Composable, Scalable, and Accurate Weight Summarization of Unaggregated Data Sets

Edith Cohen (AT&T)    Nick Duffield (AT&T)
Haim Kaplan (Tel-Aviv)    Carsten Lund (AT&T)
Mikkel Thorup (AT&T)
Weight summaries for unaggregated data

**Data** set $\mathcal{D}$ of weighted keys $(i, w)$ is *unaggregated:* key $i$ may appear multiple times with different weights $w$.

**Query** For arbitrary selection $Q$ of keys, report total weight associated with these keys.

- Aggregation over each key $\rightarrow \{9, 3, 3\}$.
- Sampling with unbiased estimation $\rightarrow \{9, 6\}$ or $\{9, 6\}$.
Weight summaries for unaggregated data

**Data** set $D$ of weighted keys $(i, w)$ is *unaggregated*: key $i$ may appear multiple times with different weights $w$.

**Query** For arbitrary selection $Q$ of keys, report total weight associated with these keys.

Resource constraints dictate smaller **weight summary** of $D$.

- Aggregation over each key $\rightarrow \{9, 3, 3\}$.
- Sampling with unbiased estimation $\rightarrow \{9, 6\}$ or $\{9, 6\}$.
Weight summaries for unaggregated data

Data set $D$ of weighted keys $(i, w)$ is unaggregated:
key $i$ may appear multiple times with different weights $w$.

Query For arbitrary selection $Q$ of keys,
report total weight associated with these keys.

Resource constraints dictate smaller weight summary of $D$.

Summarizing $\{4, 3, 1, 5, 2\}$

- Aggregation over each key $\rightarrow \{9, 3, 3\}$.
- Sampling with unbiased estimation $\rightarrow \{9, 6\}$ or $\{9, 6\}$. 
Weight summaries for unaggregated data

**Data** set $\mathcal{D}$ of weighted keys $(i, w)$ is *unaggregated*:
key $i$ may appear multiple times with different weights $w$.

**Query** For arbitrary selection $Q$ of keys,
report total weight associated with these keys.

Resource constraints dictate smaller **weight summary** of $\mathcal{D}$.

Summarizing $\{4, 3, 1, 5, 2\}$

- Aggregation over each key $\rightarrow \{9, 3, 3\}$.
- Sampling with unbiased estimation $\rightarrow \{9, 6\}$ or $\{9, 6\}$. 
Weight summaries for unaggregated data

Data set $\mathcal{D}$ of weighted keys $(i, w)$ is unaggregated:
key $i$ may appear multiple times with different weights $w$.

Query For arbitrary selection $Q$ of keys,
report total weight associated with these keys.

Resource constraints dictate smaller weight summary of $\mathcal{D}$.

Summarizing $\{4, 3, 1, 5, 2\}$

- **Aggregation** over each key $\rightarrow \{9, 3, 3\}$.
- **Sampling** with unbiased estimation $\rightarrow \{9, 6\}$ or $\{9, 6\}$. 
Weight summaries for unaggregated data

**Data** set $\mathcal{D}$ of weighted keys $(i, w)$ is *unaggregated*: key $i$ may appear multiple times with different weights $w$.

**Query** For arbitrary selection $Q$ of keys, report total weight associated with these keys. Resource constraints dictate smaller **weight summary** of $\mathcal{D}$.

Summarizing $\{4, 3, 1, 5, 2\}$

- **Aggregation** over each key $\rightarrow \{9, 3, 3\}$.
- **Sampling** with unbiased estimation $\rightarrow \{9, 6\}$ or $\{9, 6\}$.

Queries directed to resulting weight summary. For selection $Q = \{\text{red, blue}\}$, the estimated weight is 15 or 9.
Weight summaries for unaggregated data

Data set $\mathcal{D}$ of weighted keys $(i, w)$ is unaggregated: key $i$ may appear multiple times with different weights $w$.

Query For arbitrary selection $Q$ of keys, report total weight associated with these keys.

Resource constraints dictate smaller weight summary of $\mathcal{D}$.

Summarizing $\{4, 3, 1, 5, 2\}$

- **Aggregation** over each key $\rightarrow \{9, 3, 3\}$.
- **Sampling** with unbiased estimation $\rightarrow \{9, 6\}$ or $\{9, 6\}$.

Queries directed to resulting weight summary.

For selection $Q = \{\text{red}, \text{blue}\}$, the estimated weight is 15 or 9.

Errors go down with larger samples and selections.
Weight summaries for unaggregated data

**Data** set $\mathcal{D}$ of weighted keys $(i, w)$ is *unaggregated*:
key $i$ may appear multiple times with different weights $w$.

**Query** For arbitrary selection $Q$ of keys,
report total weight associated with these keys.
Resource constraints dictate smaller **weight summary** of $\mathcal{D}$.

**Summarizing** $\{4, 3, 1, 5, 2\}$
- **Aggregation** over each key $\rightarrow \{9, 3, 3\}$.
- **Sampling** with unbiased estimation $\rightarrow \{9, 6\}$ or $\{9, 6\}$.

Queries directed to resulting weight summary.
For selection $Q = \{\text{red, blue}\}$, the estimated weight is 15 or 9.
Errors go down with larger samples and selections.
Extensions to multiple and signed weights possible.
Information flow trees (IFT)

Iteratively collecting and summarizing data spread over time and space.
Information flow trees (IFT)

Iteratively collecting and summarizing data spread over time and space.

Each node summarizes information from children, producing weight summary of descending leaves.
IFT examples

Stream:

Distributed streams:

Servers:
Classic special case of unaggregated unit keys

In experiments, our general schemes outperforms classics:
Classic special case of unaggregated unit keys

In experiments, our general schemes outperforms classics:

**Uniform as in Cisco’s netflow (NF)**

- Sampling rate \( r \in [0, 1] \).
- Weighted key \((i, w)\) added to reservoir with probability \( r \) and weight \( w/r \).
Classic special case of unaggregated unit keys

In experiments, our general schemes outperforms classics:

**Uniform as in Cisco’s netflow (NF)**
- Sampling rate $r \in [0, 1]$.
- Weighted key $(i, w)$ added to reservoir with probability $r$ and weight $w/r$.

**Sample and Hold (SH) [Gibbons and Matias, SIGMOD’98]**
- If key $i$ already in reservoir, add $w$ to its weight.
- Otherwise include $i$ with probability $r$ and initial weight $w/r$. 
Classic special case of unaggregated unit keys

In experiments, our general schemes outperforms classics:

**Uniform as in Cisco’s netflow (NF)**

- Sampling rate \( r \in [0, 1] \).
- Weighted key \((i, w)\) added to reservoir with probability \( r \) and weight \( w/r \).

**Sample and Hold (SH) [Gibbons and Matias, SIGMOD’98]**

- If key \( i \) already in reservoir, add \( w \) to its weight.
- Otherwise include \( i \) with probability \( r \) and initial weight \( w/r \).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>NF</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SH
NF
3
3
4
5
6
56
Classic special case of unaggregated unit keys

In experiments, our general schemes outperforms classics:

**Uniform as in Cisco’s netflow (NF)**
- Sampling rate \( r \in [0, 1] \).
- Weighted key \((i, w)\) added to reservoir with probability \( r \) and weight \( w/r \).

**Sample and Hold (SH) [Gibbons and Matias, SIGMOD’98]**
- If key \( i \) already in reservoir, add \( w \) to its weight.
- Otherwise include \( i \) with probability \( r \) and initial weight \( w/r \).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sampling rate \( r \) adapted to give at most \( k \) keys in reservoir.
|   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| NF | 3 | 3 | 6 | 6 | 6 | 6 | 6 |
| SH | 3 | 3 | 4 | 5 | 6 | 4 | 5 | 5 | 5 |

**Legend:**
- NF: Non-Functional
- SH: Software-Hardware
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our general scheme
Choose weighted sampling scheme for each node
Choose weighted sampling scheme for each node

Summarizing 7, 3, 2, 2 as 7, 7?.
VarOpt$_k$ sampling $k$ keys $i$ with weight estimates $\hat{w}_i$

(i) Inclusion probabilities proportional to size (ipps).

For some common threshold $\tau$

If $w_i \geq \tau$, then key $i$ included with estimate $\hat{w}_i = w_i$.

Else $i$ included with probability $p_i = w_i/\tau$ and estimate $\hat{w}_i = \tau$. 
VarOpt$_k$ sampling $k$ keys $i$ with weight estimates $\hat{w}_i$

(i) Inclusion probabilities proportional to size (ipps).
For some common threshold $\tau$
If $w_i \geq \tau$, then key $i$ included with estimate $\hat{w}_i = w_i$.
Else $i$ included with probability $p_i = w_i/\tau$ and estimate $\hat{w}_i = \tau$.
Here $\tau = 0$ if $k \geq n$. Otherwise $\tau$ unique value such that

$$\sum_i p_i = \sum_i \min\{1, w_i/\tau\} = k.$$
VarOpt\(_k\) sampling \(k\) keys \(i\) with weight estimates \(\hat{w}_i\)

(i) Inclusion probabilities proportional to size (ipps).
For some common threshold \(\tau\)
If \(w_i \geq \tau\), then key \(i\) included with estimate \(\hat{w}_i = w_i\).
Else \(i\) included with probability \(p_i = w_i/\tau\) and estimate \(\hat{w}_i = \tau\).
Here \(\tau = 0\) if \(k \geq n\). Otherwise \(\tau\) unique value such that
\[
\sum_i p_i = \sum_i \min\{1, w_i/\tau\} = k.
\]

(ii) Sample contains at most \(k\) items.
VarOpt\(_k\) sampling \(k\) keys \(i\) with weight estimates \(\hat{w}_i\)

(i) Inclusion probabilities proportional to size (ipps).
For some common threshold \(\tau\)
If \(w_i \geq \tau\), then key \(i\) included with estimate \(\hat{w}_i = w_i\).
Else \(i\) included with probability \(p_i = w_i/\tau\) and estimate \(\hat{w}_i = \tau\).
Here \(\tau = 0\) if \(k \geq n\). Otherwise \(\tau\) unique value such that

\[
\sum_i p_i = \sum_i \min\{1, w_i/\tau\} = k.
\]

(ii) Sample contains at most \(k\) items.
(iii) No positive covariances.
VarOpt\(_k\) sampling \(k\) keys \(i\) with weight estimates \(\hat{w}_i\)

(i) Inclusion probabilities proportional to size (ipps).
For some common threshold \(\tau\)
If \(w_i \geq \tau\), then key \(i\) included with estimate \(\hat{w}_i = w_i\).
Else \(i\) included with probability \(p_i = w_i/\tau\) and estimate \(\hat{w}_i = \tau\).
Here \(\tau = 0\) if \(k \geq n\). Otherwise \(\tau\) unique value such that
\[\sum_i p_i = \sum_i \min\{1, w_i/\tau\} = k.\]

(ii) Sample contains at most \(k\) items.
(iii) No positive covariances.

With \(7, 3, 2, 2\) and \(k = 2\), we get \(\tau = 7,\)
\(p = 1, p = \frac{3}{7}, p = \frac{2}{7}\), and \(p = \frac{2}{7}\). Outcome \(7, 7\) with probability \(\frac{3}{7}\).
\textbf{VarOpt}_k \text{ sampling } k \text{ keys } i \text{ with weight estimates } \hat{w}_i \\

(i) Inclusion probabilities proportional to size (ipps). 
For some common \textbf{threshold} \( \tau \) 
If \( w_i \geq \tau \), then key \( i \) included with estimate \( \hat{w}_i = w_i \). 
Else \( i \) included with probability \( p_i = w_i/\tau \) and estimate \( \hat{w}_i = \tau \). 
Here \( \tau = 0 \) if \( k \geq n \). Otherwise \( \tau \) unique value such that 
\[
\sum_i p_i = \sum_i \min\{1, w_i/\tau\} = k.
\]

(ii) Sample contains at most \( k \) items. 
(iii) No positive covariances. 

With 7, 3, 2, 2 and \( k = 2 \), we get \( \tau = 7 \), 
\( p = 1, p = \frac{3}{7}, p = \frac{2}{7} \), and \( p = \frac{2}{7} \). Outcome 7, 7 with probability \( \frac{3}{7} \). 
(i)–(iii) imply minimal average variance for any subset size \( m \).
**VarOpt**$_k$ sampling $k$ keys $i$ with weight estimates $\hat{w}_i$

(i) Inclusion probabilities proportional to size (ipps). For some common threshold $\tau$
If $w_i \geq \tau$, then key $i$ included with estimate $\hat{w}_i = w_i$.
Else $i$ included with probability $p_i = w_i/\tau$ and estimate $\hat{w}_i = \tau$.
Here $\tau = 0$ if $k \geq n$. Otherwise $\tau$ unique value such that

$$\sum_i p_i = \sum_i \min\{1, w_i/\tau\} = k.$$

(ii) Sample contains at most $k$ items.
(iii) No positive covariances.

With $7, 3, 2, 2$ and $k = 2$, we get $\tau = 7$,
$p = 1$, $p = \frac{3}{7}$, $p = \frac{2}{7}$, and $p = \frac{2}{7}$. Outcome $7, 7$ with probability $\frac{3}{7}$.

(i)–(iii) imply minimal average variance for any subset size $m$.
[Sunter 77, Chao 82, Tille 96, CDHLT 09]
IFT[VarOpt_k]
IFT[VarOptₖ]

If no key appears in two leaves, the root has global VarOptₖ
[CDHLT 09]
If no key appears in two leaves, the root has global VarOpt$_k$ [CDHLT 09]

For unaggregated data allowing duplicate keys, we prove that global VarOpt$_k$ is not possible.
How good is IFT[VarOpt$_k$]?

With unaggregated data, we lose variance optimality, but preserve exact total and no positive covariances.
How good is IFT[VarOpt$_k$]?

With unaggregated data, we lose variance optimality, but preserve exact total and no positive covariances. As a heuristic we compare its empirical variance with

- the theoretical optimum $\text{VAROPT}_k$ on the aggregated data.
- the netflow solution (NF) including each unit independently.
- sample and hold (SH), only defined for input streams.
How good is IFT[VarOpt$_k$]?

With unaggregated data, we lose variance optimality, but preserve exact total and no positive covariances. As a heuristic we compare its empirical variance with

- the theoretical optimum $\text{VAROPT}_k$ on the aggregated data.
- the netflow solution (NF) including each unit independently.
- sample and hold (SH), only defined for input streams.
How good is IFT[VarOpt\textsubscript{k}]?

With unaggregated data, we lose variance optimality, but preserve exact total and no positive covariances. As a heuristic we compare its empirical variance with

- the theoretical optimum $VAROPT\textsubscript{k}$ on the aggregated data.
- the netflow solution (NF) including each unit independently.
- sample and hold (SH), only defined for input streams.
How good is $\text{IFT}[\text{VarOpt}_k]$?

With unaggregated data, we lose variance optimality, but preserve exact total and no positive covariances. As a heuristic we compare its empirical variance with

- the theoretical optimum $\text{VAROPT}_k$ on the aggregated data.
- the netflow solution (NF) including each unit independently.
- sample and hold (SH), only defined for input streams.

We consider both synthetic Pareto distributions and real data from Internet routers and Netflix.
How good is IFT[VarOpt$_k$]?

With unaggregated data, we lose variance optimality, but preserve exact total and no positive covariances. As a heuristic we compare its empirical variance with

- the theoretical optimum VAROPT$_k$ on the aggregated data.
- the netflow solution (NF) including each unit independently.
- sample and hold (SH), only defined for input streams.

We consider both synthetic Pareto distributions and real data from Internet routers and Netflix. The data appear as streams

![Data Streams Diagram]
How good is IFT[VarOpt$_k$]?

With unaggregated data, we lose variance optimality, but preserve exact total and no positive covariances. As a heuristic we compare its empirical variance with

- the theoretical optimum VarOpt$_k$ on the aggregated data.
- the netflow solution (NF) including each unit independently.
- sample and hold (SH), only defined for input streams.

We consider both synthetic Pareto distributions and real data from Internet routers and Netflix. The data appear as streams

```
(a,2) (c,1) (b,2) (a,4) (d,2) (d,6) (c,2) (a,3) (a,2) (b,1)
```

and as coming from distributed servers.
Synthetic Pareto data sets with increasing $\alpha$ ($\rightarrow$ less heavy tail)
SQE on Pareto with increasing $\alpha$

Pareto $n=1000$, $k=100$

- NF
- SH
- IFT[$\text{VarOpt}$](stream)
- IFT[$\text{VarOpt}$](5 servers)
- VarOpt (aggregated)
SQE on Pareto with increasing $\alpha$ ratio to OPT

Pareto $n=1000$ $k=100$

Ratio to optimal sum of square errors

Pareto power parameter

NF

SH

IFT[VarOpt](5 servers)

IFT[VarOpt](stream)

VarOpt (aggregated)
SQE on Pareto, $\alpha = 0.6$ with increasing sample size

Pareto $n=1000$ alpha=0.6

- NF
- SH
- IFT[VarOpt](5 servers)
- IFT[VarOpt](stream)
- VarOpt (aggregated)
SQE on Pareto, $\alpha = 0.6$ with increasing sample size ratio to OPT
Real data sets

- Fraction of total weight vs. number of heaviest keys
- Campus flows
- Campus src-dest
- Peering flows
- Peering src-dest
- Peering dest
- Netflix
SQE on Netflix with increasing sample size

![Graph showing the normalized sum of square errors for different methods across varying sample sizes. The methods include NF, SH, IFT[VarOpt](10 servers), IFT[VarOpt](100 servers), IFT[VarOpt](stream), and VarOpt (aggregated). The x-axis represents the sample size (k), and the y-axis represents the normalized sum of square errors. The graph demonstrates decreasing error rates as the sample size increases for all methods.]
SQE on Netflix with increasing sample size ratio to OPT
SQE on Netflix selecting 1991 films

![Graph showing normalized square error vs. k for different methods: NF, SH, IFT[VarOpt](100 servers), IFT[VarOpt](10 servers), IFT[VarOpt](stream).]
SQE on Campus netflows with increasing sample size

![Graph showing the normalized sum of square errors for different methods with increasing sample size. The x-axis represents the sample size (k), and the y-axis represents the normalized sum of square errors. The methods include NF, SH, IFT[VarOpt](stream), and VarOpt (aggregated).]
SQE on Campus netflows with increasing sample size ratio to OPT

![Graph showing the ratio of sum of square errors to optimal sum of square errors for different data sets. The x-axis represents the sample size ratio to OPT (k), and the y-axis represents the ratio of the sum of square errors. The data sets include NF, SH, IFT[VarOpt](stream), and VarOpt (aggregated). The graph illustrates how the ratio increases with increasing sample size.]
Concluding remarks

Introduced flexible IFT[VarOpt$_k$] to iteratively collect and summarize weighted keys spread over time and space.
Concluding remarks

Introduced flexible IFT[VarOpt$_k$] to iteratively collect and summarize weighted keys spread over time and space.

In experiments:

<table>
<thead>
<tr>
<th>scheme</th>
<th>input</th>
<th>distance to OPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF</td>
<td>same on streams and servers</td>
<td>50-1000%</td>
</tr>
<tr>
<td>SH</td>
<td>only for streams</td>
<td>10-40%</td>
</tr>
<tr>
<td>IFT[Varopt$_k$]</td>
<td>streams</td>
<td>0-5%</td>
</tr>
<tr>
<td>IFT[Varopt$_k$]</td>
<td>servers</td>
<td>0-1%</td>
</tr>
<tr>
<td>Varopt$_k$</td>
<td>preaggregated data</td>
<td>0%</td>
</tr>
</tbody>
</table>
Concluding remarks

Introduced flexible IFT[VarOpt\textsubscript{k}] to iteratively collect and summarize weighted keys spread over time and space.

In experiments:

<table>
<thead>
<tr>
<th>scheme</th>
<th>input</th>
<th>distance to OPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF</td>
<td>same on streams and servers</td>
<td>50-1000%</td>
</tr>
<tr>
<td>SH</td>
<td>only for streams</td>
<td>10-40%</td>
</tr>
<tr>
<td>IFT[Varopt\textsubscript{k}]</td>
<td>streams</td>
<td>0-5%</td>
</tr>
<tr>
<td>IFT[Varopt\textsubscript{k}]</td>
<td>servers</td>
<td>0-1%</td>
</tr>
<tr>
<td>Varopt\textsubscript{k}</td>
<td>preaggregated data</td>
<td>0%</td>
</tr>
</tbody>
</table>

IFT[Varopt\textsubscript{k}] converges to aggregated optimum when weights more aggregated.
Concluding remarks

Introduced flexible IFT[VarOpt$_k$] to iteratively collect and summarize weighted keys spread over time and space.

In experiments:

<table>
<thead>
<tr>
<th>scheme</th>
<th>input</th>
<th>distance to OPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF</td>
<td>same on streams and servers</td>
<td>50-1000%</td>
</tr>
<tr>
<td>SH</td>
<td>only for streams</td>
<td>10-40%</td>
</tr>
<tr>
<td>IFT[Varopt$_k$]</td>
<td>streams</td>
<td>0-5%</td>
</tr>
<tr>
<td>IFT[Varopt$_k$]</td>
<td>servers</td>
<td>0-1%</td>
</tr>
<tr>
<td>Varopt$_k$</td>
<td>preaggregated data</td>
<td>0%</td>
</tr>
</tbody>
</table>

IFT[Varopt$_k$] converges to aggregated optimum when weights more aggregated.

—NF and SH do not benefit as they divide weights into units.
Concluding remarks

Introduced flexible IFT[VarOpt$_k$] to iteratively collect and summarize weighted keys spread over time and space.

In experiments:

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Input</th>
<th>Distance to OPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF</td>
<td>same on streams and servers</td>
<td>50-1000%</td>
</tr>
<tr>
<td>SH</td>
<td>only for streams</td>
<td>10-40%</td>
</tr>
<tr>
<td>IFT[Varopt$_k$]</td>
<td>streams</td>
<td>0-5%</td>
</tr>
<tr>
<td>IFT[Varopt$_k$]</td>
<td>servers</td>
<td>0-1%</td>
</tr>
<tr>
<td>Varopt$_k$</td>
<td>preaggregated data</td>
<td>0%</td>
</tr>
</tbody>
</table>

IFT[Varopt$_k$] converges to aggregated optimum when weights more aggregated.
—NF and SH do not benefit as they divide weights into units.

We conjecture that IFT[Varopt$_k$] has competitive ratio of $O(1)$ for all possible streams.
Concluding remarks

Introduced flexible IFT[VarOpt\(_k\)] to iteratively collect and summarize weighted keys spread over time and space.

In experiments:

<table>
<thead>
<tr>
<th>scheme</th>
<th>input</th>
<th>distance to OPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF</td>
<td>same on streams and servers</td>
<td>50-1000%</td>
</tr>
<tr>
<td>SH</td>
<td>only for streams</td>
<td>10-40%</td>
</tr>
<tr>
<td>IFT[Varopt(_k)]</td>
<td>streams</td>
<td>0-5%</td>
</tr>
<tr>
<td>IFT[Varopt(_k)]</td>
<td>servers</td>
<td>0-1%</td>
</tr>
<tr>
<td>Varopt(_k)</td>
<td>preaggregated data</td>
<td>0%</td>
</tr>
</tbody>
</table>

IFT[Varopt\(_k\)] converges to aggregated optimum when weights more aggregated.
—NF and SH do not benefit as they divide weights into units.

We conjecture that IFT[Varopt\(_k\)] has competitive ratio of \(O(1)\) for all possible streams.
—SH is \(\Omega(\log k)\) for some concrete streams.