Tight lower bound for quantile summaries

\[ \left( \frac{N^{3/2} \log \frac{3}{\varepsilon}}{I} \right) \cdot O \]

Maintains a subset of items + bounds on their ranks

[Greenwald & Khanna '01]

Deterministic comparison-based

[Shrivastava et al. '04]

Not for floats or strings

[Karnin et al. '16]

Randomized

Const. probability of violating \( \pm N \) error guarantee

Many more papers:

[Munro & Paterson '80, Manku et al. '98, Manku et al. '99, Hung & Ting '10, Agarwal et al. '12, Wang et al. '13, Felber & Ostrovsky '15, ...]
Approximate Median & Quantiles: Streaming Algorithms

State-of-the-art results

- \( \mathcal{O} \left( \frac{N \log \log \frac{3}{\epsilon}}{1} \right) \) space of stored items

Many more papers: [Munro & Paterson '80, Manku et al. '98, Manku et al. '99, Hung & Ting '10, Agarwal et al. '12, Wang et al. '13, Felber & Ostrovsky '15, ...]

\[ \text{State-of-the-art results} \]

\[ \text{not for floats or strings} \]

[Shrivastava et al. '04]

\[ \mathcal{M} \cup \{1, \ldots, M\} \] deterministic for integers

\[ \mathcal{W} \cup \{1, \ldots, W\} \] deterministic comparison-based

\[ \mathcal{G} \text{ & Khanna '01} \]
Approximate Median & Quantiles: Streaming Algorithms

State-of-the-art results

- \( O \left( \frac{\epsilon}{\epsilon} \cdot \log N \right) \) for deterministic comparison-based [Greenwald & Khanna '01]
- \( O \left( \frac{\epsilon}{\epsilon} \cdot \log \frac{1}{\epsilon} \right) \) for randomized [Karnin et al. '16]
- \( O \left( \frac{\epsilon}{\epsilon} \cdot \log \frac{1}{\epsilon} \right) \) not for floats or strings [Shrivastava et al. '04]
- \( O \left( \frac{\epsilon}{\epsilon} \cdot \log \frac{1}{\epsilon} \right) \) for integers

Maintains a subset of items + bounds on their ranks

- \( O \left( \frac{\epsilon}{\epsilon} \cdot \log \frac{1}{\epsilon} \right) \) for deterministic comparison-based

Many more papers: [Munro & Paterson '80, Manku et al. '98, Manku et al. '99, Hung & Ting '10, Agarwal et al. '12, Wang et al. '13, Felber & Ostrovsky '15, . . . ]

Space of stored items \# \sim \neq
Many more papers: [Munro & Paterson '80, Manku et al. '98, Manku et al. '99, Greenwald & Khanna 2001, ...]

- Approximate Median & Quantiles: Streaming Algorithms

- State-of-the-art bounds for quantile summaries

- Many more papers: [Hung & Ting '10, Agarwal et al. '12, Wang et al. '13, Felder & Ostrovsky '15, ...]

- Approximate Median & Quantiles: Streaming Algorithms

- State-of-the-art bounds for quantile summaries

- Many more papers: [Munro & Paterson '80, Manku et al. '98, Manku et al. '99, Greenwald & Khanna 2001, ...]

- Approximate Median & Quantiles: Streaming Algorithms

- State-of-the-art bounds for quantile summaries

- Many more papers: [Hung & Ting '10, Agarwal et al. '12, Wang et al. '13, Felder & Ostrovsky '15, ...]
Approx. Median & Quantiles: Is There a "Perfect" Algorithm?

What would be a "perfect" streaming algorithm?

- finds 
- deterministic
- constant space for fixed 
- ideally 

\[ O(1) \]

- no additional knowledge about items
- comparison-based

Theorem (Cormode, V. ‘20)

There is no perfect streaming algorithm for 

- Optimal space lower bound 

\[ \Omega \left( \frac{1}{\varepsilon} \cdot \log N \right) \]

- Matches the result in [Greenwald & Khanna ‘01]
Approx. Median & Quantiles: Is There a "Perfect" Algorithm?

What would be a "perfect" streaming algorithm?

- deterministic
- finds ε-approximate median
- optimal space lower bound $\Omega\left(\frac{1}{\epsilon^2} \cdot \log N\right)$
  - Matches the result in [Greenwald & Khanna '01]

Tight Lower Bound for Quantile Summaries

Pavel Veselý
What would be a "perfect" streaming algorithm?

- finds \( \varepsilon \)-approximate median
- deterministic
- constant space for fixed \( \varepsilon \)

**Theorem** (Cormode, V.'20)

There is no perfect streaming algorithm for \( \varepsilon \)-approximate median.

Optimal space lower bound \( \Omega \left( \varepsilon^3 \right) \cdot \log_\varepsilon N \) matches the result in [Greenwald & Khanna '01].
What would be a “perfect” streaming algorithm?

• finds "\(\varepsilon\)-approximate median"
• deterministic
• constant space for fixed \(\varepsilon\)
• no additional knowledge about items
• ideally \(O\left(\frac{\varepsilon^2}{\varepsilon^3}I\right)\) or e.g. \(O\left(\frac{\varepsilon^3}{\varepsilon^4}I\right)\)

What would be a “perfect” streaming algorithm?

Approx. Median & Quantiles: Is There a “Perfect” Algorithm?
Approx. Median & Quantiles: Is There a "Perfect" Algorithm?

What would be a "perfect" streaming algorithm for ε-approximate median?

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deterministic</td>
<td>no additional knowledge about items</td>
</tr>
<tr>
<td>constant space for fixed ε</td>
<td>constant space for fixed ε</td>
</tr>
<tr>
<td>ε-approximate median</td>
<td>finds ε-approximate median</td>
</tr>
<tr>
<td>comparison-based</td>
<td>no additional knowledge about items</td>
</tr>
</tbody>
</table>

Theorem (Cormode, V., 20)

There is no perfect streaming algorithm for ε-approximate median.

Optimal space lower bound

\[ \Omega \left( \frac{\varepsilon^2}{1} \right) \]

\( \Omega \left( \frac{\varepsilon^3}{1} \right) \)

\( \Omega \left( \frac{\varepsilon^3}{1} \right) \)

What would be a "perfect" streaming algorithm?
What would be a “perfect” streaming algorithm?

- finds \( \varepsilon \)-approximate median
- deterministic
- constant space for fixed \( \varepsilon \)
- ideally \( O\left(\frac{3}{\varepsilon} \cdot \log N\right) \)
- no additional knowledge about items
- optimal space lower bound \( \Omega\left(\frac{3}{\varepsilon} \cdot \log N\right) \)

There is no perfect streaming algorithm for \( \varepsilon \)-approximate median.

**Theorem (Cormode, V., ’20)**

This matches the result in [Greenwald & Khanna, 01].

\[
\left(\frac{3}{\varepsilon} \cdot \log N\right) \geq \Omega\left(\frac{3}{\varepsilon} \cdot \log N\right)
\]

What would be a “perfect” streaming algorithm?
Approx. Median & Quantiles: Lower Bound Idea

Comparison-based algorithm cannot compare with items deleted from the memory.

Idea: Introduce uncertainty
- too high uncertainty
  - not accurate enough answers
- need to show: low uncertainty
  - many items stored
  - large space needed

Recursive construction of worst-case stream

Lower bound $\Omega \left( \frac{1}{\epsilon} \cdot \log N \right)$
Comparison-based algorithm cannot compare with items deleted from the memory.

Idea: Introduce uncertainty
• too high uncertainty not accurate enough answers
• need to show: low uncertainty many items stored large space needed!

Recursive construction of worst-case stream ! lower bound \( \Omega \) · \( \log \) \( N \)
Comparison-based algorithm cannot compare with items deleted from memory.

Idea: Introduce uncertainty

- too high uncertainty
- not accurate enough answers

- need to show: low uncertainty
- many items stored
- large space needed

Recursive construction of worst-case stream

Lower bound $\Omega(1) \cdot \log N$.

---

Approx. Median & Quantiles: Lower Bound Idea

How does 30 compare to discarded items between 10 and 50?
Approx. Median & Quantiles: Lower Bound

Comparison-based algorithm cannot compare with items deleted from the memory.

Idea: Introduce uncertainty

• too high uncertainty
  not accurate-enough answers
• need to show: low uncertainty
  many items stored
  large space needed

Recursive construction of worst-case stream

Lower bound ⌦ 1 \cdot \log N

How does 30 compare to discarded items between 10 and 50?

new item: 30

\[ \text{cannot compare with items deleted from the memory} \]
Approx. Median & Quantiles: Lower Bound

Comparison-based algorithm

市场经济
cannot compare with items deleted from the memory

Idea: Introduce uncertainty

• too high uncertainty
  leads to inaccurate enough answers

How does 30 compare to discarded items between 10 and 50?

new item: 30

Recursive construction of worst-case stream

lower bound $\Omega(1 + \log N)$

Pavel Veselý
Tight Lower Bound for Quantile Summaries
9/10
Approx. Median & Quantiles: Lower Bound Ideas

Comparison-based algorithm

- cannot compare with items deleted from the memory

How does 30 compare to discarded items between 10 and 50?

Idea: Introduce uncertainty

- too high uncertainty → not accurate enough answers
- need to show: low uncertainty → large space needed

new item: 30

Comparison-based algorithm

Approx. Median & Quantiles: Lower Bound Ideas
Approx. Median & Quantiles: Lower Bound Idea

Comparison-based algorithm cannot compare with items deleted from the memory

How does 30 compare to discarded items between 10 and 50?

new item: 30

Idea: Introduce uncertainty

• too high uncertainty not accurate enough answers

• need to show: low uncertainty many items stored large space needed

Recursive construction of worst-case stream lower bound $\mathcal{N}$

$\left( 2 \cdot \log \frac{N}{I} \right)$
Approximating Median & Quantiles: Conclusions & Open Problems

Problem solved:

- Deterministic algorithms: space $\Theta \left( \frac{1}{\epsilon} \cdot \log \frac{1}{\eta} \right)$ optimal
  - [Greenwald & Khanna '01]
  - Cormode, V. '20

- Randomized algorithms: space $\Theta \left( \frac{1}{\epsilon} \right)$ optimal
  - [Karnin et al.'16]
  - Cormode, V. '20

Future work:

- Figure out constant factors
- Randomized algorithm with good expected space, but guaranteed $\pm \epsilon N$ error
- An non-trivial lower bound for $\{1, \ldots, M\}$
- Or can we do better than $O \left( \frac{1}{\epsilon} \cdot \log M \right)$?
- Dynamic streams w/ insertions and deletions of items

Pavel Veselý
Tight Lower Bound for Quantile Summaries

10 / 10
Approximating Median & Quantiles: Conclusions & Open Problems

Problem solved:

• Deterministic algorithms: space \( \Theta \left( \frac{3}{1} \cdot \log \frac{N}{\varepsilon} \right) \) optimal

[Greenwald & Khanna '01]

[10]

[Cormode, V. '20]

Randomized algorithms: space \( \Theta \left( \frac{3}{1} \right) \) optimal

Future work:

• Figure out constant factors

• Randomized algorithms with good expected space, but guaranteed ± \( \varepsilon \cdot \log M \)

[Greenwald & Khanna, 01]

• Dynamic streams with insertions and deletions of items

Pavel Veselý
Tight Lower Bound for Quantile Summaries

10 / 10
Approximating Median & Quantiles: Conclusions & Open Problems

Problem solved:
- Deterministic algorithms:
  \[ \text{space} \propto \Theta \left( \frac{1}{3} \cdot \log N \right) \]

- Randomized algorithms:
  \[ \text{space} \propto \Theta \left( \frac{1}{3} \cdot \log N \right) \]

Future work:
- Figure out constant factors
- Randomized algorithm with good expected space but guaranteed finite error
- Deterministic algorithms: space \( \Theta \left( \frac{1}{3} \cdot \log M \right) \)
- An on-trivial lower bound for integers \( \{1, \ldots, M\} \)
- Or can we do better than \( O \left( \frac{1}{3} \cdot \log M \right) \)?
- Dynamic streams with insertions and deletions of items

References:
- Karnin et al., 2016
- Cormode, V., 2020
- Greenwald & Khanna, 2001

Pavel Veselý
Tight Lower Bound for Quantile Summaries
Approximating Median & Quantiles: Conclusions & Open Problems

Problem solved:
- Deterministic algorithms: space $\Theta\left(\frac{3^\epsilon}{\epsilon} \cdot \log \frac{1}{\epsilon}\right)$
- Randomized algorithms: space $\Theta\left(\frac{3^\epsilon}{\epsilon} \cdot \log \frac{1}{\epsilon}\right)$

Future work:
- Figure out constant factors
- Randomized algorithm with good expected space, but guaranteed $\pm \epsilon N$ error
- A non-trivial lower bound for integers, but guaranteed $\pm \epsilon N$

Open Problems:
- Tight lower bound for quantile summaries

References:
- Greenwald & Khanna, '01
- Cormode, '20
- Karnin et al., '16
Approximating Median & Quantiles: Conclusions & Open Problems

Problem solved:
- Deterministic algorithms: space $\mathcal{O} \left( \frac{N^\varepsilon \cdot \log^\gamma}{1} \right)$ optimal
  - Greenwald & Khanna '01
- Randomized algorithms: space $\Theta(\mathcal{O} \left( \frac{N^\varepsilon \cdot \log^\gamma}{1} \right))$
  - Karnin et al. '16

Future work:
- Figure out constant factors
- Randomized algorithm with good expected space, but guaranteed $\pm N$ error
- A non-trivial lower bound for integers $\left\{ 1, \ldots, M \right\}$
- Dynamic streams w/ insertions and deletions of items
Thank You!