Building Wavelet Histograms on Large Data in MapReduce

Jeffrey Jestes\textsuperscript{1} Ke Yi\textsuperscript{2} Feifei Li\textsuperscript{1}

\textsuperscript{1}School of Computing, University of Utah

\textsuperscript{2}Department of Computer Science, Hong Kong University of Science and Technology

November 16, 2011
For large data we often wish to obtain a concise summary.
For large data we often wish to obtain a concise summary.
For large data we often wish to obtain a concise summary.
1 Introduction and Motivation
   • Histograms
     • MapReduce and Hadoop

2 Exact Top-k Wavelet Coefficients
   • Naive Solution
   • Hadoop Wavelet Top-k: Our Efficient Exact Solution

3 Approximate Top-k Wavelet Coefficients
   • Linearly Combinable Sketch Method
   • Our First Sampling Based Approach
   • An Improved Sampling Approach
   • Two-Level Sampling

4 Experiments

5 Conclusions
   • Hadoop Wavelet Top-k in Hadoop
A widely accepted and utilized summarization tool is the Histogram.

Let $A$ be an attribute of dataset $R$. Values of $A$ are drawn from finite domain $\{1, \ldots, u\}$. Define for each $x \in \{1, \ldots, u\}$, $v(x) = \{\text{count}(R.A = x)\}$. Define frequency vector $v$ of $R.A$ as $v = (v(1), \ldots, v(u))$. A histogram over $R.A$ is any compact (lossy) representation of $v$. 

<table>
<thead>
<tr>
<th>Record ID</th>
<th>User ID</th>
<th>Object ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>12872</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>19832</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>231</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$v(x) = (3, 5, 10, 8, 2, 2, 10, 14)$
Introduction: Histograms

- A widely accepted and utilized summarization tool is the **Histogram**.
- Let $A$ be an attribute of dataset $R$. 

---

<table>
<thead>
<tr>
<th>Record ID</th>
<th>User ID</th>
<th>Object ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>12872</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>19832</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>231</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$$v(x) = \begin{cases} 
3 & \text{if } x = 1 \\
5 & \text{if } x = 2 \\
10 & \text{if } x = 3 \\
8 & \text{if } x = 4 \\
2 & \text{if } x = 5 \\
2 & \text{if } x = 6 \\
10 & \text{if } x = 7 \\
14 & \text{if } x = 8 
\end{cases}$$
A widely accepted and utilized summarization tool is the **Histogram**.

Let $A$ be an attribute of dataset $R$.

Values of $A$ are drawn from finite domain $[u] = \{1, \ldots, u\}$. 
A widely accepted and utilized summarization tool is the **Histogram**.

- Let $A$ be an attribute of dataset $R$.
  - Values of $A$ are drawn from finite domain $[u] = \{1, \ldots, u\}$.
  - Define for each $x \in \{1, \ldots, u\}$, $v(x) = \{\text{count}(R.A = x)\}$. 
A widely accepted and utilized summarization tool is the **Histogram**.

- Let $A$ be an attribute of dataset $R$.
  - Values of $A$ are drawn from finite domain $[u] = \{1, \cdots, u\}$.
- Define for each $x \in \{1, \cdots, u\}$, $v(x) = \{\text{count}(R.A = x)\}$.
- Define frequency vector $v$ of $R.A$ as $v = (v(1), \ldots, v(u))$. 

<table>
<thead>
<tr>
<th>Record ID</th>
<th>User ID</th>
<th>Object ID</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>12872</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>19832</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>231</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$v(x)$

1 2 3 4 5 6 7 8
Introduction: Histograms

- A widely accepted and utilized summarization tool is the **Histogram**.
  - Let $A$ be an attribute of dataset $R$.
    - Values of $A$ are drawn from finite domain $[u] = \{1, \cdots, u\}$.
  - Define for each $x \in \{1, \cdots, u\}$, $v(x) = \{\text{count}(R.A = x)\}$.
  - Define frequency vector $v$ of $R.A$ as $v = (v(1), \ldots, v(u))$.
  - A **histogram** over $R.A$ is any compact (lossy) representation of $v$. 

<table>
<thead>
<tr>
<th>Record ID</th>
<th>User ID</th>
<th>Object ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>12872</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>19832</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>231</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$v(x) = (3, 5, 10, 8, 2, 2, 10, 14)$

Jeffrey Jestes, Ke Yi, Feifei Li
A widely accepted and utilized summarization tool is the **Histogram**.

- Let $A$ be an attribute of dataset $R$.
- Values of $A$ are drawn from finite domain $[u] = \{1, \cdots, u\}$.
- Define for each $x \in \{1, \cdots, u\}$, $v(x) = \{\text{count}(R.A = x)\}$.
- Define frequency vector $v$ of $R.A$ as $v = (v(1), \ldots, v(u))$.
- A **histogram** over $R.A$ is any compact (lossy) representation of $v$.
A common choice for a histogram is the Haar wavelet histogram.
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:
Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

Original data signal at level $\ell = \log_2 u$.

![Wavelet Histogram Diagram]

$\ell = 3$

$v(1)$  $v(2)$  $v(3)$  $v(4)$  $v(5)$  $v(6)$  $v(7)$  $v(8)$

Original data signal at level $\ell = \log_2 u$. 

---

A common choice for a histogram is the **Haar wavelet histogram**.

We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

Compute **detail coefficients** $c_i$ and **average coefficients** $a_i$ for level $\ell = 2$.
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients \( w_i \) recursively as follows:

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\ell = 3 \\
\ell = 2 \\
\ell = 1 \\
\end{array}
\]

Compute *detail coefficients* \( c_i \) and *average coefficients* \( a_i \) for level \( \ell = 2 \).
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

$\ell = 3$

$\ell = 2$

\[ a_4 = \frac{(v(2) + v(1))}{2} \]

Compute *detail coefficients* $c_i$ and *average coefficients* $a_i$ for level $\ell = 2$. 

\[ a_4 = \frac{(v(2) + v(1))}{2} \]
A common choice for a histogram is the Haar wavelet histogram.

We obtain the Haar wavelet coefficients \( w_i \) recursively as follows:

\[
\begin{align*}
\ell &= 3 \\
x & 1 2 3 4 5 6 7 8 \\
v(x) & 3 5 10 8 2 2 10 14 \\
\downarrow & \\
& \\
c_4 &= (v(2) - v(1))/2
\end{align*}
\]

Compute detail coefficients \( c_i \) and average coefficients \( a_i \) for level \( \ell = 2 \).
Introduction: Wavelet Histograms

- A common choice for a histogram is the *Haar wavelet histogram*.
- We obtain the Haar wavelet coefficients \( w_i \) recursively as follows:

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
v(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\ell = 3 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\ell = 2 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\ell = 1 \\
\end{array}
\]

Compute *detail coefficients* \( c_i \) and *average coefficients* \( a_i \) for level \( \ell = 1 \).
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

$$
\begin{array}{c|cccccccc}
\text{x} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
\text{v(x)} & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
$$

We compute the **detail coefficients** $c_i$ and **average coefficients** $a_i$ for level $\ell = 1$. 

Compute the detail coefficients $c_i$ and average coefficients $a_i$ for level $\ell = 1$. 

---

A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients \( w_i \) recursively as follows:

<table>
<thead>
<tr>
<th>( x )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v(x) )</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

\[
a_2 = \frac{a_5 + a_4}{2}
\]

Compute *detail coefficients* \( c_i \) and *average coefficients* \( a_i \) for level \( \ell = 1 \).
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

Compute *detail coefficients* $c_i$ and *average coefficients* $a_i$ for level $\ell = 1$.

$$c_2 = (a_5 - a_4)/2$$
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

\[
\begin{array}{cccccccc}
  x & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
  v(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
  c_1 & 0.3 & 6.8 & a_1 \\
  c_2 & 2.5 & 6.5 & a_2 \\
  c_3 & 5 & 7 & a_3 \\
  c_4 & 1 & 4 & a_4 \\
  c_5 & -1 & 9 & a_5 \\
  c_6 & 0 & 2 & a_6 \\
  c_7 & 2 & 12 & a_7 \\
\end{array}
\]

Compute *detail coefficients* $c_i$ and *average coefficients* $a_i$ for level $\ell = 0$. 

---

A common choice for a histogram is the **Haar wavelet histogram**. We obtain the Haar wavelet coefficients \( w_i \) recursively as follows:

\[
\begin{array}{cccccccc}
  x & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
  v(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

![Diagram of wavelet coefficients and averages](image)

Compute **detail coefficients** \( c_i \) and **average coefficients** \( a_i \) for level \( \ell = 0 \).
A common choice for a histogram is the Haar wavelet histogram. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

\begin{align*}
  x & = 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \\
  v(x) & = 3 \quad 5 \quad 10 \quad 8 \quad 2 \quad 2 \quad 10 \quad 14
\end{align*}

\begin{align*}
  a_1 & = 6.8 \quad \text{total average} \\
  c_1 & = 0.3 \quad 6.8 \\
  c_2 & = 2.5 \quad 6.5 \\
  c_3 & = 5 \quad 7 \\
  c_4 & = 1 \quad 4 \\
  c_5 & = -1 \quad 9 \\
  c_6 & = 0 \quad 2 \\
  c_7 & = 2 \quad 12
\end{align*}

\begin{align*}
  a_2 & = \frac{a_3 + a_7}{2} \\
  a_3 & = \text{total average} \\
  a_4 & = \frac{a_5 + a_6}{2} \\
  a_5 & = \text{total average} \\
  a_6 & = \frac{a_7}{2} \\
  a_7 & = \frac{a_4 + a_5}{2}
\end{align*}

Compute detail coefficients $c_i$ and average coefficients $a_i$ for level $\ell = 0$. 

\[ \text{Jeffrey Jestes, Ke Yi, Feifei Li} \] 

\[ \text{Building Wavelet Histograms on Large Data in MapReduce} \]
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

\[
\begin{array}{cccccccc}
  x & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
  v(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

\[
\begin{array}{cccc}
  & c_1 & 0.3 & 6.8 & a_1 \\
  & c_2 & 2.5 & 6.5 & a_2 \\
  c_4 & 1 & 4 & a_4 & c_5 & -1 & 9 & a_5 & c_6 & 0 & 2 & a_6 & c_7 & 2 & 12 & a_7 \\
  v(1) & 3 & v(2) & 5 & v(3) & 10 & v(4) & 8 & v(5) & 2 & v(6) & 2 & v(7) & 10 & v(8) & 14 \\
\end{array}
\]

Compute *detail coefficients* $c_i$ and *average coefficients* $a_i$ for level $\ell = 0$. 

\[c_1 = (a_3 - a_2)/2\]
A common choice for a histogram is the **Haar wavelet histogram**.

We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

The **wavelet coefficients** $w_i$ are $[a_1, c_1, \ldots, c_{u-1}]$. 

A common choice for a histogram is the Haar wavelet histogram. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

\[
\begin{array}{cccccccc}
\ell &=& 0 & & & & & \\
3 & & & & & & & \\
5 & & & & & & & \\
10 & & & & & & & \\
8 & & & & & & & \\
2 & & & & & & & \\
2 & & & & & & & \\
10 & & & & & & & \\
14 & & & & & & & \\
\end{array}
\]

The wavelet coefficients $w_i$ are $[a_1, c_1, \ldots, c_{u-1}]$. 

\[ x \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \]

\[ v(x) \quad 3 \quad 5 \quad 10 \quad 8 \quad 2 \quad 2 \quad 10 \quad 14 \]
A common choice for a histogram is the *Haar wavelet histogram*.

We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

The wavelet coefficients $w_i$ are $[a_1, c_1, \ldots, c_{u-1}]$. 

---

**References:**


A common choice for a histogram is the *Haar wavelet histogram*.

We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

One scales $w_i$ by $\sqrt{u/2^\ell}$ to preserve energy, i.e.

$$
\|v\|_2^2 = \sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2
$$
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

scale by: $\sqrt{u/2^\ell}$

energy: $\sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2$

One scales $w_i$ by $\sqrt{u/2^\ell}$ to preserve energy, i.e.

$\|v\|_2^2 = \sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2$
A common choice for a histogram is the **Haar wavelet histogram**. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

$$
\begin{array}{cccccccc}
  x & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
  \mathbf{v}(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
$$

scale by: $\sqrt{u/2^\ell}$

energy: $\sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2$

One scales $w_i$ by $\sqrt{u/2^\ell}$ to preserve energy, i.e.

$$
\|\mathbf{v}\|_2^2 = \sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2
$$
A common choice for a histogram is the **Haar wavelet histogram**. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

scale by: $\sqrt{u/2^\ell}$

energy: $\sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2$

Select top-$k$ $w_i$ in the **absolute value** to obtain best $k$-term representation minimizing error in energy, i.e. minimize $\sum_{i=1}^{u} v(i)^2 - \sum_{i=1}^{u} w_i^2$
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

scale by: $\sqrt{u/2^{\ell}}$

energy: $\sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2$

$k = 3$

$\ell = 0$

$\ell = 1$

$\ell = 2$

$\ell = 3$

Select top-$k$ $w_i$ in the absolute value to obtain best $k$-term representation minimizing error in energy, i.e. minimize $\sum_{i=1}^{u} v(i)^2 - \sum_{i=1}^{u} w_i^2$
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

scale by: $\sqrt{\frac{u}{2^\ell}}$

energy: $\sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2$

$k = 3$

We maintain the best $k$-term $w_i$. Other $w_i$ are treated as 0.
A common choice for a histogram is the **Haar wavelet histogram**.

We obtain the Haar wavelet coefficients \(w_i\) recursively as follows:

\[
\begin{align*}
\text{scale by:} & \quad \sqrt{\frac{u}{2^\ell}} \\
\text{energy:} & \quad \sum_{i=1}^{u} v(i)^2 = \sum_{i=1}^{u} w_i^2
\end{align*}
\]

The error in energy is
\[
\sum_{i=1}^{u} v(i)^2 - \sum_{i=1}^{u} w_i^2 = 502 - 489.5 = 12.5.
\]
A common choice for a histogram is the *Haar wavelet histogram*.
We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

scale by: $\sqrt{u/2^\ell}$
energy: $\sum_{i=1}^u v(i)^2 = \sum_{i=1}^u w_i^2$

To reconstruct the original signal we compute the *average* and *difference coefficients* in reverse, i.e. top to bottom.
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

The reconstructed signal is a reasonably close approximation.
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

All $w_i$ can be calculated in $O(u \log u)$ time:
1. We maintain $O(\log u)$ partial $w_i$s at a time.
2. Compute affected $w_i$ and contribution from each $v(x)$ in $O(\log u)$ time.
2. Process $v(x)$s in sorted order. [GKMS01]
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

All $w_i$ can be calculated in $O(u \log u)$ time:
1. We maintain $O(\log u)$ partial $w_i$s at a time.
2. Compute affected $w_i$ and contribution from each $v(x)$ in $O(\log u)$ time.
2. Process $v(x)$s in sorted order. [GKMS01]
A common choice for a histogram is the Haar wavelet histogram. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

$$\begin{array}{cccccccc}
x & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
v(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}$$

All $w_i$ can be calculated in $O(u \log u)$ time:
1. We maintain $O(\log u)$ partial $w_i$s at a time.
2. Compute affected $w_i$ and contribution from each $v(x)$ in $O(\log u)$ time.
2. Process $v(x)$s in sorted order. [GKMS01]
A common choice for a histogram is the *Haar wavelet histogram.*

We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

All $w_i$ can be calculated in $O(u \log u)$ time:

1. We maintain $O(\log u)$ partial $w_i$s at a time.
2. Compute affected $w_i$ and contribution from each $v(x)$ in $O(\log u)$ time.
3. Process $v(x)$s in sorted order. [GKMS01]

---

A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

![Diagram](image)

All $w_i$ can be calculated in $O(u \log u)$ time:
1. We maintain $O(\log u)$ partial $w_i$s at a time.
2. Compute affected $w_i$ and contribution from each $v(x)$ in $O(\log u)$ time.
3. Process $v(x)$s in sorted order. [GKMS01]
A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

\[
\begin{array}{cccccccc}
  x & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
  v(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
  w_1 & 6.8 & \text{total average} \\
  w_2 & 0.3 & 6.8 \\
  w_3 & 2.5 & 6.5 \\
  w_4 & 5 & 7 \\
  w_5 & 1 & 4 \\
  w_6 & -1 & 9 \\
  w_7 & 0 & 2 \\
  w_8 & 2 & 12 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
  v(1) & v(2) & v(3) & v(4) & v(5) & v(6) & v(7) & v(8) \\
  3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

All $w_i$ can be calculated in $O(u \log u)$ time:
1. We maintain $O(\log u)$ partial $w_i$s at a time.
2. Compute affected $w_i$ and contribution from each $v(x)$ in $O(\log u)$ time.
2. Process $v(x)$s in sorted order. [GKMS01]

A common choice for a histogram is the **Haar wavelet histogram**. We obtain the Haar wavelet coefficients \( w_i \) recursively as follows:

\[
\ell = 0
\]

\[
\ell = 1
\]

\[
\ell = 2
\]

\[
\ell = 3
\]

All \( w_i \) can be calculated in \( O(u \log u) \) time:
1. We maintain \( O(\log u) \) partial \( w_i \)'s at a time.
2. Compute affected \( w_i \) and contribution from each \( v(x) \) in \( O(\log u) \) time.
3. Process \( v(x) \)'s in sorted order. [GKMS01]
A common choice for a histogram is the Haar wavelet histogram. We obtain the Haar wavelet coefficients $w_i$ recursively as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

$w_1$ total average

$w_2$ $0.3$ $6.8$ $a_1$

$w_3$ $2.5$ $6.5$ $a_2$

$w_4$ $5$ $7$ $a_3$

$w_5$ $1$ $4$ $a_4$

$w_6$ $-1$ $9$ $a_5$

$w_7$ $0$ $2$ $a_6$

$w_8$ $2$ $12$ $a_7$

$w_i$ can be calculated in $O(u \log u)$ time:
1. We maintain $O(\log u)$ partial $w_i$s at a time.
2. Compute affected $w_i$ and contribution from each $v(x)$ in $O(\log u)$ time.
3. Process $v(x)$s in sorted order. [GKMS01]

A common choice for a histogram is the *Haar wavelet histogram*. We obtain the Haar wavelet coefficients \( w_i \) recursively as follows:

<table>
<thead>
<tr>
<th>( x )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v(x) )</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\ell &= 3 \\
\ell &= 2 \\
\ell &= 1 \\
\ell &= 0
\end{align*}
\]

All \( w_i \) can be calculated in \( O(u \log u) \) time:
1. We maintain \( O(\log u) \) partial \( w_i \)s at a time.
2. Compute affected \( w_i \) and contribution from each \( v(x) \) in \( O(\log u) \) time.
2. Process \( v(x) \)s in sorted order. [GKMS01]

We may also compute $w_i$ with the wavelet basis vectors $\psi_i$.

$w_i = v \cdot \psi_i$ for $i = 1, \ldots, u$
Outline

1 Introduction and Motivation
   - Histograms
   - MapReduce and Hadoop

2 Exact Top-$k$ Wavelet Coefficients
   - Naive Solution
   - Hadoop Wavelet Top-$k$: Our Efficient Exact Solution

3 Approximate Top-$k$ Wavelet Coefficients
   - Linearly Combinable Sketch Method
   - Our First Sampling Based Approach
   - An Improved Sampling Approach
   - Two-Level Sampling

4 Experiments

5 Conclusions
   - Hadoop Wavelet Top-$k$ in Hadoop
Introduction: MapReduce and Hadoop

Traditionally data is stored in a centralized setting.
Traditionally data is stored in a centralized setting. Now stored data has sky rocketed, and is increasingly distributed.
Traditionally data is stored in a centralized setting.
Now stored data has sky rocketed, and is increasingly distributed.
We study computing the top-$k$ coefficients efficiently on distributed data in MapReduce using Hadoop to illustrate our ideas.
• Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).
Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).

| Record ID | User ID | Object ID |...
|-----------|---------|-----------|---
| 1         | 1       | 12872     |...
| 2         | 8       | 19832     |...
| 3         | 4       | 231       |...
| ...       | ...     | ...       |...
Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).
Hadoop requires a Distributed File System (DFS), we utilize the Hadoop Distributed File System (HDFS).
Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.
Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.
Background: Hadoop Core

- Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.
- We assume one TaskTracker per physical machine.
Hadoop Core consists of one master *JobTracker* and several *TaskTrackers*.

We assume one TaskTracker per physical machine.
In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*. 
In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the *master*.

The other machines run DataNode and TaskTracker tasks and are called *slaves*.
In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the master.

The other machines run DataNode and TaskTracker tasks and are called slaves.

NameNode + JobTracker

DataNodes + TaskTrackers
In a Hadoop cluster one machine typically runs both the NameNode and JobTracker tasks and is called the master.

The other machines run DataNode and TaskTracker tasks and are called slaves.
Next we look at an overview of a typical MapReduce Job.
Job specific variables are first placed in the *Job Configuration* which is sent to each *Mapper Task* by the *JobTracker*.
Large data such as files or libraries are then put in the *Distributed Cache* which is copied to each *TaskTracker* by the *JobTracker*.
The JobTracker next assigns each InputSplit to a Mapper task on a TaskTracker, we assume $m$ Mappers and $m$ InputSplits.
Each Mapper maps a \((k_1, v_1)\) pair to an intermediate \((k_2, v_2)\) pair and partitions by \(k_2\), i.e. \(hash(k_2) = p_i\) for \(i \in [1, r]\), \(r = |reducers|\).
An optional **Combiner** is executed over \((k_2, \text{list}(v_2))\).
The Combiner aggregates $v_2$ for a $k_2$ and a $(k_2, v_2)$ is written to a partition on disk.
The JobTracker assigns two TaskTrackers to run the Reducers, each Reducer copies and sorts its inputs from corresponding partitions.
The JobTracker assigns two TaskTrackers to run the Reducers, each Reducer copies and sorts its inputs from corresponding partitions.
Each Reducer reduces a \((k_2, list(v_2))\) to a single \((k_3, v_3)\) and writes the results to a DFS file, \(o_i\) for \(i \in [1, r]\).
Outline

1. Introduction and Motivation
   - Histograms
   - MapReduce and Hadoop

2. Exact Top-$k$ Wavelet Coefficients
   - Naive Solution
   - Hadoop Wavelet Top-$k$: Our Efficient Exact Solution

3. Approximate Top-$k$ Wavelet Coefficients
   - Linearly Combinable Sketch Method
   - Our First Sampling Based Approach
   - An Improved Sampling Approach
   - Two-Level Sampling

4. Experiments

5. Conclusions
   - Hadoop Wavelet Top-$k$ in Hadoop
Each of the $m$ Mappers reads the input key $x$ from its input split.
Exact Top-\(k\) Wavelet Coefficients: Naive Solution

Each Mapper emits \((x, 1)\) for combining by the Combiner.
Each Combiner emits \((x, v_j(x))\), where \(v_j(x)\) is the local frequency of \(x\).
The Reducer utilizes a Centralized Wavelet Top-$k$ algorithm, supplying the $(x, v(x))$ in a streaming fashion.
At the end of the Reduce phase, the Reducer’s `close()` method is invoked. The Reducer then requests the top-$k$ $|w_i|$. 
The centralized algorithm computes the top-\(k\)|\(w_i|\) and returns these to the Reducer.
Finally, the Reducer writes the top-\(k\) \(|w_i|\) to its HDFS output file \(o_1\).
Exact Top-$k$ Wavelet Coefficients: Naive Solution

$n = \text{Total records in input file.}$

Too Expensive!!!

$O(n)$ Communication!!!
Outline

1 Introduction and Motivation
   - Histograms
   - MapReduce and Hadoop

2 Exact Top-$k$ Wavelet Coefficients
   - Naive Solution
   - Hadoop Wavelet Top-$k$: Our Efficient Exact Solution

3 Approximate Top-$k$ Wavelet Coefficients
   - Linearly Combinable Sketch Method
   - Our First Sampling Based Approach
   - An Improved Sampling Approach
   - Two-Level Sampling

4 Experiments

5 Conclusions
   - Hadoop Wavelet Top-$k$ in Hadoop
Exact Top-\(k\) Wavelet Coefficients: Our Solution

- We can try to model the problem as a distributed top-\(k\):
  \[ w_i = v \cdot \psi_i = (\sum_{j=1}^{m} v_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}. \]
Exact Top-$k$ Wavelet Coefficients: Our Solution

We can try to model the problem as a distributed top-$k$:

\[ w_i = v \cdot \psi_i = \left( \sum_{j=1}^{m} v_j \right) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}. \]

\[ w_{i,j} \] is the local value of \( w_i \) in split \( j \).
Exact Top-k Wavelet Coefficients: Our Solution

- We can try to model the problem as a distributed top-k:
  \[ w_i = v \cdot \psi_i = (\sum_{j=1}^{m} v_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}. \]
We can try to model the problem as a distributed top-k:

\[ w_i = \mathbf{v} \cdot \psi_i = (\sum_{j=1}^{m} v_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}. \]
Exact Top-$k$ Wavelet Coefficients: Our Solution

- We can try to model the problem as a distributed top-$k$:
  \[ w_i = v \cdot \psi_i = (\sum_{j=1}^{m} v_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}. \]
- Previous solutions assume local score $s_{i,j} \geq 0$ and want the largest $s_i = \sum_{j=1}^{m} s_{i,j}$.

\[ w_i = \sum_{j=1}^{m} w_{i,j} \]
We can try to model the problem as a distributed top-$k$:

$$w_i = v \cdot \psi_i = (\sum_{j=1}^{m} v_j) \cdot \psi_i = \sum_{j=1}^{m} w_{i,j}.$$ 

Previous solutions assume local score $s_{i,j} \geq 0$ and want the largest $s_i = \sum_{j=1}^{m} s_{i,j}$.

We have $w_{i,j} < 0$ and $w_{i,j} \geq 0$ and want the largest $|w_i|$.
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\[
\begin{array}{c|c|c}
\text{id} & x & s_j(x) \\
\hline
0 & 5 & 20 \\
1 & 2 & 7 \\
2 & 1 & 6 \\
3 & 4 & -2 \\
4 & 6 & -15 \\
5 & 3 & -30 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c}
\text{id} & x & s_j(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
0 & 5 & 12 & & & & \\
1 & 4 & 7 & & & & \\
2 & 1 & 2 & & & & \\
3 & 2 & -5 & & & & \\
4 & 3 & -14 & & & & \\
5 & 6 & -20 & & & & \\
\end{array}
\]

\(k = 1\)
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>$R$ id</th>
<th>$x$</th>
<th>$s_i(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$ id</td>
<td>$x$</td>
<td>$s_i(x)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 1</th>
<th>id</th>
<th>$x$</th>
<th>$s_1(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>$e_{1,2}$</td>
<td>2</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>$e_{1,3}$</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>$e_{1,4}$</td>
<td>4</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
<td></td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 2</th>
<th>id</th>
<th>$x$</th>
<th>$s_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>$e_{2,2}$</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>$e_{2,3}$</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$e_{2,4}$</td>
<td>2</td>
<td>-5</td>
<td></td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
<td></td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 3</th>
<th>id</th>
<th>$x$</th>
<th>$s_3(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>$e_{3,2}$</td>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>$e_{3,3}$</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>$e_{3,4}$</td>
<td>2</td>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>$e_{3,5}$</td>
<td>5</td>
<td>-6</td>
<td></td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
<td></td>
</tr>
</tbody>
</table>

• An item $x$ has a local score $s_i(x)$ at node $i \forall i \in [1 \ldots m]$, where if $x$ does not appear $s_i(x) = 0$
Exact Top-$k$ Wavelet Coefficients: Our Solution

Each node sends:
- the top-$k$ most positive scored items
- the top-$k$ most negative scored items.

$k = 1$

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_j(x) \\
\hline
\text{e}_{1,1} & 5 & 20 \\
\text{e}_{1,6} & 3 & -30 \\
\text{e}_{2,1} & 5 & 12 \\
\text{e}_{2,6} & 6 & -20 \\
\text{e}_{3,1} & 1 & 10 \\
\text{e}_{3,6} & 6 & -10 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
\text{node 1} & & & & & \\
\text{node 2} & & & & & \\
\text{node 3} & & & & & \\
\hline
\end{array}
\]

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
The coordinator computes useful bounds for each received item.

---

<table>
<thead>
<tr>
<th>id</th>
<th>x</th>
<th>$s_j(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>$e_{1,2}$</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id</th>
<th>x</th>
<th>$s_j(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$e_{3,2}$</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>$e_{3,3}$</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$e_{3,4}$</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>$e_{3,5}$</td>
<td>5</td>
<td>-6</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\[
R
\begin{array}{c|c|c}
\text{id} & x & s_j(x) \\
\hline
e_{1,1} & 5 & 20 \\
e_{1,2} & 3 & -30 \\
e_{2,1} & 5 & 12 \\
e_{2,2} & 6 & -20 \\
e_{3,1} & 1 & 10 \\
e_{3,2} & 6 & -10 \\
\end{array}
\]

\[
R
\begin{array}{c|c|c|c|c|c}
x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
1 & 10 & 001 & 42 & 40 & 0 \\
3 & -30 & 100 & -8 & -60 & 8 \\
5 & 32 & 110 & 42 & 22 & 22 \\
6 & -30 & 011 & -10 & -60 & 10 \\
\end{array}
\]

\(k = 1\)

- \(s(x)\) denotes the partial score sum for \(x\)

\[
R
\begin{array}{c|c|c|c|c|c|c|c|c|c}
\text{node 1} & \text{id} & x & s_1(x) \\
\hline
e_{1,1} & 5 & 20 \\
e_{1,2} & 2 & 7 \\
e_{1,3} & 1 & 6 \\
e_{1,4} & 4 & -2 \\
e_{1,5} & 6 & -15 \\
e_{1,6} & 3 & -30 \\
\end{array}
\]

\[
R
\begin{array}{c|c|c|c|c|c|c|c|c|c}
\text{node 2} & \text{id} & x & s_2(x) \\
\hline
e_{2,1} & 5 & 12 \\
e_{2,2} & 4 & 7 \\
e_{2,3} & 1 & 2 \\
e_{2,4} & 2 & -5 \\
e_{2,5} & 3 & -14 \\
e_{2,6} & 6 & -20 \\
\end{array}
\]

\[
R
\begin{array}{c|c|c|c|c|c|c|c|c|c}
\text{node 3} & \text{id} & x & s_3(x) \\
\hline
e_{3,1} & 1 & 10 \\
e_{3,2} & 3 & 6 \\
e_{3,3} & 4 & 5 \\
e_{3,4} & 2 & -3 \\
e_{3,5} & 5 & -6 \\
e_{3,6} & 6 & -10 \\
\end{array}
\]
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\[
R
\begin{array}{|c|c|c|}
\hline
id & x & s_j(x) \\
\hline
1 & 5 & 20 \\
2 & 3 & -30 \\
3 & 5 & 12 \\
4 & 6 & -20 \\
5 & 1 & 10 \\
6 & 6 & -10 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
1 & 10 & 001 & 42 & -40 & 0 \\
3 & -30 & 100 & -8 & -60 & 8 \\
5 & 32 & 110 & 42 & 22 & 22 \\
6 & -30 & 011 & -10 & -60 & 10 \\
\hline
\end{array}
\]

\(\cdot \hat{s}(x)\) denotes the partial score sum for \(x\)

\[
\begin{array}{|c|c|c|}
\hline
node 1 & id & x & s_1(x) \\
\hline
1 & e_{1,1} & 5 & 20 \\
2 & e_{1,2} & 2 & 7 \\
3 & e_{1,3} & 1 & 6 \\
4 & e_{1,4} & 4 & -2 \\
5 & e_{1,5} & 6 & -15 \\
6 & e_{1,6} & 3 & -30 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
node 2 & id & x & s_2(x) \\
\hline
1 & e_{2,1} & 5 & 12 \\
2 & e_{2,2} & 4 & 7 \\
3 & e_{2,3} & 1 & 2 \\
4 & e_{2,4} & 2 & -5 \\
5 & e_{2,5} & 3 & -14 \\
6 & e_{2,6} & 6 & -20 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
node 3 & id & x & s_3(x) \\
\hline
1 & e_{3,1} & 1 & 10 \\
2 & e_{3,2} & 3 & 6 \\
3 & e_{3,3} & 4 & 5 \\
4 & e_{3,4} & 2 & -3 \\
5 & e_{3,5} & 5 & -6 \\
6 & e_{3,6} & 6 & -10 \\
\hline
\end{array}
\]
Exact Top-\(k\) Wavelet Coefficients: Our Solution

<table>
<thead>
<tr>
<th>(R)</th>
<th>(x)</th>
<th>(s(x))</th>
<th>(F_x)</th>
<th>(\tau^+(x))</th>
<th>(\tau^-(x))</th>
<th>(\tau(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e_{1,1})</td>
<td>5</td>
<td>20</td>
<td>1</td>
<td>10</td>
<td>011</td>
<td>001</td>
</tr>
<tr>
<td>(e_{1,6})</td>
<td>3</td>
<td>-30</td>
<td>3</td>
<td>-30</td>
<td>-10</td>
<td>-60</td>
</tr>
<tr>
<td>(e_{2,1})</td>
<td>5</td>
<td>12</td>
<td>5</td>
<td>32</td>
<td>110</td>
<td>42</td>
</tr>
<tr>
<td>(e_{2,6})</td>
<td>6</td>
<td>-20</td>
<td>6</td>
<td>-30</td>
<td>011</td>
<td>-10</td>
</tr>
<tr>
<td>(e_{3,1})</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>(e_{3,6})</td>
<td>6</td>
<td>-10</td>
<td>6</td>
<td>-30</td>
<td>011</td>
<td>-10</td>
</tr>
</tbody>
</table>

- \(F_x\) is a receipt indication bit vector, if \(s_i(x)\) is received \(F_x(i) = 1\), else \(F_x(i) = 0\).

\(k = 1\)

<table>
<thead>
<tr>
<th>(R)</th>
<th>(x)</th>
<th>(s(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e_{1,1})</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>(e_{1,6})</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>(e_{2,1})</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>(e_{2,6})</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>(e_{3,1})</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>(e_{3,6})</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

Node 1

<table>
<thead>
<tr>
<th>id</th>
<th>(x)</th>
<th>(s_1(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e_{1,1})</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>(e_{1,2})</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>(e_{1,3})</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>(e_{1,4})</td>
<td>4</td>
<td>-2</td>
</tr>
<tr>
<td>(e_{1,5})</td>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>(e_{1,6})</td>
<td>3</td>
<td>-30</td>
</tr>
</tbody>
</table>

Node 2

<table>
<thead>
<tr>
<th>id</th>
<th>(x)</th>
<th>(s_2(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e_{2,1})</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>(e_{2,2})</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>(e_{2,3})</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>(e_{2,4})</td>
<td>2</td>
<td>-5</td>
</tr>
<tr>
<td>(e_{2,5})</td>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>(e_{2,6})</td>
<td>6</td>
<td>-20</td>
</tr>
</tbody>
</table>

Node 3

<table>
<thead>
<tr>
<th>id</th>
<th>(x)</th>
<th>(s_3(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e_{3,1})</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>(e_{3,2})</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>(e_{3,3})</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(e_{3,4})</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>(e_{3,5})</td>
<td>5</td>
<td>-6</td>
</tr>
<tr>
<td>(e_{3,6})</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>
Exact Top-$k$ Wavelet Coefficients: Our Solution

- $F_x$ is a receipt indication bit vector, if $s_i(x)$ is received $F_x(i) = 1$, else $F_x(i) = 0$. 

<table>
<thead>
<tr>
<th>$R$</th>
<th></th>
<th>$R'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>$x$</td>
<td>$s_j(x)$</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

- $k = 1$

![Table](table.png)
Exact Top-$k$ Wavelet Coefficients: Our Solution

$\tau^+(x)$ is an upper bound on the total score $s(x)$, if $s_i(x)$ received then $\tau^+(x) = \tau^+(x) + s_i(x)$, else $\tau^+(x) = \tau^+(x) + k^{th}$ most positive from node $i$.
Exact Top-$k$ Wavelet Coefficients: Our Solution

- $\tau^+(x)$ is an upper bound on the total score $s(x)$, if $s_i(x)$ received then $\tau^+(x) = \tau^+(x) + s_i(x)$
- else $\tau^+(x) = \tau^+(x) + k$'th most positive from node $i$
Exact Top-$k$ Wavelet Coefficients: Our Solution

$\tau^-(x)$ is a lower bound on the total score sum $s(x)$, 
if $s_i(x)$ received then $\tau^-(x) = \tau^-(x) + s_i(x)$
else $\tau^-(x) = \tau^-(x) + k$'th most negative score from node $i$
• $\tau^-(x)$ is a lower bound on the total score sum $s(x)$, 
if $s_i(x)$ received then $\tau^-(x) = \tau^-(x) + s_i(x)$
else $\tau^-(x) = \tau^-(x) + k$’th most negative score from node $i$
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_1(x) \\
\hline
e_{1,1} & 5 & 20 \\
\hline
e_{1,6} & 3 & -30 \\
\hline
\end{array}
\quad
\begin{array}{|c|c|c|c|c|c|}
\hline
x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
1 & 10 & 001 & 42 & -40 & 0 \\
3 & -30 & 100 & -8 & -60 & 8 \\
5 & 32 & 110 & 42 & 22 & 22 \\
6 & -30 & 011 & -10 & -60 & 10 \\
\hline
\end{array}
\]

\bullet \ \tau(x) \text{ is a lower bound on } |s(x)| \text{ computed as,}

\[
\tau(x) = 0 \text{ if } \text{sign}(\tau^+(x)) \neq \text{sign}(\tau^-(x)) \\
\tau(x) = \min(|\tau^+(x)|,|\tau^-(x)|) \text{ otherwise.}
\]

\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_2(x) \\
\hline
\hline
\end{array}

\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_3(x) \\
\hline
\hline
\end{array}
Exact Top-$k$ Wavelet Coefficients: Our Solution

\[ R \]

<table>
<thead>
<tr>
<th>id</th>
<th>x</th>
<th>$s_j(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

\[ R \]

<table>
<thead>
<tr>
<th>x</th>
<th>$s(x)$</th>
<th>$F_x$</th>
<th>$\tau^+(x)$</th>
<th>$\tau^-(x)$</th>
<th>$\tau(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1001</td>
<td>42</td>
<td>-40</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-30</td>
<td>100</td>
<td>-8</td>
<td>-60</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>42</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>-30</td>
<td>011</td>
<td>-10</td>
<td>-60</td>
<td>10</td>
</tr>
</tbody>
</table>

- $\tau(x)$ is a lower bound on $|s(x)|$ computed as,
  - $\tau(x) = 0$ if $\text{sign}(\tau^+(x)) \neq \text{sign}(\tau^-(x))$
  - $\tau(x) = \min(|\tau^+(x)|, |\tau^-(x)|)$ otherwise.
Exact Top-$k$ Wavelet Coefficients: Our Solution

We select the item with the $k$th largest $\tau(x)$. $\tau(x)$ is a lower bound $T_1$ on the top-$k |s(x)|$ for unseen item $x$.

<table>
<thead>
<tr>
<th>$R$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$id$</td>
<td>$x$</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
</tr>
</tbody>
</table>

$T_1 = 22$, $T_1/m = 22/3$

<table>
<thead>
<tr>
<th>node 1</th>
<th>node 2</th>
<th>node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$id$</td>
<td>$x$</td>
<td>$s_1(x)$</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>$e_{1,2}$</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>$e_{1,3}$</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>$e_{1,4}$</td>
<td>4</td>
<td>-2</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
</tr>
</tbody>
</table>
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

### Node 1

<table>
<thead>
<tr>
<th>id</th>
<th>$x$</th>
<th>$s_1(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>$e_{1,2}$</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>$e_{1,3}$</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>$e_{1,4}$</td>
<td>4</td>
<td>-2</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
</tr>
</tbody>
</table>

### Node 2

<table>
<thead>
<tr>
<th>id</th>
<th>$x$</th>
<th>$s_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>$e_{2,2}$</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>$e_{2,3}$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$e_{2,4}$</td>
<td>2</td>
<td>-5</td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
</tr>
</tbody>
</table>

### Node 3

<table>
<thead>
<tr>
<th>id</th>
<th>$x$</th>
<th>$s_3(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$e_{3,2}$</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>$e_{3,3}$</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$e_{3,4}$</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>$e_{3,5}$</td>
<td>5</td>
<td>-6</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

$R$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$x$</th>
<th>$s(x)$</th>
<th>$F_x$</th>
<th>$\tau^+(x)$</th>
<th>$\tau^-(x)$</th>
<th>$\tau(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>42</td>
<td>-40</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-30</td>
<td>100</td>
<td>-8</td>
<td>-60</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>42</td>
<td>22</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-30</td>
<td>011</td>
<td>-10</td>
<td>-60</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

$T_1 = 22$, $T_1/m = 22/3$

- Any unseen item $x$ must have at least:
  - one $s_i(x) > T_1/m$ or
  - one $s_i(x) < -T_1/m$

To get into the top-$k$. 

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Our Solution

\[ k = 1 \]

\[
\begin{array}{|c|c|c|}
\hline
id & x & s_j(x) \\
\hline
\hline
 e_{1,1} & 5 & 20 \\
 e_{1,2} & 3 & -30 \\
 e_{2,1} & 5 & 12 \\
 e_{2,6} & 6 & -20 \\
 e_{3,1} & 1 & 10 \\
 e_{3,6} & 6 & -10 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
1 & 10 & 001 & 42 & -40 & 0 \\
3 & -30 & 100 & -8 & -60 & 8 \\
5 & 32 & 110 & 42 & 22 & 22 \\
6 & -30 & 011 & -10 & -60 & 10 \\
\hline
\end{array}
\]

\[ T_1 = 22, \ T_1/m = 22/3 \]

Round 1 End

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
id & x & s_j(x) \\
\hline
\hline
 e_{1,1} & 5 & 20 \\
 e_{1,2} & 7 & 3 \\
 e_{1,3} & 6 & 5 \\
 e_{1,4} & 4 & 2 \\
 e_{1,5} & 6 & 10 \\
 e_{1,6} & 3 & -20 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
id & x & s_j(x) \\
\hline
\hline
 e_{2,1} & 5 & 12 \\
 e_{2,2} & 4 & 7 \\
 e_{2,3} & 1 & 2 \\
 e_{2,4} & 2 & -5 \\
 e_{2,5} & 3 & -14 \\
 e_{2,6} & 6 & -20 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
id & x & s_j(x) \\
\hline
\hline
 e_{3,1} & 1 & 10 \\
 e_{3,2} & 3 & 6 \\
 e_{3,3} & 4 & 5 \\
 e_{3,4} & 2 & -3 \\
 e_{3,5} & 5 & -6 \\
 e_{3,6} & 6 & -10 \\
\hline
\end{array}
\]

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\(k = 1\)

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_j(x) \\
\hline
\text{e}_{1,1} & 5 & 20 \\
\text{e}_{1,6} & 3 & -30 \\
\text{e}_{2,1} & 5 & 12 \\
\text{e}_{2,6} & 6 & -20 \\
\text{e}_{3,1} & 1 & 10 \\
\text{e}_{3,6} & 6 & -10 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
1 & 10 & 001 & 42 & -40 & 0 \\
3 & -30 & 100 & -8 & -60 & 8 \\
5 & 32 & 110 & 42 & 22 & 22 \\
6 & -30 & 011 & -10 & -60 & 10 \\
\hline
\end{array}
\]

\(T_1 = 22, \ T_1/m = 22/3\)

- \(T_1/m\) sent to each site.

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_1(x) \\
\hline
\text{e}_{1,1} & 5 & 20 \\
\text{e}_{1,2} & 2 & 7 \\
\text{e}_{1,3} & 1 & 6 \\
\text{e}_{1,4} & 4 & -2 \\
\text{e}_{1,5} & 6 & -15 \\
\text{e}_{1,6} & 3 & -30 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_2(x) \\
\hline
\text{e}_{2,1} & 5 & 12 \\
\text{e}_{2,2} & 4 & 7 \\
\text{e}_{2,3} & 1 & 2 \\
\text{e}_{2,4} & 2 & -5 \\
\text{e}_{2,5} & 3 & -14 \\
\text{e}_{2,6} & 6 & -20 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_3(x) \\
\hline
\text{e}_{3,1} & 1 & 10 \\
\text{e}_{3,2} & 3 & 6 \\
\text{e}_{3,3} & 4 & 5 \\
\text{e}_{3,4} & 2 & -3 \\
\text{e}_{3,5} & 5 & -6 \\
\text{e}_{3,6} & 6 & -10 \\
\hline
\end{array}
\]
Each site finds items with $s_i(x) > T_1/m$ or $s_i(x) < T_1/m$. 

$T_1 = 22$, $T_1/m = 22/3$
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\[ R \]

\[ \begin{array}{|c|c|c|} \hline \text{id} & x & s_j(x) \\ \hline e_{1,1} & 5 & 20 \\ e_{1,6} & 3 & -30 \\ e_{2,1} & 5 & 12 \\ e_{2,6} & 6 & -20 \\ e_{3,1} & 1 & 10 \\ e_{3,6} & 6 & -10 \\ \hline \end{array} \]

\[ R \]

\[ \begin{array}{|c|c|c|c|c|c|} \hline x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\ \hline 1 & 10 & 001 & 42 & -40 & 0 \\ 3 & -30 & 100 & -8 & -60 & 8 \\ 5 & 32 & 110 & 42 & 22 & 22 \\ 6 & -30 & 011 & -10 & -60 & 10 \\ \hline \end{array} \]

\(T_1 = 22, \ T_1/m = 22/3\)

\(k = 1\)

- Items with \(|s_i(x)| > T_1/m\) are sent to coordinator.
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\(k = 1\)

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & \text{x} & s_j(x) \\
\hline
\text{e}_{1,1} & 5 & 20 \\
\text{e}_{1,5} & 6 & -15 \\
\text{e}_{1,6} & 3 & -30 \\
\text{e}_{2,1} & 5 & 12 \\
\text{e}_{2,5} & 3 & -14 \\
\text{e}_{2,6} & 6 & -20 \\
\text{e}_{3,1} & 1 & 10 \\
\text{e}_{3,6} & 6 & -10 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{x} & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
1 & 10 & 001 & 42 & -40 & 0 \\
3 & -30 & 100 & -8 & -60 & 8 \\
5 & 32 & 110 & 42 & 22 & 22 \\
6 & -30 & 011 & -10 & -60 & 10 \\
\hline
\end{array}
\]

\(T_1 = 22, \ T_1/m = 22/3\)

- Items with \(|s_i(x)| > T_1/m\) are sent to coordinator.

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & \text{x} & s_1(x) \\
\hline
\text{e}_{1,1} & 5 & 20 \\
\text{e}_{1,2} & 2 & 7 \\
\text{e}_{1,3} & 1 & 6 \\
\text{e}_{1,4} & 4 & -2 \\
\text{e}_{1,5} & 6 & -15 \\
\text{e}_{1,6} & 3 & -30 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & \text{x} & s_2(x) \\
\hline
\text{e}_{2,1} & 5 & 12 \\
\text{e}_{2,2} & 4 & 7 \\
\text{e}_{2,3} & 1 & 2 \\
\text{e}_{2,4} & 2 & -5 \\
\text{e}_{2,5} & 3 & -14 \\
\text{e}_{2,6} & 6 & -20 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & \text{x} & s_3(x) \\
\hline
\text{e}_{3,1} & 1 & 10 \\
\text{e}_{3,2} & 3 & 6 \\
\text{e}_{3,3} & 4 & 5 \\
\text{e}_{3,4} & 2 & -3 \\
\text{e}_{3,5} & 5 & -6 \\
\text{e}_{3,6} & 6 & -10 \\
\hline
\end{array}
\]

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Our Solution

### $k = 1$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$x$</th>
<th>$s_j(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$R$</th>
<th>$x$</th>
<th>$s_j(x)$</th>
<th>$F_x$</th>
<th>$\tau^+(x)$</th>
<th>$\tau^-(x)$</th>
<th>$\tau(x)$</th>
<th>$\tau'(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>24.6</td>
<td>-4.6</td>
<td>0</td>
<td>24.6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-44</td>
<td>110</td>
<td>-36.6</td>
<td>-51.3</td>
<td>36.6</td>
<td>51.3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>39.3</td>
<td>24.6</td>
<td>24.6</td>
<td>39.3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-45</td>
<td>111</td>
<td>-45</td>
<td>-45</td>
<td>45</td>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

$T_1 = 22, \frac{T_1}{m} = 22/3$

- The coordinator updates the bounds for each item it has ever received.
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>$R$</th>
<th></th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>$x$</td>
<td>$s_j(x)$</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

- The coordinator updates the bounds for each item it has ever received.
- Partial score sum $s(5) = 20 + 12$
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>$R$</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>$x$</td>
<td>$s_j(x)$</td>
<td></td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
<td></td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
<td></td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
<td></td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
<td></td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$R$</th>
<th>$x$</th>
<th>$s(x)$</th>
<th>$F_x$</th>
<th>$\tau^+(x)$</th>
<th>$\tau^-(x)$</th>
<th>$\tau(x)$</th>
<th>$\tau'(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>24.6</td>
<td>-4.6</td>
<td>0</td>
<td>24.6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-44</td>
<td>110</td>
<td>-36.6</td>
<td>-51.3</td>
<td>36.6</td>
<td>51.3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>39.3</td>
<td>24.6</td>
<td>24.6</td>
<td>39.3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-45</td>
<td>111</td>
<td>-45</td>
<td>-45</td>
<td>45</td>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

$T_1 = 22$, $T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.
- Receipt vector $F_5 = [110]$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>R</th>
<th>id</th>
<th>$s_j(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$e_{1,1}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$e_{1,5}$</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>$e_{1,6}$</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>$e_{2,1}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$e_{2,5}$</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>$e_{2,6}$</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>$e_{3,1}$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$e_{3,6}$</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>$x$</th>
<th>$s(x)$</th>
<th>$F_x$</th>
<th>$\tau^+(x)$</th>
<th>$\tau^-(x)$</th>
<th>$\tau(x)$</th>
<th>$\tau'(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>24.6</td>
<td>-4.6</td>
<td>0</td>
<td>24.6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-44</td>
<td>110</td>
<td>-36.6</td>
<td>-51.3</td>
<td>36.6</td>
<td>51.3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>39.3</td>
<td>24.6</td>
<td>24.6</td>
<td>39.3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-45</td>
<td>111</td>
<td>-45</td>
<td>-45</td>
<td>45</td>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

$T_1 = 22$, $T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- $\tau^+(x)$ is now tighter, if $s_i(x)$ received then $\tau^+(x) = \tau^+(x) + s_i(x)$

  else $\tau^+(x) = \tau^+(x) + T_1/m$

<table>
<thead>
<tr>
<th>node 1</th>
<th>id</th>
<th>$s_1(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\checkmark$</td>
<td>$e_{1,1}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$e_{1,2}$</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>$e_{1,3}$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$e_{1,4}$</td>
<td>4</td>
</tr>
<tr>
<td>$\checkmark$</td>
<td>$e_{1,5}$</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>$e_{1,6}$</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 2</th>
<th>id</th>
<th>$s_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\checkmark$</td>
<td>$e_{2,1}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$e_{2,2}$</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>$e_{2,3}$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$e_{2,4}$</td>
<td>2</td>
</tr>
<tr>
<td>$\checkmark$</td>
<td>$e_{2,5}$</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>$e_{2,6}$</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 3</th>
<th>id</th>
<th>$s_3(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\checkmark$</td>
<td>$e_{3,1}$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$e_{3,2}$</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>$e_{3,3}$</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>$e_{3,4}$</td>
<td>2</td>
</tr>
<tr>
<td>$\checkmark$</td>
<td>$e_{3,5}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>$e_{3,6}$</td>
<td>6</td>
</tr>
</tbody>
</table>
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
</tr>
<tr>
<td>$e_{2,5}$</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

$T_1 = 22, T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- $\tau^-(x)$ is also tighter, if $s_i(x)$ received then $\tau^-(x) = \tau^-(x) + s_i(x)$

- else $\tau^-(x) = \tau^-(x) - T_1/m$

<table>
<thead>
<tr>
<th>node 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
</tr>
<tr>
<td>$e_{1,2}$</td>
</tr>
<tr>
<td>$e_{1,3}$</td>
</tr>
<tr>
<td>$e_{1,4}$</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
</tr>
<tr>
<td>$e_{2,2}$</td>
</tr>
<tr>
<td>$e_{2,3}$</td>
</tr>
<tr>
<td>$e_{2,4}$</td>
</tr>
<tr>
<td>$e_{2,5}$</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
</tr>
<tr>
<td>$e_{3,2}$</td>
</tr>
<tr>
<td>$e_{3,3}$</td>
</tr>
<tr>
<td>$e_{3,4}$</td>
</tr>
<tr>
<td>$e_{3,5}$</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
</tr>
</tbody>
</table>
Exact Top-$k$ Wavelet Coefficients: Our Solution

$\bullet$ The coordinator updates the bounds for each item it has ever received.

$\bullet$ Score absolute value bound $\tau(5) = \min(39.3, 24.6)$.

$\begin{array}{|c|c|c|}
\hline
\text{id} & \text{x} & \text{s}_j(x) \\
\hline
\text{e}_{1,1} & 5 & 20 \\
\text{e}_{1,5} & 6 & -15 \\
\text{e}_{1,6} & 3 & -30 \\
\text{e}_{2,1} & 5 & 12 \\
\text{e}_{2,5} & 3 & -14 \\
\text{e}_{2,6} & 6 & -20 \\
\text{e}_{3,1} & 1 & 10 \\
\text{e}_{3,6} & 6 & -10 \\
\hline
\end{array}$

$\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{x} & \text{s}(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) & \tau'(x) \\
\hline
1 & 10 & 001 & 24.6 & -4.6 & 0 & 24.6 \\
3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\
6 & -45 & 111 & -45 & -45 & 45 & 45 \\
\hline
\end{array}$

$T_1 = 22$, $T_1/m = 22/3$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>$x$</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
</tr>
<tr>
<td>$e_{1,2}$</td>
<td>2</td>
</tr>
<tr>
<td>$e_{1,3}$</td>
<td>1</td>
</tr>
<tr>
<td>$e_{1,4}$</td>
<td>4</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
</tr>
</tbody>
</table>

$T_1 = 22$, $T_1/m = 22/3$

- The coordinator updates the bounds for each item it has ever received.

- $\tau'(x)$ is an upper bound on $|s(x)|$,

$$\tau'(x) = \max\{|\tau^+(x)|, |\tau^-(x)|\}$$
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\[
\begin{array}{c|c|c|c|c|c|c|c|c}
    & \mathbf{x} & s_j(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) & \tau'(x) \\
\hline
    \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} \\
    e_{1,1} & 5 & 20 & 10 & 24.6 & -4.6 & 0 & 24.6 \\
    e_{1,5} & 6 & -15 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
    e_{1,6} & 3 & -30 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\
    e_{2,1} & 5 & 12 & 111 & -45 & -45 & 45 & 45 \\
    e_{2,5} & 3 & -14 & & & & & \\
    e_{2,6} & 6 & -20 & & & & & \\
    e_{3,1} & 1 & 10 & & & & & \\
    e_{3,6} & 6 & -10 & & & & & \\
\end{array}
\]

\[T_1 = 22, \quad T_1/m = 22/3\]

- The coordinator updates the bounds for each item it has ever received.

- \(\tau'(x)\) is an upper bound on \(|s(x)|\),

\[\tau'(x) = \max\{|\tau^+(x)|, |\tau^-(x)|\}\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
    & \mathbf{x} & s_1(x) & & & & & & & & & \\
\hline
    \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} \\
    e_{1,1} & 5 & 20 & & & & & & & & & \\
    e_{1,2} & 2 & 7 & & & & & & & & & \\
    e_{1,3} & 1 & 6 & & & & & & & & & \\
    e_{1,4} & 4 & -2 & & & & & & & & & \\
    e_{1,5} & 6 & -15 & & & & & & & & & \\
    e_{1,6} & 3 & -30 & & & & & & & & & \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
    & \mathbf{x} & s_2(x) & & & & & & & & & \\
\hline
    \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} \\
    e_{2,1} & 5 & 12 & & & & & & & & & \\
    e_{2,2} & 4 & 7 & & & & & & & & & \\
    e_{2,3} & 1 & 2 & & & & & & & & & \\
    e_{2,4} & 2 & -5 & & & & & & & & & \\
    e_{2,5} & 3 & -14 & & & & & & & & & \\
    e_{2,6} & 6 & -20 & & & & & & & & & \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
    & \mathbf{x} & s_3(x) & & & & & & & & & \\
\hline
    \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} & \text{id} \\
    e_{3,1} & 1 & 10 & & & & & & & & & \\
    e_{3,2} & 3 & 6 & & & & & & & & & \\
    e_{3,3} & 4 & 5 & & & & & & & & & \\
    e_{3,4} & 2 & -3 & & & & & & & & & \\
    e_{3,5} & 5 & -6 & & & & & & & & & \\
    e_{3,6} & 6 & -10 & & & & & & & & & \\
\end{array}
\]

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Our Solution

• We select the item $x$ with the $k$th largest $\tau(x)$, which serves as a new lower bound $T_2$ on $|s(x)|$ for any item.

<table>
<thead>
<tr>
<th>$R$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>$x$</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
</tr>
</tbody>
</table>

$T_1 = 22$, $T_1/m = 22/3$

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
**Exact Top-k Wavelet Coefficients: Our Solution**

We select the item \( x \) with the \( k \)th largest \( \tau(x) \), which serves as a new lower bound \( T_2 \) on \( |s(x)| \) for any item.

\[
R
\begin{array}{ccc}
\text{id} & \text{x} & \text{s}_j(x) \\
e_{1,1} & 5 & 20 \\
e_{1,5} & 6 & -15 \\
e_{1,6} & 3 & -30 \\
e_{2,1} & 5 & 12 \\
e_{2,5} & 3 & -14 \\
e_{2,6} & 6 & -20 \\
e_{3,1} & 1 & 10 \\
e_{3,6} & 6 & -10 \\
\end{array}
\]

\[
R
\begin{array}{ccccccc}
\text{x} & \text{s}(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) & \tau'(x) \\
1 & 10 & 001 & 24.6 & -4.6 & 0 & 24.6 \\
3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\
6 & -45 & 111 & -45 & -45 & 45 & 45 \\
\end{array}
\]

\( T_1 = 22, \ T_1/m = 22/3 \)

\( T_2 = 45 \)

\[
\sqrt{k = 1}
\]
Exact Top-$k$ Wavelet Coefficients: Our Solution

$R$

<table>
<thead>
<tr>
<th>id</th>
<th>$x$</th>
<th>$s_j(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

$T_1 = 22, \ T_1/m = 22/3$

$T_2 = 45$

- Any item with $\tau'(x) < T_2$ cannot be in the top-$k$ and is pruned from $R$.
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>$x$</td>
</tr>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
</tr>
</tbody>
</table>

$T_1 = 22$, $T_1/m = 22/3$

$T_2 = 45$

- Any remaining items with a 0 in vector $F_x$ are selected.
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\(k = 1\)

\[
R
\begin{array}{|c|c|c|}
\hline
id & x & s_j(x) \\
\hline
 e_{1,1} & 5 & 20 \\
 e_{1,5} & 6 & -15 \\
 e_{1,6} & 3 & -30 \\
 e_{2,1} & 5 & 12 \\
 e_{2,5} & 3 & -14 \\
 e_{2,6} & 6 & -20 \\
 e_{3,1} & 1 & 10 \\
 e_{3,6} & 6 & -10 \\
\hline
\end{array}
\]

\[
R
\begin{array}{|c|c|c|c|c|c|}
\hline
x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) \\
\hline
1 & 10 & 001 & 24.6 & -4.6 & 0 & 24.6 \\
3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\
6 & -45 & 111 & -45 & -45 & 45 & 45 \\
\hline
\end{array}
\]

\(T_1 = 22, T_1/m = 22/3\)

\(T_2 = 45\)

Round 2 End

\[
\begin{array}{|c|c|c|}
\hline
\text{node 1} & \text{id} & x & s_1(x) \\
\hline
 e_{1,1} & 5 & 20 \\
 e_{1,2} & 2 & 7 \\
 e_{1,3} & 1 & 6 \\
 e_{1,4} & 4 & -2 \\
 e_{1,5} & 6 & -15 \\
 e_{1,6} & 3 & -30 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{node 2} & \text{id} & x & s_2(x) \\
\hline
 e_{2,1} & 5 & 12 \\
 e_{2,2} & 4 & 7 \\
 e_{2,3} & 1 & 2 \\
 e_{2,4} & 2 & -5 \\
 e_{2,5} & 3 & -14 \\
 e_{2,6} & 6 & -20 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{node 3} & \text{id} & x & s_3(x) \\
\hline
 e_{3,1} & 1 & 10 \\
 e_{3,2} & 3 & 6 \\
 e_{3,3} & 4 & 5 \\
 e_{3,4} & 2 & -3 \\
 e_{3,5} & 5 & -6 \\
 e_{3,6} & 6 & -10 \\
\hline
\end{array}
\]
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$s(x)$</th>
<th>$F_x$</th>
<th>$\tau^+(x)$</th>
<th>$\tau^-(x)$</th>
<th>$\tau(x)$</th>
<th>$\tau'(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
<td>24.6</td>
<td>-4.6</td>
<td>0</td>
<td>24.6</td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
<td>-36.6</td>
<td>-51.3</td>
<td>36.6</td>
<td>51.3</td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
<td>-24.6</td>
<td>24.6</td>
<td>-24.6</td>
<td>39.3</td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
<td>5</td>
<td>-14</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
<td>6</td>
<td>6</td>
<td>-10</td>
<td>6</td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
<td>-45</td>
<td>-14</td>
<td>-45</td>
<td>45</td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
<td>-10</td>
<td>10</td>
<td>-10</td>
<td>10</td>
</tr>
</tbody>
</table>

$T_1 = 22, \quad T_1/m = 22/3$

$T_2 = 45$

$s_3(3) = ?$

- The coordinator asks for missing scores for items still in $R$.
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\[ R \]

<table>
<thead>
<tr>
<th>id</th>
<th>(x)</th>
<th>(s_j(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e_{1,1})</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>(e_{1,5})</td>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>(e_{1,6})</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>(e_{2,1})</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>(e_{2,5})</td>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>(e_{2,6})</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>(e_{3,1})</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>(e_{3,6})</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

\[ \mathbf{\hat{s}}(3) =? \]

\[ T_1 = 22, \ T_1/m = 22/3 \]

\[ T_2 = 45 \]

**Node 1**

\[ \begin{array}{c|c|c|c|c|c|c|c} \hline x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) & \tau'(x) \\ \hline 1 & 10 & 001 & 24.6 & -4.6 & 0 & 24.6 \\ 3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\ 5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\ \hline \end{array} \]

\[ \sqrt{s_1(3) = 24.6} \]

**Node 2**

\[ \begin{array}{c|c|c|c|c|c|c|c} \hline x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) & \tau'(x) \\ \hline 3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\ 5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\ \hline \end{array} \]

\[ \sqrt{s_2(3) = 24.6} \]

**Node 3**

\[ \begin{array}{c|c|c|c|c|c|c|c} \hline x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) & \tau'(x) \\ \hline 3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\ 5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\ \hline \end{array} \]

\[ \sqrt{s_3(3) = 24.6} \]
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\[
\begin{array}{|c|c|c|}
\hline
\text{id} & x & s_j(x) \\
\hline
\text{e}_{1,1} & 5 & 20 \\
\text{e}_{1,5} & 6 & -15 \\
\text{e}_{1,6} & 3 & -30 \\
\text{e}_{2,1} & 5 & 12 \\
\text{e}_{2,5} & 3 & -14 \\
\text{e}_{2,6} & 6 & -20 \\
\text{e}_{3,1} & 1 & 10 \\
\text{e}_{3,2} & 3 & 6 \\
\text{e}_{3,6} & 6 & -10 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
x & s(x) & F_x & \tau^+(x) & \tau^-(x) & \tau(x) & \tau'(x) \\
\hline
1 & 10 & 001 & 24.6 & -4.6 & 0 & 24.6 \\
3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
5 & 32 & 110 & -39.3 & -24.6 & 24.6 & 39.3 \\
6 & -45 & 111 & -45 & -45 & 45 & 45 \\
\hline
\end{array}
\]

\[
\begin{align*}
T_1 &= 22, \quad T_1/m = 22/3 \\
T_2 &= 45 \\
\end{align*}
\]

\[
\begin{align*}
s_3(3) &= ? \\
\end{align*}
\]
**Exact Top-\(k\) Wavelet Coefficients: Our Solution**

### Node 1

<table>
<thead>
<tr>
<th>id</th>
<th>(x)</th>
<th>(s_1(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e_{1,1})</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>(e_{1,5})</td>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>(e_{1,6})</td>
<td>3</td>
<td>-30</td>
</tr>
<tr>
<td>(e_{2,1})</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>(e_{2,5})</td>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>(e_{2,6})</td>
<td>6</td>
<td>-20</td>
</tr>
<tr>
<td>(e_{3,1})</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>(e_{3,2})</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>(e_{3,6})</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

### Node 2

<table>
<thead>
<tr>
<th>id</th>
<th>(x)</th>
<th>(s_2(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e_{2,1})</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>(e_{2,2})</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>(e_{2,3})</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>(e_{2,4})</td>
<td>2</td>
<td>-5</td>
</tr>
<tr>
<td>(e_{2,5})</td>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>(e_{2,6})</td>
<td>6</td>
<td>-20</td>
</tr>
</tbody>
</table>

### Node 3

<table>
<thead>
<tr>
<th>id</th>
<th>(x)</th>
<th>(s_3(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e_{3,1})</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>(e_{3,2})</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>(e_{3,3})</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(e_{3,4})</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>(e_{3,5})</td>
<td>5</td>
<td>-6</td>
</tr>
<tr>
<td>(e_{3,6})</td>
<td>6</td>
<td>-10</td>
</tr>
</tbody>
</table>

**Set 1**

\[
R = \begin{array}{c|c|c|c|c|c|c}
  x & s(x) & T_1 & T_1/m & \tau(x) & \tau'(x) & \tau^+(x) \\
 1 & 10 & 001 & 24.6 & -4.6 & 24.6 & 24.6 \\
 3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
 5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\
 6 & -45 & 111 & -45 & -45 & 45 & 45 \\
\end{array}
\]

\(T_1 = 22, \quad T_1/m = 22/3\)

**Set 2**

\(T_2 = 45\)

\(s_3(3) = ?\)
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\(\text{After collecting all scores, the coordinator can determine top-} k |s(x)|.\)

\[\begin{array}{c|c|c}
\text{id} & x & s_1(x) \\
\hline
\text{e}_{1,1} & 5 & 20 \\
\text{e}_{1,2} & 6 & -15 \\
\text{e}_{1,3} & 3 & -30 \\
\text{e}_{1,4} & 3 & 6 \\
\text{e}_{1,5} & 6 & -10 \\
\end{array} \]

\[\begin{array}{c|c|c}
\text{id} & x & s_2(x) \\
\hline
\text{e}_{2,1} & 5 & 12 \\
\text{e}_{2,2} & 4 & 7 \\
\text{e}_{2,3} & 1 & 2 \\
\text{e}_{2,4} & 3 & -14 \\
\text{e}_{2,5} & 6 & -20 \\
\end{array} \]

\[\begin{array}{c|c|c}
\text{id} & x & s_3(x) \\
\hline
\text{e}_{3,1} & 1 & 10 \\
\text{e}_{3,2} & 3 & 6 \\
\text{e}_{3,3} & 4 & 5 \\
\text{e}_{3,4} & 2 & -3 \\
\text{e}_{3,5} & 5 & -6 \\
\text{e}_{3,6} & 6 & -10 \\
\end{array} \]
Exact Top-$k$ Wavelet Coefficients: Our Solution

After collecting all scores, the coordinator can determine top-$k |s(x)|$. $k = 1$

$T_1 = 22$, $T_1/m = 22/3$

$T_2 = 45$

$s_3(3) = ?$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Our Solution

\(k = 1\)

\[
R
\begin{array}{ccc}
\text{id} & x & s_j(x) \\
e_{1.1} & 5 & 20 \\
e_{1.5} & 6 & -15 \\
e_{1.6} & 3 & -30 \\
e_{2.1} & 5 & 12 \\
e_{2.5} & 3 & -14 \\
e_{2.6} & 6 & -20 \\
e_{3.1} & 1 & 10 \\
e_{3.2} & 3 & 6 \\
e_{3.6} & 6 & -10 \\
\end{array}
\]

\[
R
\begin{array}{ccc}
x & s(x) & F_i \\
1 & 10 & 001 \\
3 & 38 & 111 \\
5 & 32 & 110 \\
6 & -45 & 111 \\
\end{array}
\]

\(T_1 = 22, \ T_1/m = 22/3\)

\(T_2 = 45\)

\(\star s(6) = -45\)

\(s_3(6) = -45\star\)

\(s_3(3) = ?\)

\(\star\)

• After collecting all scores, the coordinator can determine top-\(k\) \(|s(x)|\).
Exact Top-$k$ Wavelet Coefficients: Our Solution

$k = 1$

<table>
<thead>
<tr>
<th>$R$</th>
<th>id</th>
<th>$x$</th>
<th>$s_j(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
<td></td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
<td></td>
</tr>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
<td></td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
<td></td>
</tr>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>$e_{3,2}$</td>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$R$</th>
<th>$x$</th>
<th>$\hat{s}(x)$</th>
<th>$F_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-38</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-32</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-45</td>
<td>111</td>
<td></td>
</tr>
</tbody>
</table>

$T_1 = 22$, $T_1/m = 22/3$

$T_2 = 45$

$s(6) = -45$

Round 3 End

<table>
<thead>
<tr>
<th>node 1</th>
<th>id</th>
<th>$x$</th>
<th>$s_1(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{1,1}$</td>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>$e_{1,2}$</td>
<td>2</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>$e_{1,3}$</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>$e_{1,4}$</td>
<td>4</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>$e_{1,5}$</td>
<td>6</td>
<td>-15</td>
<td></td>
</tr>
<tr>
<td>$e_{1,6}$</td>
<td>3</td>
<td>-30</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 2</th>
<th>id</th>
<th>$x$</th>
<th>$s_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{2,1}$</td>
<td>5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>$e_{2,2}$</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>$e_{2,3}$</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$e_{2,4}$</td>
<td>2</td>
<td>-5</td>
<td></td>
</tr>
<tr>
<td>$e_{2,5}$</td>
<td>3</td>
<td>-14</td>
<td></td>
</tr>
<tr>
<td>$e_{2,6}$</td>
<td>6</td>
<td>-20</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node 3</th>
<th>id</th>
<th>$x$</th>
<th>$s_3(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{3,1}$</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>$e_{3,2}$</td>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>$e_{3,3}$</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>$e_{3,4}$</td>
<td>2</td>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>$e_{3,5}$</td>
<td>5</td>
<td>-6</td>
<td></td>
</tr>
<tr>
<td>$e_{3,6}$</td>
<td>6</td>
<td>-10</td>
<td></td>
</tr>
</tbody>
</table>
Outline

1. Introduction and Motivation
   - Histograms
   - MapReduce and Hadoop

2. Exact Top-$k$ Wavelet Coefficients
   - Naive Solution
   - Hadoop Wavelet Top-$k$: Our Efficient Exact Solution

3. Approximate Top-$k$ Wavelet Coefficients
   - Linearly Combinable Sketch Method
   - Our First Sampling Based Approach
   - An Improved Sampling Approach
   - Two-Level Sampling

4. Experiments

5. Conclusions
   - Hadoop Wavelet Top-$k$ in Hadoop
Approximate Top-$k$ Wavelet Coefficients

- Hadoop Wavelet Top-$k$ is a good solution if the exact top-$k$ $|w_i|$ must be retrieved, but requires multiple phases.
Approximate Top-\(k\) Wavelet Coefficients

- Hadoop Wavelet Top-\(k\) is a good solution if the exact top-\(k\) \(|w_i|\) must be retrieved, but requires multiple phases.
- If we are allowed an approximation, we could further improve:
Hadoop Wavelet Top-$k$ is a good solution if the exact top-$k$ $|w_i|$ must be retrieved, but requires multiple phases.

If we are allowed an approximation, we could further improve:

1. communication cost
Hadoop Wavelet Top-$k$ is a good solution if the exact top-$k$ $|w_i|$ must be retrieved, but requires multiple phases.

If we are allowed an approximation, we could further improve:

1. communication cost
2. number of MapReduce rounds
Approximate Top-$k$ Wavelet Coefficients

- Hadoop Wavelet Top-$k$ is a good solution if the exact top-$k$ $|w_i|$ must be retrieved, but requires multiple phases.
- If we are allowed an approximation, we could further improve:
  1. communication cost
  2. number of MapReduce rounds
  3. amount of I/O incurred
Some natural improvement attempts:
Some natural improvement attempts:
1. Approximate distributed top-$k$. 

The state of the art wavelet sketch is the GCS Sketch [CGS06]. The GCS gives us, for $v = v_1 + v_2$
$$GCS(v) = GCS(v_1) + GCS(v_2)$$

Some natural improvement attempts:

1. Approximate distributed top-\(k\).
2. Approximating local coefficients with a linearly combinable sketch.
Some natural improvement attempts:

1. Approximate distributed top-$k$.
2. Approximating local coefficients with a linearly combinable sketch.

- For set $A = A_1 \cup A_2$,
  $\text{Sketch}(A) = \text{Sketch}(A_1) \text{ op } \text{Sketch}(A_2)$ for operator $\text{op}$.
Some natural improvement attempts:

1. Approximate distributed top-\(k\).
2. Approximating local coefficients with a linearly combinable sketch.
   - For set \(A = A_1 \cup A_2\), \(\text{Sketch}(A) = \text{Sketch}(A_1) \text{ op } \text{Sketch}(A_2)\) for operator \(\text{op}\).
   - The state of the art wavelet sketch is the GCS Sketch [CGS06].

Some natural improvement attempts:

1. Approximate distributed top-\(k\).
2. Approximating local coefficients with a linearly combinable sketch.

- For set \(A = A_1 \cup A_2\),
  \(\text{Sketch}(A) = \text{Sketch}(A_1) \circ \text{Sketch}(A_2)\) for operator \(\circ\).
- The state of the art wavelet sketch is the GCS Sketch [CGS06].
- The GCS gives us, for \(v = v_1 + v_2\)
  \(GCS(v) = GCS(v_1) + GCS(v_2)\)

Some natural improvement attempts:

1. Approximate distributed top-$k$.
2. Approximating local coefficients with a linearly combinable sketch.
   - For set $A = A_1 \cup A_2$,
     $\text{Sketch}(A) = \text{Sketch}(A_1) \circ \circ \text{Sketch}(A_2)$ for operator $\circ \circ$.
   - The state of the art wavelet sketch is the GCS Sketch [CGS06].
   - The GCS gives us, for $v = v_1 + v_2$
     $\text{GCS}(v) = \text{GCS}(v_1) + \text{GCS}(v_2)$
3. Random sampling techniques.

Outline

1 Introduction and Motivation
   • Histograms
   • MapReduce and Hadoop

2 Exact Top-k Wavelet Coefficients
   • Naive Solution
   • Hadoop Wavelet Top-k: Our Efficient Exact Solution

3 Approximate Top-k Wavelet Coefficients
   • Linearly Combinable Sketch Method
   • Our First Sampling Based Approach
   • An Improved Sampling Approach
   • Two-Level Sampling

4 Experiments

5 Conclusions
   • Hadoop Wavelet Top-k in Hadoop
Approximate Top-\( k \) Wavelet Coefficients: Sketch

JobTracker

split 1

split 2

split 3

split 4

\((x, \text{null})\)

Mapper
Approximate Top-\(k\) Wavelet Coefficients: Sketch

JobTracker

split 1
split 2
split 3
split 4

Mapper

In-Memory Map

(x, null)

(x, 1)
Approximate Top-$k$ Wavelet Coefficients: Sketch

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-\(k\) Wavelet Coefficients: Sketch
Approximate Top-k Wavelet Coefficients: Sketch

JobTracker

split 1
split 2
split 3
split 4

Mapper

In-Memory
Map

(x, null)

(x, v_j(x))

Sketch

close()
Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-\(k\) Wavelet Coefficients: Sketch

Jeffrey Jestes, Ke Yi, Feifei Li
Approximate Top-$k$ Wavelet Coefficients: Sketch

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Sketch

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Sketch

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-k Wavelet Coefficients: Sketch

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Outline

1. Introduction and Motivation
   - Histograms
   - MapReduce and Hadoop

2. Exact Top-$k$ Wavelet Coefficients
   - Naive Solution
   - Hadoop Wavelet Top-$k$: Our Efficient Exact Solution

3. Approximate Top-$k$ Wavelet Coefficients
   - Linearly Combinable Sketch Method
   - Our First Sampling Based Approach
   - An Improved Sampling Approach
   - Two-Level Sampling

4. Experiments

5. Conclusions
   - Hadoop Wavelet Top-$k$ in Hadoop
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

$n_j$ Records in split $j$
Well known fact: to approximate each $v(x)$ with standard deviation $\sigma = O(\varepsilon n)$ a sample of size $\Theta(1/\varepsilon^2)$ is required.
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

Well known fact: to approximate each $v(x)$ with standard deviation $\sigma = O(\varepsilon n)$ a sample of size $\Theta(1/\varepsilon^2)$ is required.

Node $j$ samples $t_j = n_j \cdot p$ records where $p = 1/\varepsilon^2 n$.

$n_j$ Records in split $j$
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

Well known fact: to approximate each \(v(x)\) with standard deviation \(\sigma = O(\varepsilon n)\) a sample of size \(\Theta(1/\varepsilon^2)\) is required.

Node \(j\) samples \(t_j = n_j \cdot p\) records where \(p = 1/\varepsilon^2 n\).

\(n_j\) Records in split \(j\)
Well known fact: to approximate each $v(x)$ with standard deviation $\sigma = O(\varepsilon n)$ a sample of size $\Theta(1/\varepsilon^2)$ is required.

Node $j$ samples $t_j = n_j \cdot p$ records where $p = 1/\varepsilon^2 n$.
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

$s_j(x)$: Sampled Frequency Counts
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

$s_j(x)$: Sampled Frequency Counts

Emit

Emitted $s_j(x)$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

\(s_j(x)\): Sampled Frequency Counts

Emit

Coordinator

Emitted \(s_j(x)\)

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

Emitted $s_1(x)$

Emitted $s_m(x)$

Coordinator

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

\[ s(x) = \sum_{j=1}^{m} s_j(x) \]
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

\[\hat{v}(x) = \frac{s(x)}{p}\] is \(v(x)\)'s unbiased estimator.

\[s(x) = \sum_{j=1}^{m} s_j(x)\]
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

- **Note**: \(\varepsilon\) must be small for \(\hat{v}\) to approximate \(v\) well.
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

- **Note:** \(\varepsilon\) must be small for \(\hat{v}\) to approximate \(v\) well.
- Typical values for \(\varepsilon\) are \(10^{-4}\) to \(10^{-6}\).
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

- **Note:** $\varepsilon$ must be small for $\hat{\mathbf{v}}$ to approximate $\mathbf{v}$ well.
  
  - Typical values for $\varepsilon$ are $10^{-4}$ to $10^{-6}$.
  
  - The communication for basic sampling is $O(1/\varepsilon^2)$. 

Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

- **Note:** \(\varepsilon\) must be small for \(\hat{v}\) to approximate \(v\) well.
  - Typical values for \(\varepsilon\) are \(10^{-4}\) to \(10^{-6}\).
- The communication for basic sampling is \(O(1/\varepsilon^2)\).
  - With 1 byte keys, 100MB to 1TB of data must be communicated!
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

- **Note:** \(\varepsilon\) must be small for \(\hat{v}\) to approximate \(v\) well.
  - Typical values for \(\varepsilon\) are \(10^{-4}\) to \(10^{-6}\).
- The communication for basic sampling is \(O(1/\varepsilon^2)\).
  - With 1 byte keys, 100MB to 1TB of data must be communicated!
- We improve basic random sampling with *Improved Sampling*.
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

- **Note:** \(\varepsilon\) must be small for \(\hat{v}\) to approximate \(v\) well.
  - Typical values for \(\varepsilon\) are \(10^{-4}\) to \(10^{-6}\).
- The communication for basic sampling is \(O(1/\varepsilon^2)\).
  - With 1 byte keys, 100MB to 1TB of data must be communicated!
- We improve basic random sampling with *Improved Sampling*.
  - **Key idea:** ignore sampled keys with small frequencies in a split.
Outline

1 Introduction and Motivation
   • Histograms
   • MapReduce and Hadoop

2 Exact Top-\(k\) Wavelet Coefficients
   • Naive Solution
   • Hadoop Wavelet Top-\(k\): Our Efficient Exact Solution

3 Approximate Top-\(k\) Wavelet Coefficients
   • Linearly Combinable Sketch Method
   • Our First Sampling Based Approach
     • An Improved Sampling Approach
   • Two-Level Sampling

4 Experiments

5 Conclusions
   • Hadoop Wavelet Top-\(k\) in Hadoop
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

Each node sends at most $t_j/\varepsilon$ keys. The total communication is $O(m/\varepsilon)$.

$E[\hat{v}(x)]$ may be $\varepsilon n$ away from $v(x)$ as $s_j(x)$ < $\varepsilon$ are ignored.

$n_j$ Records in split
Node $j$ samples $t_j = n_j \cdot p$ records using Basic Sampling, where $p = 1/\varepsilon^2 n$. 

$n_j$ Records in split
Node $j$ samples $t_j = n_j \cdot p$ records using Basic Sampling, where $p = 1/\varepsilon^2 n$. 

$n_j$ Records in split
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

Node \(j\) samples \(t_j = n_j \cdot p\) records using Basic Sampling, where \(p = 1/\varepsilon^2 n\).

\(n_j\) Records in split

\(s_j(x)\): Sampled Frequency Counts

\(s_j(x)\): Sampled Frequency Counts

Each node sends at most \(t_j / (\varepsilon t_j) = 1/\varepsilon\) keys.

The total communication is \(O(m/\varepsilon)\).

\(E[\hat{v}(x)]\) may be \(\varepsilon n\) away from \(v(x)\) as \(s_j(x) < \varepsilon t_j\) are ignored.
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

Node $j$ sends $(x, s_j(x))$ only if $s_j(x) > \varepsilon t_j$.

- The error in $s(x)$ is $\leq \sum_{j=1}^{m} \varepsilon t_j = \varepsilon pn = 1/\varepsilon$.
Approximate Top-k Wavelet Coefficients: Improved Sampling

Node $j$ sends $(x, s_j(x))$ only if $s_j(x) > \varepsilon t_j$.

- The error in $s(x)$ is $\leq \sum_{j=1}^{m} \varepsilon t_j = \varepsilon pn = 1/\varepsilon$.

$s_j(x)$: Sampled Frequency Counts
Node $j$ sends $(x, s_j(x))$ only if $s_j(x) > \varepsilon t_j$.

- The error in $s(x)$ is $\leq \sum_{j=1}^{m} \varepsilon t_j = \varepsilon pn = 1/\varepsilon$. 

$s_j(x)$: Sampled Frequency Counts

Emit

Emitted $s_j(x)$
Node $j$ sends $(x, s_j(x))$ only if $s_j(x) > \varepsilon t_j$.

* The error in $s(x)$ is $\leq \sum_{j=1}^{m} \varepsilon t_j = \varepsilon pn = 1/\varepsilon$. 

$E[\hat{v}(x)]$ may be $\varepsilon n$ away from $v(x)$ as $s_j(x) < \varepsilon t_j$ are ignored.
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

Each node sends at most $t_j/\varepsilon t_j = 1/\varepsilon$ keys. The total communication is $O(m/\varepsilon)$. $E[\hat{v}(x)]$ may be $\varepsilon n$ away from $v(x)$ as $s_j(x)$ are ignored.
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

Each node sends at most \(t_j / (\varepsilon t_j) = 1 / \varepsilon\) keys. The total communication is \(O(m/\varepsilon)\).

\[ E[\hat{v}(x)] \text{ may be } \varepsilon n \text{ away from } v(x) \text{ as } s_j(x) < \varepsilon t_j \text{ are ignored.} \]
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

Each node sends at most $t_j/\varepsilon t_j = 1/\varepsilon$ keys. The total communication is $O(m/\varepsilon)$. $E[\tilde{v}(x)]$ may be $\varepsilon$ away from $v(x)$ as $s_j(x) < \varepsilon t_j$ are ignored.

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-k Wavelet Coefficients: Improved Sampling

- Each node sends at most $t_j/(\varepsilon t_j) = 1/\varepsilon$ keys.
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

Each node sends at most \(t_j/(\varepsilon t_j) = 1/\varepsilon\) keys.

The total communication is \(O(m/\varepsilon)\).
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

Each node sends at most $t_j/(\varepsilon t_j) = 1/\varepsilon$ keys.
- The total communication is $O(m/\varepsilon)$.
- $E[\hat{v}(x)]$ may be $\varepsilon n$ away from $v(x)$ as $s_j(x) < \varepsilon t_j$ are ignored.
Outline

1. Introduction and Motivation
   - Histograms
   - MapReduce and Hadoop

2. Exact Top-k Wavelet Coefficients
   - Naive Solution
   - Hadoop Wavelet Top-k: Our Efficient Exact Solution

3. Approximate Top-k Wavelet Coefficients
   - Linearly Combinable Sketch Method
   - Our First Sampling Based Approach
   - An Improved Sampling Approach
   - Two-Level Sampling

4. Experiments

5. Conclusions
   - Hadoop Wavelet Top-k in Hadoop
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

\(n_j\) Records in split
Node $j$ samples $t_j = n_j \cdot p$ records using
*Basic Sampling*, where $p = 1/\varepsilon^2 n$. $n_j$ Records in split

$\hat{s}(x)$

If $(x, s_j(x))$ received, $\rho(x) = \rho(x) + s_j(x)$.

Else if $(x, \text{null})$ received, $M(x) = M(x) + 1$.

Finally, $\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}$.

Then, $\hat{v}(x) = \hat{s}(x) / p$ is an unbiased estimator for $v(x)$.
Node $j$ samples $t_j = n_j \cdot p$ records using
Basic Sampling, where $p = 1/\varepsilon^2 n$. 

$n_j$ Records in split
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

Node $j$ samples $t_j = n_j \cdot p$ records using Basic Sampling, where $p = 1/\varepsilon^2 n$. 

$n_j$ Records in split 

$s_j(x):$ Sampled Frequency Counts
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

\(s_j(x):\) Sampled Frequency Counts

To construct \(\hat{s}(x)\). If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\). Else if \((x, \text{null})\) received, \(M(x) = M(x) + 1\). Finally, \(\hat{s}(x) = \rho(x) + M(x) / \varepsilon \sqrt{m}\).

Then, \(\hat{v}(x) = \hat{s}(x) / p\) is an unbiased estimator for \(v(x)\).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

Sample record \(x\) with probability \(\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}\).

- If \(s_j(x) \geq 1/(\varepsilon \sqrt{m})\), emit \((x, s_j(x))\).
- Else emit \((x, \text{null})\) with probability \(\varepsilon \sqrt{m} \cdot s_j(x)\).

\(s_j(x)\): Sampled Frequency Counts

To construct \(\hat{s}(x)\).

If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\).

Else if \((x, \text{null})\) received, \(M(x) = M(x) + 1\).

Finally, \(\hat{s}(x) = \rho(x) / \varepsilon \sqrt{m}\).

Then, \(\hat{v}(x) = \hat{s}(x) / p\) is an unbiased estimator for \(v(x)\).
Sample record $x$ with probability $\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}$.

- If $s_j(x) \geq 1/(\varepsilon \sqrt{m})$, emit $(x, s_j(x))$.
- Else emit $(x, \text{null})$ with probability $\varepsilon \sqrt{m} \cdot s_j(x)$.

$s_j(x)$: Sampled Frequency Counts
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

Sample record \(x\) with probability \(\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}\).

- If \(s_j(x) \geq 1/(\varepsilon \sqrt{m})\), emit \((x, s_j(x))\).
- Else emit \((x, \text{null})\) with probability \(\varepsilon \sqrt{m} \cdot s_j(x)\).

\(s_j(x)\): Sampled Frequency Counts
Sample record $x$ with probability $\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}$.

- If $s_j(x) \geq 1/(\varepsilon \sqrt{m})$, emit $(x, s_j(x))$.
- Else emit $(x, \text{null})$ with probability $\varepsilon \sqrt{m} \cdot s_j(x)$.

$s_j(x)$: Sampled Frequency Counts

$\hat{s}_j(x)$: sampled $s_j(x)$
Sample record $x$ with probability $\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}$.

- If $s_j(x) \geq 1/(\varepsilon \sqrt{m})$, emit $(x, s_j(x))$.
- Else emit $(x, \text{null})$ with probability $\varepsilon \sqrt{m} \cdot s_j(x)$.

$s_j(x)$: Sampled Frequency Counts

$\hat{s}_j(x)$: sampled $s_j(x)$

$\varepsilon$: Epsilon

$\sqrt{m}$: Square root of $m$

$s_j(x)$: Sampled Frequency Counts
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

To construct \(\hat{s}(x)\).

If \((x, s_j(x))\) received, 
\[
\rho(x) = \rho(x) + s_j(x).
\]

Else if \((x, \text{null})\) received, 
\[
M(x) = M(x) + 1.
\]

Finally, 
\[
\hat{s}(x) = \frac{\rho(x) + M(x)}{\epsilon \sqrt{m}}.
\]

Then, 
\[
\hat{v}(x) = \frac{\hat{s}(x)}{p}
\]
is an unbiased estimator for \(v(x)\).
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

To construct $\hat{s}(x)$.

Coordinator

$\hat{s}_1(x)$: sampled $s_1(x)$

$\hat{s}_m(x)$: sampled $s_m(x)$

$\hat{s}(x)$: estimator of $s(x)$

$\hat{s}(x) = \rho(x) + \frac{M(x)}{\varepsilon \sqrt{m}}$. Then, $\hat{v}(x) = \hat{s}(x)/p$ is an unbiased estimator for $v(x)$. 

Jeffrey Jestes, Ke Yi, Feifei Li
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

To construct \(\hat{s}(x)\).

- If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

To construct \(\hat{s}(x)\).
- If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\).
- Else if \((x, \text{null})\) received, \(M(x) = M(x) + 1\).
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

To construct $\hat{s}(x)$.

- If $(x, s_j(x))$ received, $\rho(x) = \rho(x) + s_j(x)$.
- Else if $(x, null)$ received, $M(x) = M(x) + 1$.
- Finally, $\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}$. 
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**To construct $\hat{s}(x)$.**

- If $(x, s_j(x))$ received, $\rho(x) = \rho(x) + s_j(x)$.
- Else if $(x, null)$ received, $M(x) = M(x) + 1$.

Finally, $\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}$.

Then, $\hat{v}(x) = \hat{s}(x)/p$ is an unbiased estimator for $v(x)$. 

Jeffrey Jestes, Ke Yi, Feifei Li | Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

Theorem

$\hat{s}(x)$ is an unbiased estimator of $s(x)$ with standard deviation at most $1/\varepsilon$. 

Corollary

$\hat{v}(x)$ is an unbiased estimator of $v(x)$ with standard deviation at most $\varepsilon n$. 

Theorem

• $\hat{w}_i$ is an unbiased estimator for any $w_i$. 
• Recall $w_i = \langle v, \psi_i \rangle$, for $\psi_i = (\varphi_{j+1} - \varphi_{j}, 2k, \varphi_{j+1}, 2k+1) / \sqrt{u}/2^j$ where $\varphi$ is a $[0,1]$ vector defined for $j = 1, ..., \log u$ and $k = 0, ..., 2^j - 1$. The variance of $\hat{w}_i$ is bounded by $\varepsilon^2 j n u \sqrt{m} \sum (2^k + 2) u/2^j + 1 s(x)$. 

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$. 

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**
\[ \hat{s}(x) \text{ is an unbiased estimator of } s(x) \text{ with standard deviation at most } 1/\varepsilon. \]

**Corollary**
\[ \hat{v}(x) \text{ is an unbiased estimator of } v(x) \text{ with standard deviation at most } \varepsilon n. \]
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

\( \hat{s}(x) \) is an unbiased estimator of \( s(x) \) with standard deviation at most \( 1/\varepsilon \).

**Corollary**

\( \hat{v}(x) \) is an unbiased estimator of \( v(x) \) with standard deviation at most \( \varepsilon n \).

**Theorem**

- \( \hat{w}_i \) is an unbiased estimator for any \( w_i \).
- Recall \( w_i = \langle v, \psi_i \rangle \), for \( \psi_i = (-\phi_{j+1,2k} + \phi_{j+1,2k+1}) / \sqrt{u/2^j} \) where \( \phi \) is a \([0, 1]\) vector defined for \( j = 1, \ldots, \log u \) and \( k = 0, \ldots, 2^j - 1 \). The variance of \( \hat{w}_i \) is bounded by \( \frac{\varepsilon^2 n}{u \sqrt{m}} \sum_{x=2ku/2^{j+1}+1} s(x) \).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

\(\hat{s}(x)\) is an unbiased estimator of \(s(x)\) with standard deviation at most \(1/\varepsilon\).

**Corollary**

\(\hat{v}(x)\) is an unbiased estimator of \(v(x)\) with standard deviation at most \(\varepsilon n\).

**Theorem**

- \(\hat{w}_i\) is an unbiased estimator for any \(w_i\).
- Recall \(w_i = \langle v, \psi_i \rangle\), for \(\psi_i = (-\phi_{j+1,2k} + \phi_{j+1,2k+1})/\sqrt{u/2^j}\) where \(\phi\) is a \([0, 1]\) vector defined for \(j = 1, \ldots, \log u\) and \(k = 0, \ldots, 2^j - 1\). The variance of \(\hat{w}_i\) is bounded by \(\varepsilon^2 n / u \sqrt{m} \sum_{x=2 ku/2^{j+1}+1}^{2^{j+1} u/2^j} s(x)\).

**Theorem**

The expected total communication cost of our two-level sampling algorithm is \(O(\sqrt{m/\varepsilon})\).
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

JobTracker

\[ n_j = \text{records in split } j \]
\[ s_j = \text{split } j \text{ sample frequency vector} \]
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

\(n_j = \) records in split \(j\)
\(s_j = \) split \(j\) sample frequency vector

RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

\[ n_j = \text{records in split } j \]
\[ s_j = \text{split } j \text{ sample frequency vector} \]

1. RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.
   - \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

1. RandomizedRecordReader $j$ ($RR_j$) samples $n_j/\varepsilon^2 n$ records.
   - $RR_j$ randomly selects $n_j/\varepsilon^2 n$ offsets in split $j$.
   - $RR_j$ sorts the offsets in ascending order then seeks the record at each sampled offset.

$n_j = \text{records in split } j$
$s_j = \text{split } j \text{ sample frequency vector}$
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

RandomizedRecordReader $j$ ($RR_j$) samples $n_j/\varepsilon^2 n$ records.

1. $RR_j$ randomly selects $n_j/\varepsilon^2 n$ offsets in split $j$.
2. $RR_j$ sorts the offsets in ascending order then seeks the record at each sampled offset.

$n_j =$ records in split $j$
$s_j =$ split $j$ sample frequency vector
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.

1. \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
2. \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.

\(n_j = \) records in split \(j\)
\(s_j = \) split \(j\) sample frequency vector
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.

1. \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
2. \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

\(n_j=\) records in split \(j\)

\(s_j=\) split \(j\) sample frequency vector

1. **RandomizedRecordReader** \(j\) \((RR_j)\) samples \(n_j/\varepsilon^2 n\) records.
   - \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
   - \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

Mapper \(j\) samples key \(x\) from \(s\) with probability \(\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}\).
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Mapper** $j$ samples key $x$ from $s$ with probability $\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}$.
- If $s_j(x) \geq 1/(\varepsilon \sqrt{m})$, emit $(x, s_j(x))$. 

$n_j =$ records in split $j$
$s_j =$ split $j$ sample frequency vector
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Mapper $j$ samples key $x$ from $s$ with probability $\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}$.

- If $s_j(x) \geq 1/(\varepsilon \sqrt{m})$, emit $(x, s_j(x))$.
- Else emit $(x, 0)$ with probability $\varepsilon \sqrt{m} \cdot s_j(x)$.
**Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling**

Mapper \(j\) samples key \(x\) from \(s\) with probability \(\min\{\varepsilon \sqrt{m} \cdot s_j(x), 1\}\).

- If \(s_j(x) \geq 1/(\varepsilon \sqrt{m})\), emit \((x, s_j(x))\).
- Else emit \((x, 0)\) with probability \(\varepsilon \sqrt{m} \cdot s_j(x)\).

\(n_j\) = records in split \(j\)

\(s_j\) = split \(j\) sample frequency vector

\[n_j = \text{records in split } j\]

\[s_j = \text{split } j \text{ sample frequency vector}\]
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

**Diagram:**
- JobTracker
- RandomizedRecordReader
- MapRunner
- Mapper
- Reducer

**Nodes:**
- **RandomizedRecordReader**
  - (x, null)

**Edges:**
- Split 1 → MapRunner
- Split 2 → MapRunner
- Split 3 → MapRunner
- Split 4 → MapRunner
- (x, s_j(x)) → Mapper
- (x, s_j(x)) | 0 → Reducer
- (x, s_j(x)) | 0 → Reducer

**Annotations:**
- \( n_j \) = records in split \( j \)
- \( s_j \) = split \( j \) sample frequency vector

**Equations:**
- Construct \( \hat{s}(x) \).

**Textual Explanation:**
- In-Memory Map
- Mapper
- Reducer
- \( M(x) = M(x) + 1 \)
- \( \hat{s}(x) = \rho(x) + \frac{M(x)}{\epsilon \sqrt{m}} \)

**Diagram Details:**
- Split 1, Split 2, Split 3, Split 4
- JobTracker
- Mapper
- Reducer
- In-Memory Map
- RandomizedRecordReader

**Additional Notes:**
- Jeffrey Jestes, Ke Yi, Feifei Li
- Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

Construct $\hat{s}(x)$.

- If $(x, s_j(x))$ received, $\rho(x) = \rho(x) + s_j(x)$.

$n_j =$ records in split $j$

$s_j =$ split $j$ sample frequency vector
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

```
JobTracker

RandomizedRecordReader

split 1
split 2
split 3
split 4

MapRunner

Mapper

Reducer

n_j = records in split j
s_j = split j sample frequency vector

n_j = \text{records in split } j
s_j = \text{split } j \text{ sample frequency vector}

\textbf{Construct } \hat{s}(x). \textbf{ )}

- If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\).
- Else if \((x, 0)\) received, \(M(x) = M(x) + 1\).

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
```
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

Construct \(\hat{s}(x)\).

- If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\).
- Else if \((x, 0)\) received, \(M(x) = M(x) + 1\).

Finally, \(\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}\).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Construct \(\hat{s}(x)\).**
- If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\).
- Else if \((x, 0)\) received, \(M(x) = M(x) + 1\).

**Finally, \(\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}\).**

**Reducer uses \(\hat{v}(x) = \hat{s}(x)/\rho\), our unbiased estimator for \(v(x)\).**
**Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling**

- **JobTracker**
- **RandomizedRecordReader**
- **Mapper**
- **Reducer**
- **In-Memory Map**

**Centralized Wavelet Top-\(k\)**

\(n_j\) = records in split \(j\)
\(s_j\) = split \(j\) sample frequency vector

---

**Construct** \(\hat{s}(x)\).

- If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\).
- Else if \((x, 0)\) received, \(M(x) = M(x) + 1\).

Finally, \(\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}\).

Reducer uses \(\hat{v}(x) = \hat{s}(x)/\rho\), our unbiased estimator for \(v(x)\).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

Construct \(\hat{s}(x)\).
- If \((x, s_j(x))\) received, \(\rho(x) = \rho(x) + s_j(x)\).
- Else if \((x, 0)\) received, \(M(x) = M(x) + 1\).
- Finally, \(\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}\).
- Reducer uses \(\hat{v}(x) = \hat{s}(x)/\rho\), our unbiased estimator for \(v(x)\).
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

JobTracker

RandomizedRecordReader

MapRunner

Mapper

Reducer

In-Memory Map

Centralized Wavelet Top-$k$

$n_j = \text{records in split } j$

$s_j = \text{split } j \text{ sample frequency vector}$

Construct $\hat{s}(x)$.

- If $(x, s_j(x))$ received, $\rho(x) = \rho(x) + s_j(x)$.
- Else if $(x, 0)$ received, $M(x) = M(x) + 1$.

Finally, $\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}$.

Reducer uses $\hat{v}(x) = \hat{s}(x)/\rho$, our unbiased estimator for $v(x)$.
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

$n_j =$ records in split $j$
$s_j =$ split $j$ sample frequency vector

Construct $\hat{s}(x)$.
- If $(x, s_j(x))$ received, $\rho(x) = \rho(x) + s_j(x)$.
- Else if $(x, 0)$ received, $M(x) = M(x) + 1$.
- Finally, $\hat{s}(x) = \rho(x) + M(x)/\varepsilon \sqrt{m}$.
- Reducer uses $\hat{v}(x) = \hat{s}(x)/\rho$, our unbiased estimator for $v(x)$.
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys.
Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys.

The communication for basic sampling is $O(1/\varepsilon^2)$.
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys.
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately $400$MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.
  - Approximately $40$MB of data must be communicated.
- The communication for two-level sampling is $O(\sqrt{m}/\varepsilon)$.
  - Only $1.2$MB of data needs to be communicated!

330-fold reduction over basic sampling and 33-fold reduction over improved sampling!
Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys.

The communication for basic sampling is $O(1/\varepsilon^2)$.

- Approximately 400MB of data must be communicated!

The communication for improved sampling is $O(m/\varepsilon)$.

- Only 1.2MB of data needs to be communicated!

330-fold reduction over basic sampling and 33-fold reduction over improved sampling!
Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys.

The communication for basic sampling is $O(1/\varepsilon^2)$.
- Approximately 400MB of data must be communicated!

The communication for improved sampling is $O(m/\varepsilon)$.
- Approximately 40MB of data must be communicated.

330-fold reduction over basic sampling and 33-fold reduction over improved sampling!
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys.
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.
  - Approximately 40MB of data must be communicated.
- The communication for two-level sampling is $O(\sqrt{m}/\varepsilon)$.

330-fold reduction over basic sampling and
33-fold reduction over improved sampling!
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys.
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.
  - Approximately 40MB of data must be communicated.
- The communication for two-level sampling is $O(\sqrt{m}/\varepsilon)$.
  - Only 1.2MB of data needs to be communicated!
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

- Consider: $\varepsilon = 10^{-4}$, $m = 10^3$, and 4-byte keys.
- The communication for basic sampling is $O(1/\varepsilon^2)$.
  - Approximately 400MB of data must be communicated!
- The communication for improved sampling is $O(m/\varepsilon)$.
  - Approximately 40MB of data must be communicated.
- The communication for two-level sampling is $O(\sqrt{m}/\varepsilon)$.
  - Only 1.2MB of data needs to be communicated!
  - 330-fold reduction over basic sampling and 33-fold reduction over improved sampling!
Introduction and Motivation
- Histograms
- MapReduce and Hadoop

Exact Top-k Wavelet Coefficients
- Naive Solution
- Hadoop Wavelet Top-k: Our Efficient Exact Solution

Approximate Top-k Wavelet Coefficients
- Linearly Combinable Sketch Method
- Our First Sampling Based Approach
- An Improved Sampling Approach
- Two-Level Sampling

Experiments

Conclusions
- Hadoop Wavelet Top-k in Hadoop
Experiments: Algorithms

- We implement the following methods in Hadoop 0.20.2:
  - Exact Methods: The baseline solution is denoted \(\text{Send-V}\). Our three round exact solution is denoted \(\text{H-WTopk}\), (meaning “Hadoop Wavelet Top-k”).
  - Approximate Methods: Improved Sampling is denoted \(\text{Improved-S}\). Two-Level Sampling is denoted \(\text{TwoLevel-S}\). The Sketch-Based Approximation using the GCS-Sketch is denoted \(\text{Send-Sketch}\).
We implement the following methods in Hadoop 0.20.2:

- **Exact Methods:**
  - The baseline solution is denoted *Send-V*,

- **Approximate Methods:**
  - Improved Sampling is denoted *Improved-S*.
  - Two-Level Sampling is denoted *TwoLevel-S*.
  - The Sketch-Based Approximation using the GCS-Sketch is denoted *Send-Sketch*.
Experiments: Algorithms

We implement the following methods in Hadoop 0.20.2:

- **Exact Methods:**
  - The baseline solution is denoted *Send-V*,
  - Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-k").
We implement the following methods in Hadoop 0.20.2:

- **Exact Methods:**
  - The baseline solution is denoted \textit{Send-V},
  - Our three round exact solution is denoted \textit{H-WTopk}, (meaning "Hadoop Wavelet Top-k").

- **Approximate Methods:**
We implement the following methods in Hadoop 0.20.2:

**Exact Methods:**
- The baseline solution is denoted *Send-V*.
- Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-k").

**Approximate Methods:**
- *Improved Sampling* is denoted *Improved-S*. 
We implement the following methods in Hadoop 0.20.2:

- **Exact Methods:**
  - The baseline solution is denoted *Send-V*.
  - Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-k").

- **Approximate Methods:**
  - *Improved Sampling* is denoted *Improved-S*.
  - *Two-Level Sampling* is denoted *TwoLevel-S*. 
We implement the following methods in Hadoop 0.20.2:

**Exact Methods:**
- The baseline solution is denoted *Send-V*.
- Our three round exact solution is denoted *H-WTopk*, (meaning "Hadoop Wavelet Top-k").

**Approximate Methods:**
- *Improved Sampling* is denoted *Improved-S*.
- *Two-Level Sampling* is denoted *TwoLevel-S*.
- The *Sketch-Based Approximation* using the GCS-Sketch is denoted *Send-Sketch*. 
Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  - 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
  - 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
  - 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
  - 1 machine with 2GB of RAM and an Intel Core 2 1.86GHz CPU
  - One is reserved for the master (running JobTracker and NameNode).
  - One is reserved for the (only) Reducer.
- All machines are directly connected to a 1000Mbps switch.
Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
  2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
  3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
  4. 1 machine with 2GB of RAM and an Intel Core 2 1.86GHz CPU

All machines are directly connected to a 1000Mbps switch.
Experiments: Setup

Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:

1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
   - One is reserved for the master (running JobTracker and NameNode).
3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
4. 1 machine with 2GB of RAM and an Intel Core 2 1.86GHz CPU
Experiments: Setup

Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:

1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
   - One is reserved for the master (running JobTracker and NameNode).
3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
Experiments: Setup

Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:

1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
    - One is reserved for the master (running JobTracker and NameNode).
3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
    - One is reserved for the (only) Reducer.
4. 1 machine with 2GB of RAM and an Intel Core 2 1.86GHz CPU
Experiments: Setup

Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:

1. 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
2. 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
   - One is reserved for the master (running JobTracker and NameNode).
3. 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
   - One is reserved for the (only) Reducer.
4. 1 machine with 2GB of RAM and an Intel Core 2 1.86GHz CPU

Jeffrey Jestes, Ke Yi, Feifei Li
Experiments: Setup

- Experiments are performed in a heterogeneous Hadoop cluster with 16 machines:
  - 9 machines with 2GB of RAM and an Intel Xeon 1.86GHz CPU
  - 4 machines with 4GB of RAM and an Intel Xeon 2GHz CPU
    - One is reserved for the master (running JobTracker and NameNode).
  - 2 machines with 6GB of RAM and an Intel Xeon 2.13GHz CPU
    - One is reserved for the (only) Reducer.
  - 1 machine with 2GB of RAM and an Intel Core 2 1.86GHz CPU
- All machines are directly connected to a 1000Mbps switch.
Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.
We utilize the *WorldCup* dataset to test all algorithms on real data.
- There are a total of 1.35 billion records.
- Each record has 10 4 byte integer attributes including a client id and object id.
We utilize the *WorldCup* dataset to test all algorithms on real data.

- There are a total of 1.35 billion records.
- Each record has 10 4 byte integer attributes including a client id and object id.
- We assign each record a `clientobject` 4 byte integer id in $u = 2^{29}$ which is distinct for unique pairings of a client id and object id.
Experiments: Datasets

- We utilize the *WorldCup* dataset to test all algorithms on real data.
  - There are a total of 1.35 billion records.
  - Each record has 10 4-byte integer attributes including a client id and object id.
  - We assign each record a \(clientobject\) 4-byte integer id in \(u = 2^{29}\) which is distinct for unique pairings of a client id and object id.
  - *WorldCup* is stored in binary format, in total it is 50GB.
We utilize the *WorldCup* dataset to test all algorithms on real data.

- There are a total of 1.35 billion records.
- Each record has 10 4-byte integer attributes including a client id and object id.
- We assign each record a *clientobject* 4-byte integer id in $u = 2^{29}$ which is distinct for unique pairings of a client id and object id.
- *WorldCup* is stored in binary format, in total it is 50GB.

We utilize large synthetic Zipfian datasets to evaluate all methods.
We utilize the *WorldCup* dataset to test all algorithms on real data.

- There are a total of 1.35 billion records.
- Each record has 10 4 byte integer attributes including a client id and object id.
- We assign each record a *clientobject* 4 byte integer id in \( u = 2^{29} \) which is distinct for unique pairings of a client id and object id.
- *WorldCup* is stored in binary format, in total it is 50GB.

We utilize large synthetic Zipfian datasets to evaluate all methods.

- Keys are randomly permuted and discontiguous in a dataset.
We utilize the *WorldCup* dataset to test all algorithms on real data.
- There are a total of 1.35 billion records.
- Each record has 10 4-byte integer attributes including a client id and object id.
- We assign each record a *clientobject* 4-byte integer id in \( u = 2^{29} \) which is distinct for unique pairings of a client id and object id.
- *WorldCup* is stored in binary format, in total it is 50GB.

We utilize large synthetic Zipfian datasets to evaluate all methods.
- Keys are randomly permuted and discontiguous in a dataset.
- Each key is a 4-byte integer and stored in binary format.
Default values:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Zipfian skewness</td>
<td>1.1</td>
</tr>
<tr>
<td>$u$</td>
<td>max key in domain</td>
<td>$\log_2 u = 29$</td>
</tr>
<tr>
<td>$n$</td>
<td>total records</td>
<td>13.4 billion</td>
</tr>
<tr>
<td></td>
<td>dataset size</td>
<td>50GB</td>
</tr>
<tr>
<td>$\beta$</td>
<td>split size</td>
<td>256MB</td>
</tr>
<tr>
<td>$m$</td>
<td>number of splits</td>
<td>200</td>
</tr>
<tr>
<td>$B$</td>
<td>network bandwidth</td>
<td>500Mbps</td>
</tr>
</tbody>
</table>
Experiments: Vary $k$

![Graph showing communication (bytes) vs. number of coefficients ($k$). The graph indicates that as $k$ increases, the communication (bytes) also increases, with different algorithms showing varying performance.](image-url)
Experiments: Vary $k$

Number of Coefficients ($k$)

Time (Seconds)

Send−V, H−WTopk, Send−Sketch, Improved−S, TwoLevel−S

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Experiments: Vary $k$

<table>
<thead>
<tr>
<th>Number of Coefficients (k)</th>
<th>SSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Send-V</td>
</tr>
<tr>
<td></td>
<td>H-WTopk</td>
</tr>
<tr>
<td></td>
<td>Send-Sketch</td>
</tr>
<tr>
<td></td>
<td>Improved-S</td>
</tr>
<tr>
<td></td>
<td>TwoLevel-S</td>
</tr>
</tbody>
</table>

Ideal SSE
Experiments: Vary $\varepsilon$

![Graph showing the relationship between $\varepsilon$ and the SSE for different algorithms, including Ideal SSE, H-WTopk, Improved-S, and TwoLevel-S. The graph plots $\varepsilon$ on a logarithmic scale from $10^{-5}$ to $10^{-1}$ on the x-axis, and SSE on a logarithmic scale from $10^{14}$ to $10^{22}$ on the y-axis.]
Experiments: Vary $\varepsilon$

![Graph showing communication (bytes) vs $\varepsilon$ for Improved-S and TwoLevel-S.](image)

- **Improved-S**
- **TwoLevel-S**
Experiments: Vary $\varepsilon$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Experiments: Vary $n$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Experiments: Vary $n$

![Graph showing the relationship between dataset size and time for different methods.]

- Send-V
- H-WTopk
- Send-Sketch
- Improved-S
- TwoLevel-S

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Experiments: Vary $u$

![Graph showing communication (bytes) vs. $\log_2 u$ for different methods: Send−V, H−WTok, Send−Sketch, Improved−S, TwoLevel−S.](image-url)
Experiments: Vary $u$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Experiments: Vary $\beta$

![Graph showing communication (bytes) vs split size (MB)](image-url)

- Send-V
- H-WTopk
- Send-Sketch
- Improved-S
- TwoLevel-S

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Experiments: Vary $\beta$

<table>
<thead>
<tr>
<th>Split Size (MB)</th>
<th>Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td></td>
</tr>
</tbody>
</table>

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Experiments: Vary $\alpha$

Communication (Bytes)

$\alpha = 0.8$

$\alpha = 1.1$

$\alpha = 1.4$

Send−V, H−WTopk, Send−Sketch, Improved−S, TwoLevel−S
Experiments: Vary $\alpha$

![Graph showing time in seconds for different $\alpha$ values]

- $\alpha = 0.8$
- $\alpha = 1.1$
- $\alpha = 1.4$

#### Graph Details:
- **X-axis:** $\alpha$ values (0.8, 1.1, 1.4)
- **Y-axis:** Time (Seconds)
- **Legend:**
  - Send-V
  - H-WTopk
  - Send-Sketch
  - Improved-S
  - TwoLevel-S

**Legend Interpretation:**
- Black bars: Send-V
- Dark grey bars: H-WTopk
- Light grey bars: Send-Sketch
- Dark grey with medium grey: Improved-S
- Light grey with medium grey: TwoLevel-S
Experiments: Vary $B$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Experiments: WorldCup Dataset

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Experiments: *WorldCup* Dataset

![Bar chart showing comparison of different methods for wavelet histograms]

- **Send-V**
- **H-WTopk**
- **Send-Sketch**
- **Improved-S**
- **TwoLevel-S**

Jeffrey Jestes, Ke Yi, Feifei Li

*Building Wavelet Histograms on Large Data in MapReduce*
Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
We study the problem of efficiently computing wavelet histograms in MapReduce clusters.

- We present both exact and approximate algorithms.

TwoLevel-S is especially easy to implement and ideal in practice.

For 200GB of data with $\log_2 u = 29$ it takes 10 minutes with only 2MB of communication!

Our work is just the tip of the iceberg for data summarization techniques in MapReduce. Many others remain including:

- other histograms including the V-optimal histogram,
- sketches and synopsis,
- geometric summaries ($\varepsilon$-approximations and coresets),
- graph summaries (distance oracles).
We study the problem of efficiently computing wavelet histograms in MapReduce clusters.

- We present both exact and approximate algorithms.
- *TwoLevel-S* is especially easy to implement and ideal in practice.
Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
  - We present both exact and approximate algorithms.
  - TwoLevel-S is especially easy to implement and ideal in practice.
    - For 200GB of data with \( \log_2 u = 29 \) it takes 10 minutes with only 2MB of communication!

Our work is just the tip of the iceberg for data summarization techniques in MapReduce. Many others remain including:

- other histograms including the V-optimal histogram,
- sketches and synopsis,
- geometric summaries (\( \varepsilon \)-approximations and coresets),
- graph summaries (distance oracles).
Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
  - We present both exact and approximate algorithms.
  - TwoLevel-S is especially easy to implement and ideal in practice.
    - For 200GB of data with $\log_2 u = 29$ it takes 10 minutes with only 2MB of communication!
- Our work is just the tip of the iceberg for data summarization techniques in MapReduce.
Conclusions

- We study the problem of efficiently computing wavelet histograms in MapReduce clusters.
  - We present both exact and approximate algorithms.
  - TwoLevel-S is especially easy to implement and ideal in practice.
    - For 200GB of data with $\log_2 u = 29$ it takes 10 minutes with only 2MB of communication!
- Our work is just the tip of the iceberg for data summarization techniques in MapReduce.
- Many others remain including:
We study the problem of efficiently computing wavelet histograms in MapReduce clusters.

- We present both exact and approximate algorithms.
- TwoLevel-S is especially easy to implement and ideal in practice.
  - For 200GB of data with $\log_2 u = 29$ it takes 10 minutes with only 2MB of communication!

Our work is just the tip of the iceberg for data summarization techniques in MapReduce.

Many others remain including:
- other histograms including the V-optimal histogram,
- sketches and synopsis,
- geometric summaries ($\varepsilon$-approximations and coresets),
- graph summaries (distance oracles).
Thank You

Q and A
We may also compute $w_i$ with the wavelet basis vectors $\psi_i$. 
We may also compute \( w_i \) with the wavelet basis vectors \( \psi_i \).

\[
\begin{array}{c|ccccccc}
\ell & 0 & 1 & 2 & 3 \\
\hline
v(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

\[
\begin{array}{c|c}
\ell = 0 & \ell = 1 & \ell = 2 & \ell = 3 \\
\hline
w_1 & 6.8 & \text{total average} & \psi_{a_1} \\
w_2 & 0.3 & 6.8 & \psi_{a_2} \\
w_3 & 2.5 & 6.5 & \psi_{a_3} \\
w_5 & 1 & 4 & \psi_{a_4} \\
w_6 & -1 & 9 & \psi_{a_5} \\
w_4 & 5 & 7 & \psi_{a_6} \\
w_7 & 0 & 2 & \psi_{a_7} \\
w_8 & 2 & 12 & \psi_{a_8} \\
\end{array}
\]

\[
w_6 = \langle v, \psi_{6} \rangle
\]
Introduction: Histograms

We may also compute $w_i$ with the wavelet basis vectors $\psi_i$.

\[
\begin{array}{cccccccc}
  x & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
  v(x) & 3 & 5 & 10 & 8 & 2 & 2 & 10 & 14 \\
\end{array}
\]

\[
w_1 = 6.8 \quad \text{total average}
\]

\[
w_2 = 0.3 \quad a_1
\]

\[
w_3 = 2.5 \quad 6.5 \quad a_2
\]

\[
w_4 = 5 \quad 7 \quad a_3
\]

\[
w_5 = 1 \quad 4 \quad a_4
\]

\[
w_6 = -1 \quad 9 \quad a_5
\]

\[
w_7 = 0 \quad 2 \quad a_6
\]

\[
w_8 = 2 \quad 12 \quad a_7
\]

\[
\ell = 0
\]

\[
\ell = 1
\]

\[
\ell = 2
\]

\[
\ell = 3
\]

\[
w_6 = \langle v, \psi_6 \rangle
\]

\[
w_6 = \langle v, \psi_6 \rangle = \langle [0 \ 0 \ v(3) \ 0 \ 0 \ 0 \ 0 \ 0], \psi_6 \rangle + \langle [0 \ 0 \ 0 \ v(4) \ 0 \ 0 \ 0 \ 0], \psi_6 \rangle
\]
We may also compute $w_i$ with the wavelet basis vectors $\psi_i$.

$$w_3 = \langle v, \psi_3 \rangle$$
We may also compute $w_i$ with the wavelet basis vectors $\psi_i$.

$$w_2 = \langle v, \psi_2 \rangle$$
We may also compute $w_i$ with the wavelet basis vectors $\psi_i$. 

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x)$</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

$w_1 = \langle v, \psi_1 \rangle$
The JobTracker assigns an InputSplit to a TaskTracker, a MapRunner task runs on the TaskTracker to process the split.
The MapRunner acquires a RecordReader from the InputFormat for the file to view the InputSplit as a stream of records, \((k_1, v_1)\).
The MapRunner invokes the user specified Mapper for each \((k_1, v_1)\), the Mapper emits \((k_2, v_2)\) and stores in an in-memory buffer.
When the buffer fills, the optional *Combiner* is executed over \((k_2, list(v_2))\), and a \((k_2, v_2)\) is dumped to a partition on disk.
The JobTracker assigns Reducers to TaskTrackers for each partition, each reducer first copies on \((k_2, v_2)\) and then sorts on \(k_2\).
The sorting output \((k_2, \text{list}(v_2))\) is processed one \(k_2\) at a time and reduced, the reduced output \((k_3, v_3)\) is written to reducer output \(o_i\).
Exact Top-k Wavelet Coefficients: Hadoop Phase 1

JobTracker

split 1
split 2
split 3
(x, null)
(x, null)
(x, null)

Mapper

Mapper

Mapper
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 1

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 1

JobTracker

split 1
split 2
split 3

(x, null)
(x, null)
(x, null)

Mapper
Mapper
Mapper

close()
close()
close()

(x, v_1(x))
(x, v_2(x))
(x, v_3(x))

Streaming Compute \(w_i\)
Streaming Compute \(w_i\)
Streaming Compute \(w_i\)
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 1

JobTracker

split 1
split 2
split 3

(x, null)

Mapper

(close())

(x, v_1(x))

Streaming Compute \(w_i\)

Priority Queue: Largest \(w_{i,j}\)

HDFS: \(w_{i,j}\) Streamed to disk

Priority Queue: Smallest \(w_{i,j}\)

Mapper

(close())

(x, v_2(x))

Streaming Compute \(w_i\)

Priority Queue: Largest \(w_{i,j}\)

HDFS: \(w_{i,j}\) Streamed to disk

Priority Queue: Smallest \(w_{i,j}\)

Mapper

(close())

(x, v_3(x))

Streaming Compute \(w_i\)

Priority Queue: Largest \(w_{i,j}\)

HDFS: \(w_{i,j}\) Streamed to disk

Priority Queue: Smallest \(w_{i,j}\)
Exact Top-k Wavelet Coefficients: Hadoop Phase 1

JobTracker

Mapper

Mapper

Mapper

(x, null)

(x, null)

(x, null)

split 1
split 2
split 3

(x, v_1(x))

(x, v_2(x))

(x, v_3(x))

Streaming Compute w_i

Streaming Compute w_i

Streaming Compute w_i

split 1

split 2

split 3

i

w_{i,1}

5

20

2

7

1

6

4

-2

6

-15

3

-30

i

w_{i,2}

5

12

4

7

1

2

2

-5

3

-14

6

-20

i

w_{i,3}

1

10

3

6

4

5

2

-3

5

-6

6

-10
Exact Top-\( k \) Wavelet Coefficients: Hadoop Phase 1

JobTracker

split 1

(x, null)

Mapper

close()

(x, \( v_1(x) \))

Streaming

Compute \( w_i \)

split 1

i

\( w_{i,1} \)

5

20

2

7

1

6

4

-2

6

-15

3

-30

split 2

(x, null)

Mapper

close()

(x, \( v_2(x) \))

Streaming

Compute \( w_i \)

split 2

i

\( w_{i,2} \)

5

12

4

7

1

2

2

-5

3

-14

0

-20

split 3

(x, null)

Mapper

close()

(x, \( v_3(x) \))

Streaming

Compute \( w_i \)

split 3

i

\( w_{i,3} \)

1

10

3

6

4

5

2

-3

5

-6

0

-10

p_1

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 1

JobTracker

Mapper

Mapper

Mapper

Reducer

\begin{align*}
R & = i \left( \hat{w}_i F_x \tau^+ (w_i) \tau^- (w_i) \tau (w_i) \right) \\
\begin{array}{cccc}
\text{split 1} & \text{split 2} & \text{split 3} & (x, \text{null}) \\
\hline
i & (\text{split}) & j & w_{i,j} \\
1 & 3 & 10 & 001 & 42 & -40 & 0 \\
3 & 1 & -30 & 100 & -8 & -60 & 8 \\
5 & 1 & 20 & 110 & 42 & 22 & 22 \\
5 & 2 & 12 & 111 & -10 & -60 & 10 \\
6 & 2 & -20 & & & & \\
6 & 3 & -10 & & & & \\
\end{array}
\end{align*}
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1
Exact Top-k Wavelet Coefficients: Hadoop Phase 1

JobTracker

Mapper

Mapper

Mapper

Reducer

\[
\begin{array}{c|c|c}
 R & i & w_{ij} \\
\hline
 1 & 3 & 10 \\
 3 & 1 & -30 \\
 5 & 1 & 20 \\
 5 & 2 & 12 \\
 6 & 2 & -20 \\
 6 & 3 & -10 \\
\end{array}
\]

Close Phase

\[
\begin{array}{c|c|c|c|c}
 R & i & \bar{w}_i & F_x & \tau^+(w_i) & \tau^-(w_i) & \tau(w_i) \\
\hline
 1 & 10 & 001 & 42 & -40 & 0 \\
 3 & -30 & 100 & -8 & -60 & 8 \\
 5 & 32 & 110 & 42 & 22 & 22 \\
 6 & -30 & 011 & -10 & -60 & 10 \\
\end{array}
\]
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

JobTracker

Mapper

Mapper

Mapper

Reducer

Close Phase

\[ R \]

<table>
<thead>
<tr>
<th>$i$</th>
<th>(split) $j$</th>
<th>$w_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-30</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>-20</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>-10</td>
</tr>
</tbody>
</table>

\[ \begin{align*}
R & \quad i \quad \bar{w}_i \quad F_x \quad \tau^+(w_i) \quad \tau^-(w_i) \quad \tau(w_i) \\
1 & \quad 10 \quad 001 \quad 42 \quad -40 \quad 0 \\
3 & \quad -30 \quad 100 \quad -8 \quad -60 \quad 8 \\
5 & \quad 32 \quad 110 \quad 42 \quad 22 \quad 22 \\
6 & \quad -30 \quad 011 \quad -10 \quad -60 \quad 10 \\
\end{align*} \]
Exact Top-k Wavelet Coefficients: Hadoop Phase 1

JobTracker

split 1
split 2
split 3

(x, null)

Mapper

(i, (split 1, \(w_{i,1}\)))

P_1

\(\hat{R}\)

<table>
<thead>
<tr>
<th>(i)</th>
<th>(split)</th>
<th>(j)</th>
<th>(w_{i,j})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-30</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>-20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>-10</td>
<td></td>
</tr>
</tbody>
</table>

Reducer

Close Phase

\(T_1 = 22, \; T_1/m = 22/3\)

\(\bar{w}_i\)

\(F_x\)

\(\tau^+(w_i)\)

\(\tau^-(w_i)\)

\(\tau(w_i)\)

<table>
<thead>
<tr>
<th>(i)</th>
<th>(\bar{w}_i)</th>
<th>(F_x)</th>
<th>(\tau^+(w_i))</th>
<th>(\tau^-(w_i))</th>
<th>(\tau(w_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>42</td>
<td>-40</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>-30</td>
<td>100</td>
<td>-8</td>
<td>-60</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>42</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>-30</td>
<td>011</td>
<td>-10</td>
<td>-60</td>
<td>10</td>
</tr>
</tbody>
</table>
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

JobTracker

Mapper

(i, (split 1, $w_{i,1}$))

$i$ $w_{i,1}$
5 20
3 -30

Mapper

(i, (split 2, $w_{i,2}$))

$i$ $w_{i,2}$
5 12
6 -20

Mapper

(i, (split 3, $w_{i,3}$))

$i$ $w_{i,3}$
1 10
6 -10

Reducer

$R$

<table>
<thead>
<tr>
<th>$i$</th>
<th>(split)</th>
<th>$w_{i,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-30</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>-20</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>-10</td>
</tr>
</tbody>
</table>

Close Phase

$T_1 = 22, \frac{T_1}{m} = \frac{22}{3}$

Coordinator
State

$R$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\bar{w}_i$</th>
<th>$F_x$</th>
<th>$\tau^+(w_i)$</th>
<th>$\tau^-(w_i)$</th>
<th>$\tau(w_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>42</td>
<td>-40</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>-30</td>
<td>100</td>
<td>-8</td>
<td>-60</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>42</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>-30</td>
<td>011</td>
<td>-10</td>
<td>-60</td>
<td>10</td>
</tr>
</tbody>
</table>
Exact Top-k Wavelet Coefficients: Hadoop Phase 1

Close Phase

\[ T_1 = 22, \quad T_1 / m = 22/3 \]

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

### Diagram Description

- **JobTracker**:
  - Process input
  - Sends input to Mappers

- **Mappers**:
  - Process input separately
  - Emit (key, value) pairs
  - Mappers produce:
    - $(x, null)$
    - $(i, w_{i,1}, 5, 20)$
    - $(i, w_{i,2}, 5, 12)$
    - $(i, w_{i,3}, 6, -20)$

- **Reducer**:
  - Aggregates results from Mappers
  - Performs calculations:
    - $R_i \hat{w}_i F_x \tau^+(w_i) \tau^-(w_i) \tau(w_i)$
  - Closes phase

- **HDFS**:
  - Stores intermediate results

- **Coordinator State**:
  - Maintains global state
  - $T_1 = 22, T_1/m = 22/3$

### Table

<table>
<thead>
<tr>
<th>$i$</th>
<th>$(split)$</th>
<th>$w_{i,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-30</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>-20</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>-10</td>
</tr>
</tbody>
</table>

### Formulas

- $R_i \hat{w}_i F_x \tau^+(w_i) \tau^-(w_i) \tau(w_i)$

---

**Jeffrey Jestes, Ke Yi, Feifei Li**

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

JobTracker

split 1
split 2
split 3

$T_1/m = 22/3$

Job Configuration

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

JobTracker

split 1
split 2
split 3

$T_1/m = 22/3$

Job Configuration

bypass
bypass
bypass

split 1: saved $w_{i,j}$
split 2: saved $w_{i,j}$
split 3: saved $w_{i,j}$
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[ T_1/m = 22/3 \]

JobTracker

 Mapper

 Mapper

 Mapper

\begin{align*}
\text{split 1} & : i \quad w_{i,1} \\
5 & : 20 \\
2 & : 7 \\
1 & : 6 \\
4 & : -2 \\
6 & : -15 \\
3 & : 30 \\
\end{align*}

\begin{align*}
\text{split 2} & : i \quad w_{i,2} \\
5 & : 12 \\
4 & : 7 \\
1 & : 2 \\
2 & : -5 \\
3 & : -14 \\
6 & : -20 \\
\end{align*}

\begin{align*}
\text{split 3} & : i \quad w_{i,3} \\
1 & : 10 \\
3 & : 6 \\
4 & : 5 \\
2 & : -3 \\
5 & : -6 \\
6 & : -10 \\
\end{align*}

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\(T_1/m = 22/3\)

JobConfiguration

split 1

\(i \quad w_{i,1}\)

5 20
2 7
1 6
4 -2
6 -15
3 -30

split 2

\(i \quad w_{i,2}\)

5 12
4 7
1 2
2 -5
3 -14
6 -20

split 3

\(i \quad w_{i,3}\)

1 10
3 6
4 5
2 -3
5 -6
6 -10

Mapper

Mapper

Mapper
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

Given data splits and job configuration, the process involves:

1. **Mapper** stages that process the data:
   - Split 1: Coefficients \(i, w_{i1}, 5, 7, 1, 6, 4, -2, 6, -15, 3, 30\)
   - Split 2: Coefficients \(i, w_{i2}, 5, 12, 4, 7, 1, 2, 2, -5, 3, -14, 6, -20\)
   - Split 3: Coefficients \(i, w_{i3}, 1, 10, 3, 6, 4, 5, 2, -3, 5, -6, 6, -10\)

2. **Reducer** stages that aggregate the data:
   - Reducer 1: \((6, \text{split 1}, -15)\)
   - Reducer 2: \((3, \text{split 2}, -14)\)

3. **State update** phase that finalizes the processing:

The JobTracker coordinates the process, with bypass options for each split.
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

split 1
\[
\begin{array}{|c|c|}
\hline
i & w_{i,1} \\
\hline
5 & 20 \\
2 & 7 \\
1 & 6 \\
4 & -2 \\
6 & -15 \\
3 & 30 \\
\hline
\end{array}
\]

split 2
\[
\begin{array}{|c|c|}
\hline
i & w_{i,2} \\
\hline
5 & 12 \\
4 & 7 \\
1 & 2 \\
2 & -5 \\
3 & -14 \\
6 & 20 \\
\hline
\end{array}
\]

split 3
\[
\begin{array}{|c|c|}
\hline
i & w_{i,3} \\
\hline
1 & 10 \\
3 & 6 \\
4 & 5 \\
2 & -3 \\
5 & -6 \\
6 & -10 \\
\hline
\end{array}
\]

JobTracker

Mapper

Reducer

\[ \text{update} \rightarrow \text{reduce phase} \]

\[ R \]

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
i & w_i & F_x & \tau^+(w_i) & \tau^-(w_i) & \tau(w_i) \\
\hline
1 & 10 & 001 & 24.6 & -4.6 & 0 & 24.6 \\
3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\
6 & -45 & 111 & -45 & -45 & 45 & 45 \\
\hline
\end{array}
\]
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[ R \]

<table>
<thead>
<tr>
<th>(i)</th>
<th>(w_i)</th>
<th>(F_x)</th>
<th>(\tau(x))</th>
<th>(\tau^{-}(w_i))</th>
<th>(\tau^{-}(w_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>24.6</td>
<td>-4.6</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>-44</td>
<td>110</td>
<td>-36.6</td>
<td>-51.3</td>
<td>36.6</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>39.3</td>
<td>24.6</td>
<td>39.3</td>
</tr>
<tr>
<td>6</td>
<td>-45</td>
<td>111</td>
<td>-45</td>
<td>-45</td>
<td>45</td>
</tr>
</tbody>
</table>

\[ T_2 = 45 \]
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

JobTracker

\[ \frac{T_1}{m} = \frac{22}{3} \]

Job Configuration

Mapper

Mapper

Mapper

Mapper

Mapper

Mapper

Reducer

Coordinator State

close phase

\( T_2 = 45 \)

\[ R \]

\[
\begin{array}{cccccc}
  i & w_i & F_i & \tau^+(w_i) & \tau^-(w_i) & \tau(w_i) \\
  1 & 10 & 001 & -24.6 & 4.6 & 0 & 24.6 \\
  3 & -44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
  5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\
  6 & -45 & 111 & -45 & -45 & 45 & 45 \\
\end{array}
\]
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

JobTracker

 splitter 1
 splitter 2
 splitter 3

Mapper

Mapper

Mapper

Reducer

Coordinator State

Coordinator State

Table: $R$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$w_i$</th>
<th>$F_x$</th>
<th>$\tau^+(w_i)$</th>
<th>$\tau^-(w_i)$</th>
<th>$\tau(w_i)$</th>
<th>$\tau'(w_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>-24.6</td>
<td>-4.6</td>
<td>0</td>
<td>24.6</td>
</tr>
<tr>
<td>3</td>
<td>-44</td>
<td>110</td>
<td>-36.6</td>
<td>-51.3</td>
<td>36.6</td>
<td>51.3</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>-39.3</td>
<td>24.6</td>
<td>24.6</td>
<td>39.3</td>
</tr>
<tr>
<td>6</td>
<td>-45</td>
<td>111</td>
<td>-45</td>
<td>-45</td>
<td>45</td>
<td>45</td>
</tr>
</tbody>
</table>

$T_1/m = 22/3$

$T_2 = 45$
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

JobTracker

\(T_1/m = 22/3\)

Job Configuration

Mapper

\(i\) \(w_i,1\)

5 20
2 7
1 6
4 -2
6 -15
3 30

split 1

Mapper

\(i\) \(w_i,2\)

5 12
4 7
1 2
2 -5
3 -14
6 20

split 2

Mapper

\(i\) \(w_i,3\)

1 10
3 6
4 5
2 -3
5 -6
6 10

split 3

Reducer

\(R\)

\(i\) \(w_i\) \(F_x\) \(\tau^+ (w_i)\) \(\tau^- (w_i)\) \(\tau^+ (w_i)\) \(\tau^- (w_i)\)

1 10 001 -24.6 -4.6 0 24.6
3 -44 110 -36.6 -51.3 36.6 51.3
5 -32 110 -39.3 24.6 24.6 39.3
6 -45 111 -45 -45 45 45

Coordinate State

HDFS or Local

\(T_2 = 45\)

close phase

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

$T_1/m = 22/3$

JobTracker

Mapper

Mapper

Mapper

Reducer

Coordinator State

close phase

$T_2 = 45$

$R$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$w_i$</th>
<th>$F_x$</th>
<th>$\tau^+(w_i)$</th>
<th>$\tau^-(w_i)$</th>
<th>$\tau(w_i)$</th>
<th>$\tau'(w_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>001</td>
<td>-24.6</td>
<td>4.6</td>
<td>0</td>
<td>24.6</td>
</tr>
<tr>
<td>3</td>
<td>-44</td>
<td>110</td>
<td>-36.6</td>
<td>-51.3</td>
<td>36.6</td>
<td>51.3</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>110</td>
<td>39.3</td>
<td>24.6</td>
<td>24.6</td>
<td>39.3</td>
</tr>
<tr>
<td>6</td>
<td>-45</td>
<td>111</td>
<td>-45</td>
<td>-45</td>
<td>45</td>
<td>45</td>
</tr>
</tbody>
</table>

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

JobTracker

split 1
\(i \ w_{i,1}\)
5 20
2 7
1 6
4 -2
6 -15
3 30

bypass

split 2
\(i \ w_{i,2}\)
5 12
4 7
1 2
2 -5
3 -14
6 20

bypass

split 3
\(i \ w_{i,3}\)
1 10
3 6
4 5
2 -3
5 -6
6 -10

bypass

Mapper

(6, (split 1, -15))

Mapper

(3, (split 2, -14))

Reducer

Coordinator State

\(w_i \in \mathbb{R}\)

missing a \(w_{i,j}\)

close phase

\(T_1/m = 22/3\)

Job Configuration

\(T_2 = 45\)

\(w_i \in \mathbb{R}\)

\(\text{missing a } w_{i,j}\)

\(R\)

\(\begin{array}{cccccc}
  i & w_i & F_x & \tau^+(w_i) & \tau^-(w_i) & \tau(w_i) \\
  1 & 10 & 001 & 24.6 & 4.6 & 0 & 24.6 \\
  3 & 44 & 110 & -36.6 & -51.3 & 36.6 & 51.3 \\
  5 & 32 & 110 & 39.3 & 24.6 & 24.6 & 39.3 \\
  6 & -45 & 111 & -45 & -45 & 45 & 45 \\
\end{array}\)
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

JobTracker

split 1

5 20
2 7
1 6
4 -2
6 -15
3 30

split 2

5 12
4 7
1 2
2 -5
3 -14
6 -20

split 3

i $w_{i,1}$

1 10
3 6
4 5
2 -3
5 -6
6 -10

split 1

i $w_{i,2}$

6 -15
3 -14

Mapper

(6(split 1,-15))

Mapper

(3,(split 2,-14))

Reducer

HDFS

Coordinator State

$w_{i,j} \in R$

missing a $w_{i,j}$

$T_1/m = 22/3$

Job Configuration

$T_2 = 45$

Job Configuration

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

$\frac{T_1}{m} = \frac{22}{3}$

Job Configuration

Distributed Cache

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

$T_1/m = 22/3$

Job Configuration

Distributed Cache

split 1:
saved $w_{i,j}$

split 2:
saved $w_{i,j}$

split 3:
saved $w_{i,j}$
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

JobTracker

$T_1/m = \frac{22}{3}$

Job Configuration

Distributed Cache

Mapper

Mapper

Mapper

split 1

$\begin{array}{c|c}
i & w_{i,1} \\
\hline
5 & 20 \\
2 & 7 \\
1 & 6 \\
4 & -2 \\
6 & -15 \\
3 & -30 \\
\end{array}$

split 2

$\begin{array}{c|c}
i & w_{i,2} \\
\hline
5 & 12 \\
4 & 7 \\
1 & 2 \\
2 & -5 \\
3 & -14 \\
6 & -20 \\
\end{array}$

split 3

$\begin{array}{c|c}
i & w_{i,3} \\
\hline
1 & 10 \\
3 & 6 \\
4 & 5 \\
2 & -3 \\
5 & -6 \\
6 & 10 \\
\end{array}$
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[ \frac{T_1}{m} = \frac{22}{3} \]

*Job Configuration*

\[
\begin{array}{c|c}
\text{split 1} & \text{i} & w_{i,1} \\
\hline
5 & 20 \\
2 & 7 \\
1 & 6 \\
4 & -2 \\
6 & -15 \\
3 & -30 \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{split 2} & \text{i} & w_{i,2} \\
\hline
5 & 12 \\
4 & 7 \\
1 & 2 \\
2 & -5 \\
3 & 14 \\
6 & -20 \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{split 3} & \text{i} & w_{i,3} \\
\hline
1 & 10 \\
3 & 6 \\
4 & 5 \\
2 & -3 \\
5 & -6 \\
6 & 10 \\
\end{array}
\]
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[
\begin{array}{c|c}
\text{split} 1 & \text{split} 2 & \text{split} 3 \\
\hline
\begin{array}{c|c}
i & w_{i,1} \\
5 & 20 \\
2 & 7 \\
1 & 6 \\
4 & -2 \\
6 & -15 \\
3 & -30 \\
\end{array} & \\
\begin{array}{c|c}
i & w_{i,2} \\
5 & 12 \\
4 & 7 \\
1 & 2 \\
2 & -5 \\
3 & -14 \\
6 & -20 \\
\end{array} & \\
\begin{array}{c|c}
i & w_{i,3} \\
3 & 10 \\
1 & 6 \\
4 & 5 \\
2 & -3 \\
5 & -6 \\
6 & -10 \\
\end{array}
\end{array}
\]

\(T_1/m = 22/3\)

Job Configuration

Distributed Cache

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

\[ T_1/m = \frac{22}{3} \]

Job Configuration

Distributed Cache

JobTracker

Mapper

Mapper

Mapper

Reducer

(3, (site 3, 6))

\begin{align*}
\text{split 1} & \quad i \quad w_{i,1} \\
5 & \quad 20 \\
2 & \quad 7 \\
1 & \quad 6 \\
4 & \quad -2 \\
6 & \quad -15 \\
3 & \quad -30 \\
\end{align*}

\begin{align*}
\text{split 2} & \quad i \quad w_{i,2} \\
5 & \quad 12 \\
4 & \quad 7 \\
1 & \quad 2 \\
2 & \quad -5 \\
3 & \quad -14 \\
6 & \quad -20 \\
\end{align*}

\begin{align*}
\text{split 3} & \quad i \quad w_{i,3} \\
1 & \quad 10 \\
3 & \quad 6 \\
4 & \quad 5 \\
2 & \quad -3 \\
5 & \quad -6 \\
6 & \quad -10 \\
\end{align*}
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

$T_1/m = \frac{22}{3}$

Jeffrey Jestes, Ke Yi, Feifei Li
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[ T_1/m = \frac{22}{3} \]

Job Configuration

Distributed Cache

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

JobTracker

split 1
\[
\begin{array}{c|c}
\hline
i & w_{i,1} \\
\hline
5 & 20 \\
2 & 7 \\
1 & 6 \\
4 & -2 \\
6 & -15 \\
3 & -30 \\
\hline
\end{array}
\]

split 2
\[
\begin{array}{c|c}
\hline
i & w_{i,2} \\
\hline
5 & 12 \\
4 & 7 \\
1 & 2 \\
2 & -5 \\
3 & -14 \\
6 & -20 \\
\hline
\end{array}
\]

split 3
\[
\begin{array}{c|c}
\hline
i & w_{i,3} \\
\hline
1 & 10 \\
3 & 6 \\
4 & 5 \\
2 & -3 \\
5 & -6 \\
6 & 10 \\
\hline
\end{array}
\]

Mapper

Reducer
\((3,(\text{site} 3,6))\)

update

\(R\)
\[
\begin{array}{c|cc}
\hline
i & w_i & F_x \\
\hline
3 & -38 & 111 \\
6 & -45 & 111 \\
\hline
\end{array}
\]

Job Configuration

\(T_1/m = 22/3\)

Distributed Cache

Jeffrey Jestes, Ke Yi, Feifei Li
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

JobTracker

split 1

bypass

split 2

bypass

split 3

bypass

$T_1/m = 22/3$

Job Configuration

Distributed Cache

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

![Diagram showing the process of exact top-$k$ wavelet coefficients using Hadoop Phase 3. The diagram includes JobTracker, Mapper, and Reducer nodes, with nodes for split 1, split 2, and split 3. The Job Tracker is connected to the Mapper nodes, which are bypassed to the Reducer node. The diagram also includes a table showing the wavelet coefficients for each split and sites, with site 3,6 highlighted. The formula $T_1/m = 22/3$ is shown, and the distributed cache is indicated.]

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

JobTracker

**split 1**

<table>
<thead>
<tr>
<th>i</th>
<th>(w_i,1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>-2</td>
</tr>
<tr>
<td>6</td>
<td>-15</td>
</tr>
<tr>
<td>3</td>
<td>-30</td>
</tr>
</tbody>
</table>

**split 2**

<table>
<thead>
<tr>
<th>i</th>
<th>(w_i,2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>-5</td>
</tr>
<tr>
<td>3</td>
<td>-14</td>
</tr>
<tr>
<td>6</td>
<td>-20</td>
</tr>
</tbody>
</table>

**split 3**

<table>
<thead>
<tr>
<th>i</th>
<th>(w_i,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>5</td>
<td>-6</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

Reducer

\[ R = \begin{pmatrix} i & w_i & F_x \\ 3 & -38 & 111 \\ 6 & 45 & 111 \end{pmatrix} \]

\[ w_6 = -45 \]

Job Configuration

\( T_1/m = 22/3 \)

Distributed Cache

Results
Outline

1 Introduction and Motivation
   - Histograms
   - MapReduce and Hadoop

2 Exact Top-$k$ Wavelet Coefficients
   - Naive Solution
   - Hadoop Wavelet Top-$k$: Our Efficient Exact Solution

3 Approximate Top-$k$ Wavelet Coefficients
   - Linearly Combinable Sketch Method
   - Our First Sampling Based Approach
   - An Improved Sampling Approach
   - Two-Level Sampling

4 Experiments

5 Conclusions
   - Hadoop Wavelet Top-$k$ in Hadoop
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

$k = 1$

JobTracker

split 1

split 2

split 3

(x, null)

Mapper

Mapper

Mapper
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

JobTracker

\[ k = 1 \]

split 1

\((x, \text{null})\)

Mapper

\((x, 1)\)

In-Memory Map

split 2

\((x, \text{null})\)

Mapper

\((x, 1)\)

In-Memory Map

split 3

\((x, \text{null})\)

Mapper

\((x, 1)\)

In-Memory Map
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

$$k = 1$$

JobTracker

split 1

split 2

split 3

$$(x, \text{null})$$

$$(x, \text{null})$$

$$(x, \text{null})$$

Mapper

close()

In-Memory Map

Mapper

close()

In-Memory Map

Mapper

close()

In-Memory Map

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

JobTracker

\[ k = 1 \]

split 1
split 2
split 3

(x, null)

(x, null)

(x, null)

Mapper

Mapper

Mapper

close()

close()

close()

Streaming Compute \( w_1 \)

Streaming Compute \( w_2 \)

Streaming Compute \( w_3 \)

(x, v_1(x))

(x, v_2(x))

(x, v_3(x))

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

JobTracker

$k = 1$

split 1
split 2
split 3

(x, null) -> Mapper -> close() -> Streaming Compute $w_i$

Priority Queue: Largest $w_{i,j}$

HDFS: streamed to disk

Priority Queue: Smallest $w_{i,j}$

HDFS: streamed to disk

Priority Queue: Largest $w_{i,j}$

HDFS: streamed to disk

Priority Queue: Smallest $w_{i,j}$

Priority Queue: Largest $w_{i,j}$

HDFS: streamed to disk

Priority Queue: Smallest $w_{i,j}$

Priority Queue: Largest $w_{i,j}$

HDFS: streamed to disk

Priority Queue: Smallest $w_{i,j}$

HDFS: streamed to disk

Priority Queue: Largest $w_{i,j}$

HDFS: streamed to disk

Priority Queue: Smallest $w_{i,j}$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

JobTracker

$k = 1$

split 1

split 2

split 3

Mapper

Mapper

Mapper

close()

close()

close()

(x, null)

(x, null)

(x, null)

(x, $v_1(x)$)

(x, $v_2(x)$)

(x, $v_3(x)$)

Streaming Compute $w_i$

Streaming Compute $w_i$

Streaming Compute $w_i$

split 1

split 2

split 3

$w_{2,1}$

$w_{1,1}$

$w_{4,1}$

$w_{5,1}$

$w_{6,1}$

$w_{3,1}$

$w_{1,2}$

$w_{2,2}$

$w_{5,2}$

$w_{3,2}$

$w_{6,2}$

$w_{4,2}$

$w_{2,3}$

$w_{1,3}$

$w_{6,3}$

$w_{4,3}$

$w_{3,3}$

$w_{5,3}$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

$\textbf{k} = 1$

JobTracker

split 1

$(x, \text{null})$

$(x, \text{null})$

$(x, \text{null})$

Mapper

Mapper

Mapper

$$\text{close}() \xrightarrow{(x, v_1(x))} \text{Streaming Compute } w_i$$

$$\text{close}() \xrightarrow{(x, v_2(x))} \text{Streaming Compute } w_i$$

$$\text{close}() \xrightarrow{(x, v_3(x))} \text{Streaming Compute } w_i$$

split 1

$k$ largest

$w_{2,1}$

$w_{1,1}$

$w_{4,1}$

$w_{5,1}$

$w_{6,1}$

$k$ smallest

$p_1$

split 2

$k$ largest

$w_{1,2}$

$w_{2,2}$

$w_{5,2}$

$w_{3,2}$

$w_{6,2}$

$k$ smallest

$p_1$

split 3

$k$ largest

$w_{2,3}$

$w_{1,3}$

$w_{6,3}$

$w_{4,3}$

$w_{3,3}$

$k$ smallest

$p_1$
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

$\tilde{w}_1 - w_{2,1}$

$\tilde{w}_1 - w_{3,1}$

$\tilde{w}_2 - w_{1,2}$

$\tilde{w}_2 - w_{4,2}$

$\tilde{w}_3 - w_{2,3}$

$\tilde{w}_3 - w_{5,3}$

$\tilde{w}_4 - w_{1,3}$

$\tilde{w}_4 - w_{6,3}$

$\tilde{w}_5 - w_{2,4}$

$\tilde{w}_5 - w_{7,4}$

$\tilde{w}_6 - w_{3,5}$

$\tilde{w}_6 - w_{8,5}$

$\tilde{w}_7 - w_{4,6}$

$\tilde{w}_7 - w_{9,6}$

$\tilde{w}_8 - w_{5,7}$

$\tilde{w}_8 - w_{10,7}$
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

$\tilde{w}_1 w_{2,1}$
$\tilde{w}_2 w_{1,2}$
$\tilde{w}_3 w_{2,3}$

$\tilde{w}_1 w_{3,1}$
$\tilde{w}_2 w_{4,2}$
$\tilde{w}_3 w_{5,3}$

$(i, (\text{split 1, } w_{i,1}))$
$(i, (\text{split 2, } w_{i,2}))$
$(i, (\text{split 3, } w_{i,3}))$

Mapper
Mapper
Mapper
Reducer
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

$\tilde{w} + 1 w_2, 1$  
$\tilde{w} - 2 w_4, 2$  
$\tilde{w} + 3 w_2, 3$  
$\tilde{w} - 3 w_5, 3$  

$\hat{R} \hat{w}_i F_i \tau^+(w_i)$  
$\hat{R} \tau^-(w_i)$  
$\hat{R} \tau(w_i)$
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 1

$k = 1$

Reduce phase

\[
\begin{align*}
R &\equiv \frac{w_i}{F_i} \\
&= \frac{\tau^+(w_i)}{\tau^-(w_i)} \\
&= \tau(w_i)
\end{align*}
\]
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 1

\[ k = 1 \]

\[ \begin{align*}
\text{Close phase} \\
\text{Compute} \\
R & = \begin{bmatrix}
\tilde{w}_i \\
F_i \\
\tau^+(w_i) \\
\tau^-(w_i) \\
\tau(w_i)
\end{bmatrix}
\end{align*} \]
Exact Top-k Wavelet Coefficients: Hadoop Phase 1

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 1

\[ k = 1 \]

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]

Reducer

\[ \tilde{w}_1^+, w_{2,1} \]
\[ \tilde{w}_1^-, w_{3,1} \]
\[ \tilde{w}_2^+, w_{1,2} \]
\[ \tilde{w}_2^-, w_{4,2} \]
\[ \tilde{w}_3^+, w_{2,3} \]
\[ \tilde{w}_3^-, w_{5,3} \]

\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 1}, w_{1,1})) \]
\[ (i, (\text{split 2}, w_{2,2})) \]
\[ (i, (\text{split 3}, w_{3,3})) \]
JobTracker

\[ k = 1 \]

\[ \frac{T_1}{m} \]

Job Configuration

Split 1
Split 2
Split 3
Exact Top-

1 Wavelet Coefficients: Hadoop Phase 2

JobTracker

split 1

bypass

split 1:
saved \( w_{i,j} \)

split 2:
saved \( w_{i,j} \)

split 3:
saved \( w_{i,j} \)

\( T_{1/m} \)

Job Configuration

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 2

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Hadoop Phase 2

JobTracker

split 1

split 2

split 3

Job Configuration

|split 1| |split 2| |split 3|
|---|---|---|---|
|w2,1| w1,1| w4,1|
|w5,1| w6,1| w9,1|

|w2,2| w2,2| w6,2|
|w3,2| w5,2| w4,2|

|w2,3| w1,3| w6,3|
|w4,3| w3,3| w5,3|

|w1,| w1,| w1,|
|w2,| w2,| w2,|
|w3,| w3,| w3,|

|w4,| w4,| w4,|
|w5,| w5,| w5,|

|w6,| w6,| w6,|

|w9,| w9,| w9,|

Mapper

Reducer

(i, (split i, wi,j))

$|w_{i,j}| > \frac{T_1}{m}$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 2

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 2

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Hadoop Phase 2

split 1
split 2
split 3

JobTracker

bypass
Mapper
Mapper
Mapper
bypass
bypass

Reducer

Coordinator
State

Refine

$|w_{i,j}| > \frac{T_1}{m}$

$\hat{R} w_i$

$F_i$

$\tau^+(w_i)$

$\tau^-(w_i)$

$\tau(w_i)$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 2

JobTracker

split 1
\[ w_{2,1}, w_{1,1}, w_{4,1}, w_{5,1}, w_{6,1}, w_{3,1} \]

split 2
\[ w_{2,2}, w_{2,2}, w_{6,2}, w_{5,2}, w_{4,2} \]

split 3
\[ w_{2,3}, w_{1,3}, w_{6,3}, w_{4,3}, w_{3,3}, w_{5,3} \]

\(|w_{i,j}| > T_1/m\)

Mapper

(i, (split \(i\), \(w_{i,j}\)))

Reducer

Refine → Reduce phase

Coordinator State

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Hadoop Phase 2

|wi ,j| > T1/m

(i,(split i,wi,j))

R

\hat{w}_i

F_i

\tau^+(w_i)

\tau^-(w_i)

\tau(w_i)

Refine  Close phase

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 2

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 2

JobTracker

split 1

| $w_{2,1}$ |
| $w_{1,1}$ |
| $w_{4,1}$ |
| $w_{5,1}$ |
| $w_{6,1}$ |
| $w_{9,1}$ |

| $|w_{i,j}| > T_1/m$ |

split 2

| $w_{2,2}$ |
| $w_{2,2}$ |
| $w_{6,2}$ |
| $w_{3,2}$ |
| $w_{5,2}$ |
| $w_{4,2}$ |

| $|w_{i,j}| > T_1/m$ |

split 3

| $w_{2,3}$ |
| $w_{1,3}$ |
| $w_{6,3}$ |
| $w_{4,3}$ |
| $w_{3,3}$ |
| $w_{5,3}$ |

| $|w_{i,j}| > T_1/m$ |

Reducer

$(i, (\text{split } i, w_{i,j}))$

Mapper

Mapper

Mapper

Coordinator State

Close phase

- Reducer computes $\tau'(w_i)$ and prunes $w_i$ from $R$ with $\tau'(w_i) < T_2$

$R$

$w_i$

$F_i$

$\tau^+(w_i)$

$\tau(w_i)$

$k'th largest \tau(w_i)$

$T_2$
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 2

JobTracker

split 1
- $w_{2,1}$
- $w_{1,1}$
- $w_{4,1}$
- $w_{5,1}$
- $w_{6,1}$
- $w_{3,1}$

<table>
<thead>
<tr>
<th>$w_{i,j}$</th>
<th>$&gt; T_1/m$</th>
</tr>
</thead>
</table>

split 2
- $w_{2,2}$
- $w_{2,2}$
- $w_{6,2}$
- $w_{3,2}$
- $w_{5,2}$
- $w_{4,2}$

split 3
- $w_{2,3}$
- $w_{1,3}$
- $w_{6,3}$
- $w_{4,3}$
- $w_{3,3}$
- $w_{5,3}$

Mapper

$(i, (\text{split } i, w_{i,j}))$

Reducer

$|w_{i,j}| > T_1/m$

Mapper

Mapper

Mapper

$bypass$

$bypass$

$bypass$

Job Configuration

$T_1/m$

Coordinator State

Local or HDFS

$R$

$F_i$

$\tau^+(w_i)$

$\tau(w_i)$

$k$'th largest $\tau(w_i)$

$T_2$

Close phase

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 2

split 1
\[ w_{2,1}, w_{1,1}, w_{4,1}, w_{5,1}, w_{6,1}, w_{3,1} \]

\[ |w_{i,j}| > T_1/m \]

split 2
\[ w_{1,2}, w_{2,2}, w_{6,2}, w_{3,2}, w_{5,2}, w_{4,2} \]

\[ |w_{i,j}| > T_1/m \]

split 3
\[ w_{2,3}, w_{1,3}, w_{6,3}, w_{4,3}, w_{3,3}, w_{5,3} \]

\[ |w_{i,j}| > T_1/m \]

JobTracker

Mapper

Reducer

Mapper

Mapper

Mapper

bypass

bypass

bypass

bypass

Coordinator

State

HDFS

\[ R \]

\[ \tau^+(w_i), \tau^-(w_i), \tau(w_i) \]

\[ k'th \ largest \ \tau(w_i) \]

Close phase

i for \( w_i \in R \) missing a \( w_{i,j} \)

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

$T_1/m$

Job Configuration

$i \in R$

missing $w_{i,j}$

Distributed Cache

$k = 1$

split 1

split 2

split 3

JobTracker

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[ k = 1 \]

\[
\begin{align*}
T_1/m & \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\end{align*}
\]

Job Configuration

Distributed Cache

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[ k = 1 \]

\[ i \in R \text{ and } |w_{i,j}| \leq \frac{T_1}{m} \]

\[ i \in R \text{ and } |w_{i,j}| \leq \frac{T_1}{m} \]

\[ i \in R \text{ and } |w_{i,j}| \leq \frac{T_1}{m} \]

\( (i, (\text{split } i, w_{i,j})) \)

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[ k = 1 \]

\[ i \in R \text{ and } |w_{i,j}| \leq T_1/m \]

\[ (i, (\text{split } i, w_{i,j})) \]

\[ R \]

\[ \hat{w}_i \]

\[ F_i \]

\[ \tau^+(w_i) \]

\[ \tau(w_i) \]

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-k Wavelet Coefficients: Hadoop Phase 3

$\hat{w}_i F_i \tau^+ (w_i) \tau (w_i)$

$T_1/m$

$i \in R$ and $|w_{i,j}| \leq T_1/m$

$i \in R$ and $|w_{i,j}| \leq T_1/m$

$i \in R$ and missing $w_{i,j}$

$i \in R$ and $|w_{i,j}| \leq T_1/m$

$i \in R$ and $|w_{i,j}| \leq T_1/m$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Exact Top-\(k\) Wavelet Coefficients: Hadoop Phase 3

\[ i \in R \text{ and } |w_{i,j}| \leq T_1/m \]

\[ i \in R \text{ and } |w_{i,j}| \leq T_1/m \]

\[ i \in R \text{ and } |w_{i,j}| \leq T_1/m \]

\(k = 1\)

\(\text{split 1}
\text{split 2}
\text{split 3}
\)

JobTracker

\[ w_{2,1}
\text{split 1}
\text{split 2}
\text{split 3}
\]

Mapper

Reducer

\(R w_i\)

Update

\(\text{Exact } w_i\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)

\(i \in R \text{ and } |w_{i,j}| \leq T_1/m\)
Exact Top-$k$ Wavelet Coefficients: Hadoop Phase 3

$\text{split 1}$

$\text{split 2}$

$\text{split 3}$

$\text{Mapper}$

$\text{Reducer}$

$i \in R$ and $|w_{i,j}| \leq T_1/m$

$i \in R$ and $|w_{i,j}| \leq T_1/m$

$i \in R$ and $|w_{i,j}| \leq T_1/m$

$\text{Mapper}$

$\text{Mapper}$

$\text{Mapper}$

$\text{Results}$

$\text{Exact w}_i$

$\text{top-k (i, } |w_i|)$

$\text{Bypass}$

$\text{Bypass}$

$\text{Bypass}$

$k = 1$

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

$n_j =$ records in split $j$
$s_j =$ split $j$ sample frequency vector

JobTracker \[ \rightarrow \] MapRunner

split 1 \[ \rightarrow \] split 2 \[ \rightarrow \] split 3 \[ \rightarrow \] split 4

RandomizedRecordReader $j$ (RR$^j$) samples $n_j / \varepsilon^2 \times n$ records.

RR$^j$ randomly selects $n_j / \varepsilon^2 \times n$ offsets in split $j$.

RR$^j$ sorts the offsets in ascending order then seeks the record at each sampled offset.

Reducer uses $\hat{v}(x) = s(x) / p$, our unbiased estimator for $v(x)$.
Approximate Top-k Wavelet Coefficients: Basic Random Sampling

$n_j =$ records in split $j$

$s_j =$ split $j$ sample frequency vector

RandomizedRecordReader $j$ ($RR_j$) samples $n_j/\varepsilon^2 n$ records.
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

RandomizedRecordReader $j$ ($RR_j$) samples $n_j/\varepsilon^2 n$ records.

1. $RR_j$ randomly selects $n_j/\varepsilon^2 n$ offsets in split $j$. 

$n_j =$ records in split $j$

$s_j =$ split $j$ sample frequency vector
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

\[
\begin{align*}
n_j &= \text{records in split } j \\
s_j &= \text{split } j \text{ sample frequency vector}
\end{align*}
\]

---

1. RandomizedRecordReader \( j \) (\( RR_j \)) samples \( n_j/\varepsilon^2 n \) records.
   - \( RR_j \) randomly selects \( n_j/\varepsilon^2 n \) offsets in split \( j \).
   - \( RR_j \) sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-k Wavelet Coefficients: Basic Random Sampling

$\hat{v}(x) = s(x)/p$, our unbiased estimator for $v(x)$.

1. RandomizedRecordReader $j$ ($RR_j$) samples $n_j/\varepsilon^2 n$ records.
   1. $RR_j$ randomly selects $n_j/\varepsilon^2 n$ offsets in split $j$.
   2. $RR_j$ sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.

1. \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
2. \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.

\(n_j\) = records in split \(j\)
\(s_j\) = split \(j\) sample frequency vector

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

1. RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.
   - \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
   - \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

\[ n_j = \text{records in split } j \]
\[ s_j = \text{split } j \text{ sample frequency vector} \]

1. RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.
   - \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
   - \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.

1. \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
2. \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.

1. \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
2. \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

1. RandomizedRecordReader $j$ ($RR_j$) samples $n_j/\varepsilon^2 n$ records.
   1. $RR_j$ randomly selects $n_j/\varepsilon^2 n$ offsets in split $j$.
   2. $RR_j$ sorts the offsets in ascending order then seeks the record at each sampled offset.
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

RandomizedRecordReader $j$ ($RR_j$) samples $n_j/\varepsilon^2 n$ records.

1. $RR_j$ randomly selects $n_j/\varepsilon^2 n$ offsets in split $j$.
2. $RR_j$ sorts the offsets in ascending order then seeks the record at each sampled offset.

3. Reducer uses $\hat{v}(x) = s(x)/p$, our unbiased estimator for $v(x)$.
Approximate Top-$k$ Wavelet Coefficients: Basic Random Sampling

1. RandomizedRecordReader $j$ ($RR_j$) samples $n_j/\varepsilon^2 n$ records.
   1. $RR_j$ randomly selects $n_j/\varepsilon^2 n$ offsets in split $j$.
   2. $RR_j$ sorts the offsets in ascending order then seeks the record at each sampled offset.

2. Reducer uses $\hat{v}(x) = s(x)/p$, our unbiased estimator for $v(x)$.

\[ n_j = \text{records in split } j \]
\[ s_j = \text{split } j \text{ sample frequency vector} \]
**Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling**

- **RandomizedRecordReader** \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.
  - \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
  - \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.

- **Reducer** uses \(\hat{v}(x) = s(x)/p\), our unbiased estimator for \(v(x)\).
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

RandomizedRecordReader \(j\) (\(RR_j\)) samples \(n_j/\varepsilon^2 n\) records.

1. \(RR_j\) randomly selects \(n_j/\varepsilon^2 n\) offsets in split \(j\).
2. \(RR_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.

Reducer uses \(\hat{v}(x) = s(x)/p\), our unbiased estimator for \(v(x)\).
Approximate Top-\(k\) Wavelet Coefficients: Basic Random Sampling

RandomizedRecordReader \(j\) (RR\(_j\)) samples \(n_j/\varepsilon^2n\) records.

1. RR\(_j\) randomly selects \(n_j/\varepsilon^2n\) offsets in split \(j\).
2. RR\(_j\) sorts the offsets in ascending order then seeks the record at each sampled offset.

Reducer uses \(\hat{v}(x) = s(x)/p\), our unbiased estimator for \(v(x)\).
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

$\hat{s}(x)$ is an unbiased estimator of $s(x)$. 

Proof.

1. Our estimator is $\hat{s}(x) = \rho(x) + \frac{M}{\varepsilon \sqrt{m}}$.

2. Assume in the first $m'$ splits $s_j(x) < \frac{1}{\varepsilon \sqrt{m}}$.

3. Let $X_j = 1$ if $x$ is sampled in split $j$ and 0 otherwise.

4. $E[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)$.

Let $M = \sum_{j=1}^{m'} X_j$.

5. $E[M] = \varepsilon \sqrt{m} \cdot (s(x) - \rho(x))$.

6. $E[\hat{s}(x)] = E[\rho(x) + \frac{M}{\varepsilon \sqrt{m}}] = \rho(x) + (s(x) - \rho(x)) = s(x)$.
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

\(\hat{s}(x)\) is an unbiased estimator of \(s(x)\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

\( \hat{s}(x) \) is an unbiased estimator of \( s(x) \).

Proof.

- Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).
- Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

\(\hat{s}(x)\) is an unbiased estimator of \(s(x)\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).

2. Assume in the first \(m'\) splits \(s_j(x) < 1/(\varepsilon \sqrt{m})\).

3. Let \(X_j = 1\) if \(x\) is sampled in split \(j\) and 0 otherwise.
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**
\(\hat{s}(x)\) is an unbiased estimator of \(s(x)\).

**Proof.**
1. Our estimator is \(\hat{s}(x) = \rho(x) + \frac{M}{\varepsilon \sqrt{m}}\).

2. Assume in the first \(m'\) splits \(s_j(x) < 1/(\varepsilon \sqrt{m})\).

3. Let \(X_j = 1\) if \(x\) is sampled in split \(j\) and 0 otherwise.

4. \(E[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)\).
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

$\hat{s}(x)$ is an unbiased estimator of $s(x)$.

**Proof.**

1. Our estimator is $\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}$.

2. Assume in the first $m'$ splits $s_j(x) < 1/(\varepsilon \sqrt{m})$.

3. Let $X_j = 1$ if $x$ is sampled in split $j$ and 0 otherwise.

4. $E[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)$.

5. Let $M = \sum_{j=1}^{m'} X_j$. 
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

\(\hat{s}(x)\) is an unbiased estimator of \(s(x)\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).

2. Assume in the first \(m'\) splits \(s_j(x) < 1/(\varepsilon \sqrt{m})\).

3. Let \(X_j = 1\) if \(x\) is sampled in split \(j\) and \(0\) otherwise.

   \[E[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)\]

4. Let \(M = \sum_{j=1}^{m'} X_j\).

   \[E[M] = \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x)\]
Theorem

\( \hat{s}(x) \) is an unbiased estimator of \( s(x) \).

Proof.

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).

2. Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).

3. Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.

   \[ E[X_j] = \varepsilon \sqrt{m} \cdot s_j(x). \]

4. Let \( M = \sum_{j=1}^{m'} X_j \).

   \[ E[M] = \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x) = \varepsilon \sqrt{m}(s(x) - \rho(x)). \]
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

Theorem

\( \hat{s}(x) \) is an unbiased estimator of \( s(x) \).

Proof.

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).
2. Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).
3. Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.
   \[ E[X_j] = \varepsilon \sqrt{m} \cdot s_j(x). \]
4. Let \( M = \sum_{j=1}^{m'} X_j \).
   \[ E[M] = \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x) = \varepsilon \sqrt{m} (s(x) - \rho(x)). \]
5. \[ E[\hat{s}(x)] = E[\rho(x) + M/\varepsilon \sqrt{m}] \]
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

\( \hat{s}(x) \) is an unbiased estimator of \( s(x) \).

Proof.

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).

2. Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).
   Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.
   \( E[X_j] = \varepsilon \sqrt{m} \cdot s_j(x) \).

3. Let \( M = \sum_{j=1}^{m'} X_j \).
   \( E[M] = \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x) = \varepsilon \sqrt{m}(s(x) - \rho(x)) \).

4. \( E[\hat{s}(x)] = E[\rho(x) + M/\varepsilon \sqrt{m}] = \rho(x) + (s(x) - \rho(x)) \).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

\(\hat{s}(x)\) is an unbiased estimator of \(s(x)\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon\sqrt{m}\).

2. Assume in the first \(m'\) splits \(s_j(x) < 1/(\varepsilon\sqrt{m})\).

3. Let \(X_j = 1\) if \(x\) is sampled in split \(j\) and 0 otherwise.

4. Let \(M = \sum_{j=1}^{m'} X_j\).

5. \(E[X_j] = \varepsilon\sqrt{m} \cdot s_j(x)\).

6. \(E[M] = \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot s_j(x) = \varepsilon\sqrt{m}(s(x) - \rho(x))\).

7. \(E[\hat{s}(x)] = E[\rho(x) + M/\varepsilon\sqrt{m}] = \rho(x) + (s(x) - \rho(x)) = s(x)\).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

\( \hat{s}(x) \) is an estimator of \( s(x) \) with standard deviation at most \( 1/\varepsilon \).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

\(\hat{s}(x)\) is an estimator of \(s(x)\) with standard deviation at most \(1/\varepsilon\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

\[ \hat{s}(x) \text{ is an estimator of } s(x) \text{ with standard deviation at most } 1/\varepsilon. \]

**Proof.**

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).

2. Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).
Theorem

\( \hat{s}(x) \) is an estimator of \( s(x) \) with standard deviation at most \( 1/\varepsilon \).

Proof.

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).

   - Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).
   - Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

**Theorem**

\[ \hat{s}(x) \text{ is an estimator of } s(x) \text{ with standard deviation at most } 1/\varepsilon. \]

**Proof.**

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}. \)

- Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}). \)
- Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.

2. \( \text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \)
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

$\hat{s}(x)$ is an estimator of $s(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}$.

   - Assume in the first $m'$ splits $s_j(x) < 1/(\varepsilon \sqrt{m})$.
   - Let $X_j = 1$ if $x$ is sampled in split $j$ and 0 otherwise.

2. $\text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x)$. 
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

Theorem

$\hat{s}(x)$ is an estimator of $s(x)$ with standard deviation at most $1/\varepsilon$.

Proof.

1. Our estimator is $\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}$.

2. Assume in the first $m'$ splits $s_j(x) < 1/(\varepsilon \sqrt{m})$.
3. Let $X_j = 1$ if $x$ is sampled in split $j$ and 0 otherwise.
4. $\text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x)$.
5. Let $M = \sum_{j=1}^{m'} X_j$. 

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**
\(\hat{s}(x)\) is an estimator of \(s(x)\) with standard deviation at most \(1/\varepsilon\).

**Proof.**
1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).

2. Assume in the first \(m'\) splits \(s_j(x) < 1/(\varepsilon \sqrt{m})\).

3. Let \(X_j = 1\) if \(x\) is sampled in split \(j\) and 0 otherwise.

4. \(\text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x)\).

5. Let \(M = \sum_{j=1}^{m'} X_j\).

6. \(\text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j]\)
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

$s(\hat{x})$ is an estimator of $s(x)$ with standard deviation at most $1/\varepsilon$.

**Proof.**

1. Our estimator is $\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}$.

2. Assume in the first $m'$ splits $s_j(x) < 1/(\varepsilon \sqrt{m})$.

3. Let $X_j = 1$ if $x$ is sampled in split $j$ and 0 otherwise.

4. $\text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x)$.

5. Let $M = \sum_{j=1}^{m'} X_j$.

6. $\text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x)$
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

$\hat{s}(x)$ is an estimator of $s(x)$ with standard deviation at most $1/\varepsilon$.

**Proof.**

1. Our estimator is $\hat{s}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.

   - Assume in the first $m'$ splits $s_j(x) < 1/(\varepsilon\sqrt{m})$.
   - Let $X_j = 1$ if $x$ is sampled in split $j$ and 0 otherwise.
2. $\text{Var}[X_j] = \varepsilon\sqrt{m} \cdot s_j(x)(1 - \varepsilon\sqrt{m} \cdot s_j(x)) \leq \varepsilon\sqrt{m} \cdot s_j(x)$.

   - Let $M = \sum_{j=1}^{m'} X_j$.

3. $\text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot s_j(x) \leq m' \cdot \varepsilon\sqrt{m} \cdot 1/(\varepsilon\sqrt{m})$
Theorem

\( \hat{s}(x) \) is an estimator of \( s(x) \) with standard deviation at most \( 1/\varepsilon \).

Proof.

1. Our estimator is
   \[
   \hat{s}(x) = \rho(x) + \frac{M}{\varepsilon \sqrt{m}}.
   \]

2. Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).
   
   Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.
   
   \[ \text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x). \]

3. Let \( M = \sum_{j=1}^{m'} X_j \).

4. \[ \text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x) \leq m' \cdot \varepsilon \sqrt{m} \cdot 1/(\varepsilon \sqrt{m}) = m'. \]
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

\( \hat{s}(x) \) is an estimator of \( s(x) \) with standard deviation at most \( 1/\varepsilon \).

**Proof.**

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).

2. Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).

3. Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.

   \( \text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x) \).

4. Let \( M = \sum_{j=1}^{m'} X_j \).

5. \( \text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x) \leq m' \cdot \varepsilon \sqrt{m} \cdot 1/(\varepsilon \sqrt{m}) = m' \).

6. \( \text{Var}[\hat{s}(x)] = \text{Var}[M/\varepsilon \sqrt{m}] \).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

Theorem

\(\hat{s}(x)\) is an estimator of \(s(x)\) with standard deviation at most \(1/\varepsilon\).

Proof.

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon\sqrt{m}\).

2. Assume in the first \(m'\) splits \(s_j(x) < 1/(\varepsilon\sqrt{m})\).
   
   Let \(X_j = 1\) if \(x\) is sampled in split \(j\) and 0 otherwise.
   
   \[\text{Var}[X_j] = \varepsilon\sqrt{m} \cdot s_j(x)(1 - \varepsilon\sqrt{m} \cdot s_j(x)) \leq \varepsilon\sqrt{m} \cdot s_j(x).\]

3. Let \(M = \sum_{j=1}^{m'} X_j\).
   
   \[\text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon\sqrt{m} \cdot s_j(x) \leq m' \cdot \varepsilon\sqrt{m} \cdot 1/(\varepsilon\sqrt{m}) = m'.\]

4. \(\text{Var}[\hat{s}(x)] = \text{Var}[M/\varepsilon\sqrt{m}] = \text{Var}[M]/\varepsilon^2 m\)
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

Theorem

\( \hat{s}(x) \) is an estimator of \( s(x) \) with standard deviation at most \( 1/\varepsilon \).

Proof.

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).

2. Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).

3. Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.

   \[ \text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x). \]

4. Let \( M = \sum_{j=1}^{m'} X_j \).

   \[ \text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x) \leq m' \cdot \varepsilon \sqrt{m} \cdot 1/(\varepsilon \sqrt{m}) = m'. \]

5. \[ \text{Var}[\hat{s}(x)] = \text{Var}[M/\varepsilon \sqrt{m}] = \text{Var}[M]/\varepsilon^2 m \leq m'/\varepsilon^2 m \]
Approximate Top-k Wavelet Coefficients: Two-Level Sampling

**Theorem**

\( \hat{s}(x) \) is an estimator of \( s(x) \) with standard deviation at most \( 1/\varepsilon \).

**Proof.**

1. Our estimator is \( \hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m} \).

2. Assume in the first \( m' \) splits \( s_j(x) < 1/(\varepsilon \sqrt{m}) \).
   
   Let \( X_j = 1 \) if \( x \) is sampled in split \( j \) and 0 otherwise.
   
   \( \text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x)(1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x) \).

3. Let \( M = \sum_{j=1}^{m'} X_j \).
   
   \( \text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x) \leq m' \cdot \varepsilon \sqrt{m} \cdot 1/(\varepsilon \sqrt{m}) = m' \).

4. \( \text{Var}[\hat{s}(x)] = \text{Var}[M/\varepsilon \sqrt{m}] = \text{Var}[M]/\varepsilon^2 m \leq m'/\varepsilon^2 m \leq 1/\varepsilon^2 \)
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

$\hat{s}(x)$ is an estimator of $s(x)$ with standard deviation at most $1/\varepsilon$.

**Proof.**

1. Our estimator is $\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}$.

   - Assume in the first $m'$ splits $s_j(x) < 1/(\varepsilon \sqrt{m})$.
   - Let $X_j = 1$ if $x$ is sampled in split $j$ and 0 otherwise.
   
   2. $\text{Var}[X_j] = \varepsilon \sqrt{m} \cdot s_j(x) (1 - \varepsilon \sqrt{m} \cdot s_j(x)) \leq \varepsilon \sqrt{m} \cdot s_j(x)$.

   - Let $M = \sum_{j=1}^{m'} X_j$.
   
   3. $\text{Var}[M] \leq \sum_{j=1}^{m'} \text{Var}[X_j] \leq \sum_{j=1}^{m'} \varepsilon \sqrt{m} \cdot s_j(x) \leq m' \cdot \varepsilon \sqrt{m} \cdot 1/(\varepsilon \sqrt{m}) = m'$.

   - $\text{Var}[\hat{s}(x)] = \text{Var}[M/\varepsilon \sqrt{m}] = \text{Var}[M]/\varepsilon^2 m \leq m'/\varepsilon^2 m \leq 1/\varepsilon^2 \leq 1/\varepsilon$. 
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

Theorem

The expected total communication cost of our two-level sampling algorithm is \(O(\sqrt{m}/\varepsilon)\).

Proof.

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).

2. The first-level sample size is \(pn = 1/\varepsilon^2\).

3. There are \(\leq (1/\varepsilon^2)/(1/\varepsilon \sqrt{m}) = \sqrt{m}/\varepsilon\) such keys.

4. If \(s_j(x) \geq 1/(\varepsilon \sqrt{m})\), we emit \((x, s_j(x))\).

5. On expectation there are, \(P_j P_x \varepsilon \sqrt{m} \cdot s_j(x) \leq \varepsilon \sqrt{m} \cdot 1/\varepsilon^2 = \sqrt{m}/\varepsilon\).

6. By (2) and (3), the total number of emitted keys is \(O(\sqrt{m}/\varepsilon)\).
The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\widehat{s}(x) = \rho(x) + M/\varepsilon\sqrt{m}$. 
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

*The expected total communication cost of our two-level sampling algorithm is* \(O(\sqrt{m}/\varepsilon)\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).

2. The first-level sample size is \(pn = 1/\varepsilon^2\).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

The expected total communication cost of our two-level sampling algorithm is \(O(\sqrt{m}/\varepsilon)\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).

2. The first-level sample size is \(pn = 1/\varepsilon^2\).

3. If \(s_j(x) \geq 1/(\varepsilon \sqrt{m})\) we emit \((x, s_j(x))\).
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

*The expected total communication cost of our two-level sampling algorithm is \(O(\sqrt{m}/\varepsilon)\).*

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon\sqrt{m}\).
   - The first-level sample size is \(pn = 1/\varepsilon^2\).
   - If \(s_j(x) \geq 1/(\varepsilon\sqrt{m})\) we emit \((x, s_j(x))\).
2. There are \(\leq (1/\varepsilon^2)/(1/\varepsilon\sqrt{m}) = \sqrt{m}/\varepsilon\) such keys.
Approximate Top-$k$ Wavelet Coefficients: Two-Level Sampling

**Theorem**

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\epsilon)$.

**Proof.**

1. Our estimator is $\hat{s}(x) = \rho(x) + M/\epsilon\sqrt{m}$.

   - The first-level sample size is $pn = 1/\epsilon^2$.
   - If $s_j(x) \geq 1/(\epsilon\sqrt{m})$ we emit $(x, s_j(x))$.
     - There are $\leq (1/\epsilon^2)/(1/\epsilon\sqrt{m}) = \sqrt{m}/\epsilon$ such keys.
   - If $s_j(x) < 1/(\epsilon\sqrt{m})$, we emit $(x, \text{null})$ with probability $\epsilon\sqrt{m} \cdot s_j x$. 

Theorem

The expected total communication cost of our two-level sampling algorithm is $O(\sqrt{m}/\varepsilon)$.

Proof.

1. Our estimator is $\hat{s}(x) = \rho(x) + M/\varepsilon\sqrt{m}$.
   - The first-level sample size is $pn = 1/\varepsilon^2$.
   - If $s_j(x) \geq 1/(\varepsilon\sqrt{m})$ we emit $(x, s_j(x))$.
   - There are $\leq (1/\varepsilon^2)/(1/\varepsilon\sqrt{m}) = \sqrt{m}/\varepsilon$ such keys.
2. If $s_j(x) < 1/(\varepsilon\sqrt{m})$, we emit $(x, \text{null})$ with probability $\varepsilon\sqrt{m} \cdot s_j(x)$.
   - On expectation there are,
   $$\sum_j \sum_x \varepsilon\sqrt{m} \cdot s_j(x) \leq \varepsilon\sqrt{m} \cdot 1/\varepsilon^2$$
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

The expected total communication cost of our two-level sampling algorithm is \(O(\sqrt{m}/\epsilon)\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\epsilon \sqrt{m}\).

2. The first-level sample size is \(pn = 1/\epsilon^2\).
   - If \(s_j(x) \geq 1/(\epsilon \sqrt{m})\) we emit \((x, s_j(x))\).
     - There are \(\leq (1/\epsilon^2)/(1/\epsilon \sqrt{m}) = \sqrt{m}/\epsilon\) such keys.
   - If \(s_j(x) < 1/(\epsilon \sqrt{m})\), we emit \((x, \text{null})\) with probability \(\epsilon \sqrt{m} \cdot s_j x\).

3. On expectation there are,
   \[
   \sum_j \sum_x \epsilon \sqrt{m} \cdot s_j(x) \leq \epsilon \sqrt{m} \cdot 1/\epsilon^2 = \sqrt{m}/\epsilon.
   \]
Approximate Top-\(k\) Wavelet Coefficients: Two-Level Sampling

**Theorem**

The expected total communication cost of our two-level sampling algorithm is \(O(\sqrt{m}/\varepsilon)\).

**Proof.**

1. Our estimator is \(\hat{s}(x) = \rho(x) + M/\varepsilon \sqrt{m}\).
2. The first-level sample size is \(pn = 1/\varepsilon^2\).
3. If \(s_j(x) \geq 1/(\varepsilon \sqrt{m})\) we emit \((x, s_j(x))\).
   - There are \(\leq (1/\varepsilon^2)/(1/\varepsilon \sqrt{m}) = \sqrt{m}/\varepsilon\) such keys.
4. If \(s_j(x) < 1/(\varepsilon \sqrt{m})\), we emit \((x, \text{null})\) with probability \(\varepsilon \sqrt{m} \cdot s_j x\).
   - On expectation there are,
     \[\sum_j \sum_x \varepsilon \sqrt{m} \cdot s_j(x) \leq \varepsilon \sqrt{m} \cdot 1/\varepsilon^2 = \sqrt{m}/\varepsilon.\]
5. By (2) and (3), the total number of emitted keys is \(O(\sqrt{m}/\varepsilon)\).
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

\[ n_j = \text{records in split } j \]

\[ s_j = \text{split } j \text{ sample frequency vector} \]

---

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

$n_j = \text{records in split } j$

$s_j = \text{split } j \text{ sample frequency vector}$

1. RandomizedRecordReader $j$ samples $t_j = n_j/\varepsilon^2 n$ records.
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

$n_j = \text{records in split } j$
$s_j = \text{split } j \text{ sample frequency vector}$

1. RandomizedRecordReader $j$ samples $t_j = n_j/\varepsilon^2 n$ records.
RandomizedRecordReader \( j \) samples \( t_j = \frac{n_j}{\epsilon^2 n} \) records.
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

1. RandomizedRecordReader $j$ samples $t_j = n_j/\varepsilon^2 n$ records.

$n_j =$ records in split $j$
$s_j =$ split $j$ sample frequency vector
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

\[ n_j = \text{records in split } j \]
\[ s_j = \text{split } j \text{ sample frequency vector} \]

1. RandomizedRecordReader \(j\) samples \(t_j = n_j/\varepsilon^2 n\) records.
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

1. RandomizedRecordReader \(j\) samples \(t_j = n_j/\varepsilon^2 n\) records.
2. If \(s_j(x) > \varepsilon t_j\), the Mapper emits \((x, s_j(x))\).

\(n_j\) = records in split \(j\)
\(s_j\) = split \(j\) sample frequency vector
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

RandomizedRecordReader $j$ samples $t_j = n_j / \varepsilon^2 n$ records.

If $s_j(x) > \varepsilon t_j$, the Mapper emits $(x, s_j(x))$. 

$n_j =$ records in split $j$
$s_j =$ split $j$ sample frequency vector
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

1. RandomizedRecordReader \(j\) samples \(t_j = n_j/\varepsilon^2 n\) records.
2. If \(s_j(x) > \varepsilon t_j\), the Mapper emits \((x, s_j(x))\).

\(n_j = \) records in split \(j\)
\(s_j = \) split \(j\) sample frequency vector
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

1. RandomizedRecordReader \(j\) samples \(t_j = n_j/\epsilon^2 n\) records.
2. If \(s_j(x) > \epsilon t_j\), the Mapper emits \((x, s_j(x))\).
3. Reducer uses \(\hat{v}(x) = s(x)/p\), our estimator for \(v(x)\).

\(n_j = \text{records in split } j\)
\(s_j = \text{split } j \text{ sample frequency vector}\)

Jeffrey Jestes, Ke Yi, Feifei Li
Building Wavelet Histograms on Large Data in MapReduce
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

1. RandomizedRecordReader $j$ samples $t_j = n_j/\varepsilon^2 n$ records.
2. If $s_j(x) > \varepsilon t_j$, the Mapper emits $(x, s_j(x))$.
3. Reducer uses $\hat{v}(x) = s(x)/p$, our estimator for $v(x)$.

$n_j =$ records in split $j$
$s_j =$ split $j$ sample frequency vector

Jeffrey Jestes, Ke Yi, Feifei Li

Building Wavelet Histograms on Large Data in MapReduce
 aproximado Top-\(k\) coeficientes de ondaletas: muestreo mejorado

\[
\begin{align*}
\text{split 1} & \quad \text{split 2} \\
\text{split 3} & \quad \text{split 4} \\
\text{JobTracker} & \quad \text{RandomizedRecordReader} \\
& \quad \text{MapRunner} \\
& \quad \text{Mapper} \\
& \quad \text{Reducer} \\
& \quad \text{Centralized Wavelet Top-}k \\
& \quad \text{close()} \\
& \quad (x, \hat{v}(x)) \\
& \quad (x, s(x))
\end{align*}
\]

1. RandomizedRecordReader \(j\) samples \(t_j = n_j / \varepsilon^2 n\) records.
2. If \(s_j(x) > \varepsilon t_j\), the Mapper emits \((x, s_j(x))\).
3. Reducer uses \(\hat{v}(x) = s(x) / p\), our estimator for \(v(x)\).
Approximate Top-\(k\) Wavelet Coefficients: Improved Sampling

1. RandomizedRecordReader \(j\) samples \(t_j = n_j/\varepsilon^2 n\) records.
2. If \(s_j(x) > \varepsilon t_j\), the Mapper emits \((x, s_j(x))\).
3. Reducer uses \(\hat{v}(x) = s(x)/p\), our estimator for \(v(x)\).
Approximate Top-$k$ Wavelet Coefficients: Improved Sampling

1. RandomizedRecordReader $j$ samples $t_j = n_j / \varepsilon^2 n$ records.
2. If $s_j(x) > \varepsilon t_j$, the Mapper emits $(x, s_j(x))$.
3. Reducer uses $\hat{v}(x) = s(x)/p$, our estimator for $v(x)$.