Sketching for Data Streams

Michael Kapralov

EPFL

August 30, 2023
Streaming model (Alon, Matias, Szegedy’96)

Observe a (very long) stream of data, e.g. IP packets, tweets, search queries....

Task: maintain (approximate) statistics of the stream
Streaming model

- Single pass over the data: \(i_1, i_2, \ldots, i_N\)
  Typically, assume \(N\) is known

- Small (sublinear) storage: typically \(N^{\alpha}, \alpha < 1\) or \(\log^{O(1)} N\)
  Units of storage: bits, words or ‘data items’ (e.g., points, nodes/edges)

- Fast processing time per element

- Mostly randomized algorithms
  Randomness often necessary
Heavy hitters problem

- Single pass over the data: \(i_1, i_2, \ldots, i_N\)

  Assume \(N\) is known

- Output \(k\) most frequent items

  (Heavy hitters)

- Small storage: will get \(O(k \log N)\)

  Much better than storing all items!

---

1  2  3  4  5  6  7  8  9  10
Heavy hitters problem

- Single pass over the data: \(i_1, i_2, \ldots, i_N\)
  
  Assume \(N\) is known

- Output \(k\) most frequent items
  
  (Heavy hitters)

- Small storage: will get \(O(k \log N)\)
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!

```
1 2 3 4 5 6 7 8 9 10
```

```plaintext
3 4
```
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!

```
3 4 6 3
```
Heavy hitters problem

- Single pass over the data: \(i_1, i_2, \ldots, i_N\)
  
  Assume \(N\) is known

- Output \(k\) most frequent items
  
  (Heavy hitters)

- Small storage: will get \(O(k \log N)\)
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!

```
1 2 3 4 6 3 2 10 3
```
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!
**Heavy hitters problem**

- Single pass over the data: \( i_1, i_2, \ldots, i_N \)
  
  Assume \( N \) is known

- Output \( k \) most frequent items
  
  (Heavy hitters)

- Small storage: will get \( O(k \log N) \)
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: \( i_1, i_2, \ldots, i_N \)
  
  Assume \( N \) is known

- Output \( k \) most frequent items
  
  (Heavy hitters)

- Small storage: will get \( O(k \log N) \)
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!

![Bar chart showing frequency distribution of items 1 to 10]
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!

```
3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7 4 4 2 2
```
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

  Assume $N$ is known

- Output $k$ most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
Estimating IP flows through a router

Estimate the dominant IP flows through a router

<table>
<thead>
<tr>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

destination

<table>
<thead>
<tr>
<th>Destination</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

```
<table>
<thead>
<tr>
<th></th>
<th>DATA</th>
<th>DATA</th>
<th>DATA</th>
<th>DATA</th>
<th>DATA</th>
<th>DATA</th>
<th>DATA</th>
<th>DATA</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>dest</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td></td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

```
source 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

destination
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Source**: The row on the left represents the source IP addresses.
- **Destination**: The column at the top represents the destination IP addresses.
- **Data**: The values in the table indicate the number of IP flows from the source to the destination.
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

The table shows the traffic patterns for different source and destination IP addresses. The numbers indicate the flow of data packets.
Estimating IP flows through a router

![Image of a router](image)

<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>

```
<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
```
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 2 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 0 0 1 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 3 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 1 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

### Table Explanation:
- **Source**: Represents the origin of IP flows.
- **Destination**: Represents the destination of IP flows.
- The numbers in the table indicate the flow rates between different source and destination pairs.
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

destination

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

![Router Diagram]

<table>
<thead>
<tr>
<th>Data</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

destination

[^image]
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 3</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

**Note:** The table above represents the traffic flow through a router, with each cell indicating the number of packets sent from a source to a destination.
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td></td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The table above represents the IP flows through a router. The source and destination are listed, with each row indicating the number of flows from one source to one destination.
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>3</td>
<td>0 0 0 0 0 0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 4 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 3 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 1 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
## Estimating IP flows through a router

### Table: IP flow estimation

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

**Note:** The table above represents the estimated IP flows through a router. The values indicate the number of packets sent from a source to a destination.
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 4 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>4 0 0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 1 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 4 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 4 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 1 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td></td>
</tr>
</tbody>
</table>

The table shows the flows between sources and destinations, with values indicating the number of data packets.
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

The table represents the flows from different sources to different destinations through a router. Each row indicates the number of flows from one source to one destination.
Estimating IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 5 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>4 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

The table represents the estimated IP flows through a router. The source and destination IP addresses are listed, with the number of flows indicated for each pair.
Estimating IP flows through a router

Estimate the dominant IP flows through a router

<table>
<thead>
<tr>
<th>destination</th>
<th>1 0 0 0 0 0 0 0 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>source</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Estimating IP flows through a router

Estimate the dominant IP flows through a router
Estimating IP flows through a router

Estimate the dominant IP flows through a router

Trivial: store all distinct IP pairs
Space complexity: $\Theta(N)$
Estimating IP flows through a router

Estimate the dominant IP flows through a router

Source  1  4  1  1  1

Destination  1  5  1  1

Trivial: store all distinct IP pairs

Space complexity: $\Theta(N)$

This lecture: solve in space $O(\log N)$

Exponential improvement!
Given a set of items as a stream (e.g. queries on google.com over a period of time)

Geneva to NYC, coffee in Geneva, Geneva to NYC
Estimating search statistics

Given a set of items as a stream (e.g. queries on google.com over a period of time)

Geneva to NYC, coffee in Geneva, Geneva to NYC

Find the most frequent items in the set

Geneva to NYC, coffee in Geneva
Estimating search statistics

Given a set of items as a stream (e.g. queries on google.com over a period of time)

Geneva to NYC, coffee in Geneva, Geneva to NYC

Find the most frequent items in the set

Geneva to NYC, coffee in Geneva
Estimating search statistics

Given a set of items as a stream (e.g. queries on google.com over a period of time)

Geneva to NYC, coffee in Geneva, Geneva to NYC

Find the most frequent items in the set

Geneva to NYC, coffee in Geneva

<table>
<thead>
<tr>
<th>Solution</th>
<th>Trivial</th>
<th>This lecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td><code>hash&lt;string&gt; h;</code></td>
<td><code>COUNTSKETCH</code></td>
</tr>
<tr>
<td></td>
<td><code># of distinct items</code></td>
<td><code>O(log N)</code></td>
</tr>
</tbody>
</table>
### Streaming model

<table>
<thead>
<tr>
<th>Solution</th>
<th>Trivial</th>
<th>This lecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>hash&lt;string&gt; h; # of distinct items</td>
<td>COUNTSKETCH $O(\log N)$</td>
</tr>
</tbody>
</table>
### Streaming model

<table>
<thead>
<tr>
<th>Solution</th>
<th>Trivial</th>
<th>This lecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>hash&lt;string&gt; h;</td>
<td>COUNTSKETCH</td>
</tr>
<tr>
<td></td>
<td># of distinct items</td>
<td>$O(\log N)$</td>
</tr>
</tbody>
</table>

Are constants small?
Streaming model

<table>
<thead>
<tr>
<th>Solution</th>
<th>Trivial</th>
<th>This lecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>$\text{hash}\langle\text{string}\rangle \ h;$</td>
<td>$\text{COUNTSKETCH}$</td>
</tr>
<tr>
<td></td>
<td># of distinct items</td>
<td>$O(\log N)$</td>
</tr>
</tbody>
</table>

Are constants small?

HyperLogLog: estimate Shakespeare’s vocabulary using 128 bits of memory
Streaming model

Widely used in practice for **scalable data analytics**

most frequent searches on google.com over a time period

most frequent tweets
Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$
  
  Assume $N$ is known

- Output $k$ most frequent items
  
  (Heavy hitters)

- Small storage: will get $O(k \log N)$
  
  Much better than storing all items!
**Goal:** design a small space data structure

\[ \text{FINDTOP}(S, k): \text{returns top } k \text{ most frequent items seen so far} \]
Goal: design a small space data structure

FINDTOP(S, k): returns top k most frequent items seen so far

Useful to first design

POINTQUERY(S, i): processes stream, then for any query item i can return $f_i=$number of times item i appeared
Denote the number of times item $i$ appears in the stream by $f_i$ (frequency of $i$)

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Denote the number of times item $i$ appears in the stream by $f_i$ (frequency of $i$)

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$

**POINTQUERY**($S, i$) in space $O(k \log N)$?
Denote the number of times item $i$ appears in the stream by $f_i$ (frequency of $i$)

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$

**PointQuery**($S, i$) in space $O(k \log N)$?

Impossible in general...

Imagine a stream where all elements occur with about the same frequency
**FINDAPPROXTOP**(*S, k, ε*): returns set of *k* items such that 
\[ f_i \geq (1 - \varepsilon)f_k \] for all reported *i*

**APPROXPOINTQUERY**(*S, i, ε*): processes stream, then for any query item *i* can return approximation \( \hat{f}_i \in [f_i - \varepsilon f_k, f_i + \varepsilon f_k] \)
\textsc{FindApproxTop}(S, k, \varepsilon): \text{returns set of } k \text{ items such that } f_i \geq (1 - \varepsilon)f_k \text{ for all reported } i$

\textsc{ApproxPointQuery}(S, i, \varepsilon): \text{processes stream, then for any query item } i \text{ can return approximation } \hat{f}_i \in [f_i - \varepsilon f_k, f_i + \varepsilon f_k]$

In this lecture: find most frequent (head) items \textbf{if they contribute the bulk of the stream} under some measure
Observe a stream of updates, maintain small space data structure

**Task:** after observing the stream, given $i \in \{1, 2, \ldots, m\}$, compute estimate $\hat{f}_i$ of $f_i$
Observe a stream of updates, maintain small space data structure

**Task:** after observing the stream, given $i \in \{1, 2, \ldots, m\}$, compute estimate $\hat{f}_i$ of $f_i$

To be specified:

- space complexity?
- quality of approximation?
- success probability?
1. Finding top $k$ elements via \( \text{APPROXPOINTQUERY} \)

2. Basic version of \( \text{APPROXPOINTQUERY} \)

3. \( \text{APPROXPOINTQUERY} \) and the \( \text{COUNTSKETCH} \) algorithm
1. Finding top $k$ elements via \textit{(APPROX)POINTQUERY}

2. \textbf{Basic version of APPROXPOINTQUERY}

3. \textit{APPROXPOINTQUERY} and the \textit{COUNTSKETCH} algorithm
Assume elements are ordered by frequency: \( f_1 \geq f_2 \geq \ldots \geq f_m \)
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: \( f_1 \geq f_2 \geq \ldots \geq f_m \)
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: \( f_1 \geq f_2 \geq \ldots \geq f_m \)
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: \( f_1 \geq f_2 \geq \ldots \geq f_m \)

\[
\begin{array}{cccccccccccc}
1 & 4 & 6 & 1 & 2 & 10 & 1 & 5 & 1 & 5 & 2 & 2 & 3 & 3 & 3 & 9 & 8 & 7
\end{array}
\]
Assume elements are ordered by frequency: \( f_1 \geq f_2 \geq \ldots \geq f_m \)
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: \( f_1 \geq f_2 \geq \ldots \geq f_m \)
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$
Assume elements are ordered by frequency: \( f_1 \geq f_2 \geq \ldots \geq f_m \)
Basic estimate

Will design a basic estimate with $O(1)$ space complexity, analyze precision
Basic estimate

Will design a basic estimate with $O(1)$ space complexity, analyze precision

Choose a hash function $s : [m] \rightarrow \{-1, +1\}$ uniformly at random

**INITIALIZE**

$C \leftarrow 0$

**UPDATE**($C$, $i$)

$C \leftarrow C + s(i)$

Show that $C \cdot s(i)$ is close to $f_i$ with high probability?
Basic estimate

Will design a basic estimate with $O(1)$ space complexity, analyze precision

Choose a hash function $s: [m] \rightarrow \{-1, +1\}$ uniformly at random

**INITIALIZE**

\[
C \leftarrow 0
\]

**UPDATE** $(C, i)$

\[
C \leftarrow C + s(i)
\]

**for** every $p = 1, \ldots, N$ (every element in the stream)

**UPDATE** $(C, i_p)$

**end for**

Show that $C \cdot s(i)$ is close to $f$ with high probability?
Basic estimate

Will design a basic estimate with $O(1)$ space complexity, analyze precision

Choose a hash function $s : [m] \rightarrow \{-1, +1\}$ uniformly at random

\[
\text{INITIALIZE} \quad C \leftarrow 0
\]

\[
\text{UPDATE}(C, i) \quad C \leftarrow C + s(i)
\]

\[
\text{for every } p = 1, \ldots, N \text{ (every element in the stream)}
\]

\[
\text{UPDATE}(C, i_p)
\]

\[
\text{end for}
\]

\[
\text{ESTIMATE}(C, i) \quad \text{return } C \cdot s(i)
\]

Show that $C \cdot s(i)$ is close to $f_i$ ‘with high probability’?
\textbf{UPDATE}(C, i) \quad \text{ESTIMATE}(C, i)

\[ C \leftarrow C + s(i) \quad \text{return } C \cdot s(i) \]

Show that $C \cdot s(i)$ is close to $f_i$ ‘with high probability’?
Show that \( C \cdot s(i) \) is close to \( f_i \) ‘with high probability’?

Two steps:

- show that \( \mathbb{E}_s[C \cdot s(i)] = f_i \)
  
  (so \( C \cdot s(i) \) is an unbiased estimate of \( f_i \))

- show that \( \text{Var}_s[C \cdot s(i)] \) is ‘small’
Basic estimate: mean

**UPDATE**($C$, $i$)

$C \leftarrow C + s(i)$

**ESTIMATE**($C$, $i$)

return $C \cdot s(i)$

$$C \cdot s(i) = \sum_{p=1}^{N} s(i_p) s(i)$$

Our estimator is unbiased!

Is the estimate $C \cdot s(i)$ close to $f_i$ with high probability?
Basic estimate: mean

\[
\text{UPDATE}(C, i) \quad \text{UPDATE}(C, i)
\]
\[
C \leftarrow C + s(i) \quad \text{return } C \cdot s(i)
\]

\[
C \cdot s(i) = \sum_{p=1}^{N} s(i_p)s(i) = \sum_{j \in [m]} f_j \cdot s(j)s(i)
\]
Basic estimate: mean

**UPDATE**\((C, i)\)
\[
C \leftarrow C + s(i)
\]

**ESTIMATE**\((C, i)\)
\[
\text{return } C \cdot s(i)
\]

\[
C \cdot s(i) = \sum_{p=1}^{N} s(i_p)s(i) = \sum_{j \in [m]} f_j \cdot s(j)s(i)
\]
\[
= f_is(i)^2 + \sum_{j \in [m] \setminus i} f_j \cdot s(j)s(i)
\]

Our estimator is unbiased! Is the estimate \(C \cdot s(i)\) close to \(f_i\) with high probability?
Basic estimate: mean

\[
\text{UPDATE}(C, i) \quad C \leftarrow C + s(i) \\
\text{ESTIMATE}(C, i) \quad \text{return } C \cdot s(i)
\]

\[
C \cdot s(i) = \sum_{p=1}^{N} s(i_p)s(i) = \sum_{j \in [m]} f_j \cdot s(j)s(i) \\
= f_i s(i)^2 + \sum_{j \in [m] \setminus i} f_j \cdot s(j)s(i) \\
= f_i + \sum_{j \in [m] \setminus i} f_j \cdot s(j)s(i) \quad \text{← random } \pm 1's
\]
Basic estimate: mean

\[
\text{UPDATE}(C, i) \quad C \leftarrow C + s(i) \\
\text{ESTIMATE}(C, i) \quad \text{return } C \cdot s(i)
\]

\[
C \cdot s(i) = f_i + \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)
\]

The mean is correct: our estimator is unbiased! Is the estimate \(C \cdot s(i)\) close to \(f_i\) with high probability?
Basic estimate: mean

\begin{align*}
\text{UPDATE}(C, i) & \quad \text{ESTIMATE}(C, i) \\
C \leftarrow C + s(i) & \quad \text{return } C \cdot s(i)
\end{align*}

\[ E[C \cdot s(i)] = f_i + E[\sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)] \]
Basic estimate: mean

\[ \text{UPDATE}(C, i) \]
\[ C \leftarrow C + s(i) \]

\[ \text{ESTIMATE}(C, i) \]
\[ \text{return } C \cdot s(i) \]

\[
E[C \cdot s(i)] = f_i + E[ \sum_{j \in [m] \backslash i} f_j \cdot s(j) s(i)] \\
= f_i + \sum_{j \in [m] \backslash i} f_j \cdot E[s(j)] E[s(i)] \quad \text{(by independence of } s(i))
\]
Basic estimate: mean

\[ \text{UPDATE}(C, i) \quad C \leftarrow C + s(i) \]

\[ \text{ESTIMATE}(C, i) \quad \text{return } C \cdot s(i) \]

\[
\mathbb{E}[C \cdot s(i)] = f_i + \mathbb{E}\left[ \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i) \right] \\
= f_i + \sum_{j \in [m] \setminus i} f_j \cdot \mathbb{E}[s(j)] \mathbb{E}[s(i)] \quad \text{(by independence of } s(i)) \\
= f_i
\]
Basic estimate: mean

UPDATE(C, i)
\[ C \leftarrow C + s(i) \]

ESTIMATE(C, i)
\[ \text{return } C \cdot s(i) \]

\[
\mathbb{E}[C \cdot s(i)] = f_i + \mathbb{E}\left[ \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i) \right] \\
= f_i + \sum_{j \in [m] \setminus i} f_j \cdot \mathbb{E}[s(j)] \mathbb{E}[s(i)] \quad \text{(by independence of } s(i)) \\
= f_i
\]

The mean is correct: our estimator is unbiased!
Basic estimate: mean

\text{UPDATE}(C, i)
\quad C \leftarrow C + s(i) \quad \text{ESTIMATE}(C, i)
\quad \text{return } C \cdot s(i)

\[\mathbb{E}[C \cdot s(i)] = f_i + \mathbb{E}[ \sum_{j \in [m] \setminus i} f_j \cdot s(j)s(i)]\]

\[= f_i + \sum_{j \in [m] \setminus i} f_j \cdot \mathbb{E}[s(j)]\mathbb{E}[s(i)] \quad \text{(by independence of } s(i))\]

\[= f_i\]

The mean is correct: our estimator is unbiased!

Is the estimate $C \cdot s(i)$ close to $f_i$ with high probability?
Basic estimate: variance

\text{UPDATE}(C, i)
\begin{align*}
C &\leftarrow C + s(i) \\
\end{align*}

\text{ESTIMATE}(C, i)
\begin{align*}
&\quad \text{return } C \cdot s(i)
\end{align*}

We have
\[ C \cdot s(i) = f_i + \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i) \]

\text{and}
\[ \mathbb{E}[C \cdot s(i)] = f_i. \]
Basic estimate: variance

\[ \text{UPDATE}(C, i) \]
\[ C \leftarrow C + s(i) \]

\[ \text{ESTIMATE}(C, i) \]
\[ \text{return } C \cdot s(i) \]

We have

\[ C \cdot s(i) = f_i + \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i) \]

and

\[ \mathbb{E}[C \cdot s(i)] = f_i. \]

We need to bound

\[ \text{Var}(C \cdot s(i)) = \mathbb{E}[(C \cdot s(i) - \mathbb{E}[C \cdot s(i)])^2] \]
\[ = \mathbb{E}[(C \cdot s(i) - f_i)^2] \]
\[ = \mathbb{E} \left[ \left( \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i) \right)^2 \right] \]
Basic estimate: variance

\[ \text{UPDATE}(C, i) \]
\[ C \leftarrow C + s(i) \]

\[ \text{ESTIMATE}(C, i) \]
\[ \text{return } C \cdot s(i) \]

\[
(C \cdot s(i) - f_i)^2 = \left( \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i) \right)^2
\]
\[
= \sum_{j \in [m] \setminus i} \sum_{j' \in [m] \setminus i} f_j f_{j'} \cdot s(j) s(j') \cdot s^2(i)
\]
\[
= \sum_{j \in [m] \setminus i} \sum_{j' \in [m] \setminus i} f_j f_{j'} \cdot s(j) s(j')
\]
Basic estimate: variance

\begin{align*}
\text{UPDATE}(C, i) & \quad \text{ESTIMATE}(C, i) \\
C & \leftarrow C + s(i) & \quad \text{return } C \cdot s(i) \\
\end{align*}

\[
E[(C \cdot s(i) - f_i)^2] = E\left[ \sum_{j \in [m]} \sum_{j' \in [m]} f_j f_{j'} \cdot s(j) s(j') \right] \\
= \sum_{j \in [m]} \sum_{j' \in [m]} f_j f_{j'} \cdot E[s(j) s(j')] \\
= \sum_{j \in [m]} f_j^2
\]

since

- $s(j)^2 = 1$ for all $j$
- $E[s(j) s(j')] = E[s(j)] E[s(j')] = 0$ for $j \neq j'$. 

\[\]
Basic estimate: variance

\[ \text{UPDATE}(C, i) \]
\[ C \leftarrow C + s(i) \]

\[ \text{ESTIMATE}(C, i) \]
\[ \text{return } C \cdot s(i) \]

We have proved that
\[ \text{Var}(C \cdot s(i)) = E[(C \cdot s(i) - f_i)^2] = \sum_{j \in \{m\}} \text{if}^2 j \]

By Chebyshev's inequality
\[ \text{Pr} \left( \left| C \cdot s(i) - f_i \right| \geq 8 \cdot \sqrt{\sum_{j \in \{m\}} \text{if}^2 j} \right) \leq \frac{1}{64} \]

So \( C \cdot s(i) \) is close (?) to \( f_i \) with high probability.
Basic estimate: variance

\[ \text{UPDATE}(C, i) \]
\[ C \leftarrow C + s(i) \]

\[ \text{ESTIMATE}(C, i) \]
\[ \text{return } C \cdot s(i) \]

We have proved that

\[ \text{Var}(C \cdot s(i)) = E[(C \cdot s(i) - f_i)^2] = \sum_{j \in [m] \setminus i} f_j^2 \]
Basic estimate: variance

\[
\text{UPDATE}(C, i) \quad C \leftarrow C + s(i) \quad \text{ESTIMATE}(C, i) \quad \text{return } C \cdot s(i)
\]

We have proved that

\[
\text{Var}(C \cdot s(i)) = \mathbb{E}[(C \cdot s(i) - f_i)^2] = \sum_{j \in [m] \setminus i} f_j^2
\]

By Chebyshev’s inequality

\[
\Pr \left[ |C \cdot s(i) - f_i| \geq 8 \cdot \sqrt{\sum_{j \in [m] \setminus i} f_j^2} \right] \leq 1/64
\]
Basic estimate: variance

```
UPDATE(C, i)
C ← C + s(i)

ESTIMATE(C, i)
return C · s(i)
```

We have proved that

\[
\text{Var}(C \cdot s(i)) = \mathbb{E}[(C \cdot s(i) - f_i)^2] = \sum_{j \in [m] \setminus i} f_j^2
\]

By Chebyshev’s inequality

\[
\Pr \left[ |C \cdot s(i) - f_i| \geq 8 \cdot \sqrt{\sum_{j \in [m] \setminus i} f_j^2} \right] \leq 1/64
\]

So \( C \cdot s(i) \) is close (?) to \( f_i \) with high probability
Basic estimate: variance

**UPDATE**\((C, i)\)
\[ C \leftarrow C + s(i) \]

**ESTIMATE**\((C, i)\)
\[ \text{return } C \cdot s(i) \]

We have proved that
\[
\text{Var}(C \cdot s(i)) = \mathbb{E}[(C \cdot s(i) - f_i)^2] = \sum_{j \in [m] \setminus i} f_j^2
\]

By Chebyshev’s inequality
\[
\text{Pr}\left[|C \cdot s(i) - f_i| > 8 \cdot \sqrt{\sum_{j \in [m] \setminus i} f_j^2}\right] \leq 1/64
\]

So \(C \cdot s(i)\) is close (?) to \(f_i\) with high probability
Basic estimate: summary

**UPDATE**(C, i)
\[ C \leftarrow C + s(i) \]

**ESTIMATE**(C, i)
\[ \text{return } C \cdot s(i) \]

Estimate \( f_i \) up to

\[ 8 \cdot \sqrt{\sum_{j \in [m] \backslash i} f_j^2} \]

item to be estimated

**Pro:** works well for most frequent item, if other items are small
**Basic estimate: summary**

**UPDATE(C, i)**

\[ C \leftarrow C + s(i) \]

**ESTIMATE(C, i)**

\[ \text{return } C \cdot s(i) \]

Estimate \( f_i \) up to

\[ 8 \cdot \sqrt{\sum_{j \in [m] \setminus i} f_j^2} \]

Pro: works well for most frequent item, if other items are small

Con: estimate for a small items contaminated by large items
1. Finding top $k$ elements via (APPROX)POINTER\textsc{Query}

2. Basic version of APPROXPOINTER\textsc{Query}

3. APPROXPOINTER\textsc{Query} and the COUNTSKETCH algorithm
1. Finding top $k$ elements via \textsc{(APPROX)POINTQUERY}

2. Basic version of \textsc{APPROXPOINTQUERY}

3. \textsc{APPROXPOINTQUERY} and the \textsc{COUNTSKETCH} algorithm
ApproxPointQuery and CountSketch

CountSketch algorithm (Charikar, Chen, Farach-Colton’02)

Main ideas:

1. run basic estimate on subsampled[hashed] stream
   (reduces variance)
APPROXPOINTQUERY and COUNTSKETCH

COUNTSKETCH algorithm (Charikar, Chen, Farach-Colton’02)

Main ideas:

1. run basic estimate on subsampled/hashed stream (reduces variance)

2. aggregate independent estimates to boost confidence (take medians)
**APPROXPOINTQUERY and COUNTSKETCH**

**COUNTSKETCH algorithm** ([Charikar, Chen, Farach-Colton’02](#))

Main ideas:

1. run basic estimate on *subsampledhashed* stream (reduces variance)

2. aggregate independent estimates to boost confidence (take *medians*)

```
universe [m] 1 2 3 4 5 6 7 8 9 10

buckets [B]
```
Hashing the items

Hashed into $B = 8$ buckets, get 8 subsampled streams

For item $i$ its stream consists of $j \in [m]$ such that $h(j) = h(i)$
Hashing the items

Hashed into $B = 8$ buckets, get 8 subsampled streams

For item $i$ its stream consists of $j \in [m]$ such that $h(j) = h(i)$

For example,

- subsampled stream of item 1 is $\{1, 6\}$
Hashing the items

Hashed into $B = 8$ buckets, get 8 subsampled streams

For item $i$ its stream consists of $j \in [m]$ such that $h(j) = h(i)$

For example,

- subsampled stream of item 1 is $\{1, 6\}$
- subsampled stream of item 5 is $\{5, 7\}$
Note: hashing the universe $[m]$, not positions in the stream
Note: hashing the universe $[m]$, not positions in the stream

E.x. the subsampled stream of item 1 is \{1, 6\}
Note: hashing the universe \([m]\), not positions in the stream
Note: hashing the universe \([m]\), not positions in the stream

E.x. the subsampled stream of item 5 is \(\{5, 7\}\)
Final ApproxPointQuery

Choose

- \( t \) random hash functions \( h_1, h_2, \ldots, h_t \) from items \([m]\) to \( B \approx k \) buckets \( \{1, 2, \ldots, B\} \)

- \( t \) random hash functions \( s_1, s_2, \ldots, s_t \) from items \([m]\) to \( \{-1, +1\} \)

\[ \text{array } C \]
Final ApproxPointQuery

Choose

- \( t \) random hash functions \( h_1, h_2, \ldots, h_t \) from items \([m]\) to \( B \approx k \) buckets \( \{1, 2, \ldots, B\} \)

- \( t \) random hash functions \( s_1, s_2, \ldots, s_t \) from items \([m]\) to \( \{-1, +1\} \)

The algorithm runs \( t \) independent copies of basic estimate:

\[
\text{UPDATE}(C, i) \\
\text{for } r \in [1 : t] \\
C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \\
\text{end for} \\
\text{ESTIMATE}(C, i) \\
\text{return } \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
\]
Final ApproxPointQuery

Choose

- $t$ random hash functions $h_1, h_2, \ldots, h_t$ from items $[m]$ to $B \approx k$ buckets $\{1, 2, \ldots, B\}$

- $t$ random hash functions $s_1, s_2, \ldots, s_t$ from items $[m]$ to $\{-1, +1\}$

The algorithm runs $t$ independent copies of basic estimate:

\[
\text{UPDATE}(C, i) \\
\text{for } r \in [1 : t] \\
\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \\
\text{end for} \\
\text{ESTIMATE}(C, i) \\
\quad \text{return median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
\]
\textbf{UPDATE}(C, i)
for \( r \in [1 : t] \)
\hspace{1em} \( C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \)
end for

\textbf{ESTIMATE}(C, i)
\hspace{1em} return \ \text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \}

Lemma
If \( B \geq 8 \max \{ k, 32 \sum_{j \in \text{TAIL}} f_j^2(\varepsilon f_k^2) \} \) and \( t = O(\log N) \), then for every \( i \in [m] \)
\hspace{1em} \(| \text{ESTIMATE}(C, i) - f_i | \leq \varepsilon f_k \)
at every point in the stream whp.

Space complexity is \( O(B \log N) \).

How large is \( B \)?
UPDATE(C, i)
for $r \in [1 : t]$
\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]
end for

ESTIMATE(C, i)
return median$_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Lemma
If $B \geq 8 \max \left\{ k, \frac{32 \sum_{j \in \text{TAIL}} f_j^2}{(\varepsilon f_k)^2} \right\}$ and $t = O(\log N)$, then for every $i \in [m]$
\[ |\text{ESTIMATE}(C, i) - f_i| \leq \varepsilon f_k \]
at every point in the stream whp.
**UPDATE(C, i)**

```plaintext
for r ∈ [1 : t]
    C[r, h_r(i)] ← C[r, h_r(i)] + s_r(i)
end for
```

**ESTIMATE(C, i)**

```plaintext
return median_r \{ C[r, h_r(i)] \cdot s_r(i) \}
```

**Lemma**

If \( B \geq 8 \max \left\{ k, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_k)^2} \right\} \) and \( t = O(\log N) \), then for every \( i \in [m] \)

\[
|ESTIMATE(C, i) - f_i| \leq \varepsilon f_k
\]

at every point in the stream whp.

Space complexity is \( O(B \log N) \)
UPDATE(C, i)
for $r \in [1 : t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Lemma
If $B \geq 8 \max \left\{ k, \frac{32 \sum_{j \in \text{TAIL}} f_j^2}{(\varepsilon f_k)^2} \right\}$ and $t = O(\log N)$, then for every $i \in [m]$

$|\text{ESTIMATE}(C, i) - f_i| \leq \varepsilon f_k$

at every point in the stream whp.

Space complexity is $O(B \log N)$

How large is $B$?
Space complexity

Set $B = 8 \max \left\{ k, \frac{32 \sum_{j \in \text{TAIL}} f_j^2}{(\varepsilon f_k)^2} \right\}$

Note that $B = O(k/\varepsilon^2)$ if $\frac{1}{k} \sum_{j \in \text{TAIL}} f_j^2 = O(f_k^2)$

Note: if $B \geq k$, can detect elements with counts above $O\left(\sqrt{\frac{1}{B} \cdot \sum_{j \in \text{TAIL}} f_j^2}\right)$
Space complexity

Set $k = 1$. Suppose that $1$ appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct.
Set $k = 1$. Suppose that $1$ appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct.

Then $f_1 = \sqrt{N}$, $f_i = 1$ for $i = 2, N - \sqrt{N}$.

Set $B = 8 \max \left\{ 1, \frac{32 \sum_{j \in \text{TAIL}} f_j^2}{(\varepsilon f_1)^2} \right\}$.
Space complexity

Set $k = 1$. Suppose that $1$ appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct.

Then $f_1 = \sqrt{N}$, $f_i = 1$ for $i = 2, N - \sqrt{N}$.

Set $B = 8 \max \left\{ 1, \frac{32 \sum_{j \in \text{TAIL}} f_j^2}{(\varepsilon f_1)^2} \right\}$

We have $\sum_{j \in \text{TAIL}} f_j^2 = N - \sqrt{N} \leq N$, and $f_1^2 = N$. 

Space complexity

Set $k = 1$. Suppose that $1$ appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct.

Then $f_1 = \sqrt{N}$, $f_i = 1$ for $i = 2, N - \sqrt{N}$.

Set $B = 8 \max \left\{ 1, \frac{32 \sum_{j \in \text{TAIL}} f_j^2}{(\varepsilon f_1)^2} \right\}$

We have $\sum_{j \in \text{TAIL}} f_j^2 = N - \sqrt{N} \leq N$, and $f_1^2 = N$.

So $B = 8 \max \left\{ 1, \frac{32 \sum_{j \in \text{TAIL}} f_j^2}{(\varepsilon f_1)^2} \right\} = O(1/\varepsilon^2)$ suffices.
Space complexity

Set $k = 1$. Suppose that $1$ appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct.

Then $f_1 = \sqrt{N}$, $f_i = 1$ for $i = 2, N - \sqrt{N}$.

Set $B = 8 \max \left\{ 1, \frac{32 \sum j \in \text{TAIL} f_j^2}{(\varepsilon f_1)^2} \right\}$

We have $\sum j \in \text{TAIL} f_j^2 = N - \sqrt{N} \leq N$, and $f_1^2 = N$.

So $B = 8 \max \left\{ 1, \frac{32 \sum j \in \text{TAIL} f_j^2}{(\varepsilon f_1)^2} \right\} = O(1/\varepsilon^2)$ suffices.

Remarkable, as $1$ appears only in $\sqrt{N}$ positions out of $N$: a vanishingly small fraction of positions!
**Lemma**

If $B \geq 8 \max \left\{ k, \frac{32 \sum_{j \in \text{TAIL}} f_j^2}{(\varepsilon f_k)^2} \right\}$ and $t \geq A \log N$ for an absolute constant $A > 0$, then for every $i \in [m]$

$$|\text{ESTIMATE}(C, i) - f_i| \leq \varepsilon f_k$$

with high probability.

($f_i$ is the frequency of $i$)
**UPDATE**($C$, $i$)

for $r \in [1 : t]$

\[
C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)
\]

end for

**ESTIMATE**($C$, $i$)

return \(\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}\)

Variance of estimate for $i$ from $r$-th row:

\[
\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2
\]
UPDATE(C, i)
for $r \in [1 : t]$
\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]
end for

ESTIMATE(C, i)
return median$_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Variance of estimate for $i$ from $r$-th row:
\[ \sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 \]

Show that
\[ \sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = O(1/B) \sum_{j \in TAIL, j \neq i} f_j^2 \]
with high constant probability.
Consider contribution of head and tail items separately:

\[
\sum_{j \neq i: h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i} f_j^2
\]
Consider contribution of head and tail items separately:

\[
\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i} f_j^2
\]

For each \( r \in [1 : t] \) and each item \( i \in [m] \) define three events:

- **NO-COLLISIONS}_r(i) – \( i \) does not collide with any of the head items under hashing \( r \)
Consider contribution of head and tail items separately:

\[
\sum_{j \neq i: h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i} f_j^2
\]

For each \( r \in [1 : t] \) and each item \( i \in [m] \) define three events:

- **NO-COLLISIONS}_r(i) – \( i \) does not collide with any of the head items under hashing \( r \)

- **SMALL-VARIANCE}_r(i) – \( i \) does not collide with too many of tail items under hashing \( r \)
Consider contribution of head and tail items separately:

\[
\sum_{j \neq i: h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i \atop h_r(j) = h_r(i)} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i \atop h_r(j) = h_r(i)} f_j^2
\]

For each \( r \in [1 : t] \) and each item \( i \in [m] \) define three events:

- **NO-COLLISIONS}_r(i) – \( i \) does not collide with any of the head items under hashing \( r \)

- **SMALL-VARIANCE}_r(i) – \( i \) does not collide with too many of tail items under hashing \( r \)

- **SMALL-DEVIATION}_r(i) – success event from basic analysis
Consider contribution of head and tail items separately:

$$
\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i} f_j^2
$$

For each $r \in [1 : t]$ and each item $i \in [m]$ define three events:

- **NO-COLLISIONS$_r(i)$** – $i$ does not collide with any of the head items under hashing $r$

- **SMALL-VARIANCE$_r(i)$** – $i$ does not collide with too many of tail items under hashing $r$

- **SMALL-DEVIATION$_r(i)$** – success event from basic analysis

Show that all three events hold simultaneously with probability strictly bigger than $1/2$ – so median gives good estimate
(No) collisions with head items

\text{NO-COLLISIONS}_r(i) := \text{event that}

\{j \in HEAD \setminus i : h_r(j) = h_r(i)\} = \emptyset,

i.e. that \(i\) collides with none of top \(k\) elements under \(h_r\).
(No) collisions with head items

\textsc{No-Collisions}_r(i) := \text{event that}

\[ \{ j \in \text{HEAD} \setminus i : h_r(j) = h_r(i) \} = \emptyset, \]

i.e. that \( i \) collides with none of top \( k \) elements under \( h_r \).

For every \( j \neq i \) and every \( r \in [1 : t] \)

\[ \Pr[h_r(i) = h_r(j)] \leq 1/B \]
(No) collisions with head items

\textbf{NO-COLLISIONS}_r(i) := \text{event that}

\[ \{ j \in \text{HEAD} \setminus i : h_r(j) = h_r(i) \} = \emptyset, \]

i.e. that \( i \) collides with none of top \( k \) elements under \( h_r \).

For every \( j \neq i \) and every \( r \in [1 : t] \)

\[ \Pr[h_r(i) = h_r(j)] \leq 1 / B \]

Suppose that \( B \geq 8k \). Then by the union bound

\[ \Pr[\text{NO-COLLISIONS}_r(i)] \geq 1 - k / B \]

\[ \geq 1 - 1 / 8 \]
Consider contribution of head and tail items separately:

\[
\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i : h_r(j) = h_r(i)} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i : h_r(j) = h_r(i)} f_j^2
\]

For each \( r \in [1 : t] \) and each item \( i \in [m] \) define three events:

- **NO-COLLISIONS\(_r\)(i)** – \( i \) does not collide with any of the head items under hashing \( r \)

- **SMALL-VARIANCE\(_r\)(i)** – \( i \) does not collide with too many of tail items under hashing \( r \)

- **SMALL-DEVIATION\(_r\)(i)** – success event from basic analysis

Show that all three events hold simultaneously with probability strictly bigger than 1/2 – so median gives good estimate
Consider contribution of head and tail items separately:

\[
\sum_{j \neq i: h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i, h_r(j) = h_r(i)} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i, h_r(j) = h_r(i)} f_j^2
\]

For each \( r \in [1 : t] \) and each item \( i \in [m] \) define three events:

- **No-Collisions\(_r(i)\)** – \( i \) does not collide with any of the head items under hashing \( r \)
- **Small-Variance\(_r(i)\)** – \( i \) does not collide with too many of tail items under hashing \( r \)
- **Small-Deviation\(_r(i)\)** – success event from basic analysis

Show that all three events hold simultaneously with probability strictly bigger than \( 1/2 \) – so median gives good estimate
Small variance from tail elements

\[ \text{SMALL-VARIANCE}_r(i) := \text{event that} \]

\[ \sum_{j \in \text{TAIL}, j \neq i} f_j^2 \leq \frac{8}{B} \sum_{j \in \text{TAIL}} f_j^2 \]

For every \( i, j \in \mathbb{N} \), \( i \neq j \) and \( r \in [1:t] \)

\[ \Pr[h_r[i] = h_r[j]] = \frac{1}{B} \]

So by linearity of expectation

\[ \mathbb{E} \left[ \sum_{j \in \text{TAIL}, j \neq i} h_r[j] = h_r[i] \right] \leq \frac{1}{B} \sum_{j \in \text{TAIL}} f_j^2 \]
Small variance from tail elements

**SMALL-VARIANCE}_r(i):=event that**

\[ \sum_{j \in TAIL, j \neq i \atop h_r(j) = h_r(i)} f_j^2 \leq \frac{8}{B} \sum_{j \in TAIL} f_j^2 \]

For every \( i, j \in [m], i \neq j \) and \( r \in [1 : t] \)

\[ \Pr_{h_r}[h_r(i) = h_r(j)] = 1 / B \quad (B \text{ is the number of buckets}) \]
Small variance from tail elements

$\text{SMALL-VARIANCE}_r(i) := \text{event that}$

$$\sum_{j \in \text{TAIL}, j \neq i \atop h_r(j) = h_r(i)} f_j^2 \leq \frac{8}{B} \sum_{j \in \text{TAIL}} f_j^2$$

For every $i, j \in [m], i \neq j$ and $r \in [1 : t]$

$$\Pr_{h_r}[h_r(i) = h_r(j)] = 1 / B \quad (B \text{ is the number of buckets})$$

So by linearity of expectation

$$E \left[ \sum_{j \in \text{TAIL}, j \neq i \atop h_r(j) = h_r(i)} f_j^2 \right] = \sum_{j \in \text{TAIL}, j \neq i} f_j^2 \cdot \Pr_{h_r}[h_r(i) = h_r(j)]$$
Small variance from tail elements

\[ \text{SMALL-VARIANCE}_r(i) := \text{event that} \]

\[
\sum_{j \in \text{TAIL}, j \neq i \atop h_r(j) = h_r(i)} f_j^2 \leq \frac{8}{B} \sum_{j \in \text{TAIL}} f_j^2
\]

For every \( i, j \in [m], i \neq j \) and \( r \in [1 : t] \)

\[
\Pr_{h_r}[h_r(i) = h_r(j)] = \frac{1}{B} \quad (B \text{ is the number of buckets})
\]

So by linearity of expectation

\[
\mathbb{E} \left[ \sum_{j \in \text{TAIL}, j \neq i \atop h_r(j) = h_r(i)} f_j^2 \right] = \sum_{j \in \text{TAIL}, j \neq i} f_j^2 \cdot \Pr_{h_r}[h_r(i) = h_r(j)]
\]

\[
\leq \frac{1}{B} \sum_{j \in \text{TAIL}} f_j^2
\]
We proved that

\[
E \left[ \sum_{j \in \text{TAIL}, j \neq i \atop h_r(j) = h_r(i)} f_j^2 \right] \leq \frac{1}{B} \sum_{j \in \text{TAIL}} f_j^2
\]

By Markov’s inequality one has, for every \( i \) and every \( r \),

\[
\Pr[\text{SMALL-VARIANCE}_r(i)] \geq 1 - 1/8
\]
**NO-COLLISIONS}_r(i) and SMALL-VARIANCE}_r(i): recap**

Consider contribution of head and tail items separately:

\[
\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i, h_r(j) = h_r(i)} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i, h_r(j) = h_r(i)} f_j^2
\]

Conditioned on NO-COLLISIONS}_r(i) and SMALL-VARIANCE}_r(i)
Consider contribution of head and tail items separately:

\[
\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i \atop h_r(j) = h_r(i)} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i \atop h_r(j) = h_r(i)} f_j^2
\]

Conditioned on \(\text{NO-COLLISIONS}_r(i)\) and \(\text{SMALL-VARIANCE}_r(i)\):

- first term is zero
NO-COLLISIONS$_r(i)$ and SMALL-VARIANCE$_r(i)$: recap

Consider contribution of head and tail items separately:

\[
\sum_{j \neq i: h_r(j) = h_r(i)} f_j^2 = \sum_{j \in \text{HEAD}, j \neq i \text{ and } h_r(j) = h_r(i)} f_j^2 + \sum_{j \in \text{TAIL}, j \neq i \text{ and } h_r(j) = h_r(i)} f_j^2
\]

Conditioned on NO-COLLISIONS$_r(i)$ and SMALL-VARIANCE$_r(i)$

- first term is zero
- second term is at most

\[
\frac{8}{B} \sum_{j \in \text{TAIL}} f_j^2
\]
Small deviation event

\[ \text{SMALL-DEVIATION}_r(i) = \text{event that} \]

\[ (C[r, h_r(i)] \cdot s_r(i) - f_i)^2 \leq 8 \text{Var}(C[r, h_r(i)] \cdot s_r(i)). \]
**Small deviation event**

**\textsc{Small-Deviation}_r(i) =** event that

\[(C[r, h_r(i)] \cdot s_r(i) - f_i)^2 \leq 8 \text{Var}(C[r, h_r(i)] \cdot s_r(i)).\]

By Markov’s inequality one has, for every \(i\) and every \(r\),

\[\Pr[\textsc{Small-Deviation}_r(i)] \geq 1 - 1/8\]
\[ \Pr[\text{SMALL-VARIANCE}_r(i)] \geq 1 - 1/8 \]

\[ \Pr[\text{NO-COLLISIONS}_r(i)] \geq 1 - 1/8 \]

\[ \Pr[\text{SMALL-DEVIATION}_r(i)] \geq 1 - 1/8 \]

So by the union bound

\[ \Pr[\text{SMALL-VARIANCE}_r(i) \text{ and NO-COLLISIONS}_r(i) \text{ and SMALL-DEVIATION}_r(i)] \geq 5/8. \]
For every $p \in [1 : N]$ let $f_i(p) := \text{frequency of } i \text{ up to position } p$

**Lemma**

If $B \geq 8 \max\left\{ k, \frac{32 \sum_{j \in \text{TAL}} f_j^2}{(\varepsilon f_k)^2} \right\}$ and $t \geq A \log N$ for an absolute constant $A > 0$, then with probability $\geq 1 - 1/N^3$ for every $i \in [m]$

$$|\text{ESTIMATE}(C, i) - f_i(p)| \leq \varepsilon f_k$$

at the end of the stream.
Remarks, related results, open problems
\textbf{UPDATE}(C, i) \\
\textbf{for} \ r \in [1 : t] \\
\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \\
\textbf{end for} \\
\textbf{ESTIMATE}(C, i) \\
\textbf{return} \ \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
**UPDATE**($C$, $i$)

**for** $r \in [1 : t]$

\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]

**end for**

**ESTIMATE**($C$, $i$)

**return** $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$
UPDATE($C, i$)
for $r \in [1 : t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE($C, i$)
return median$_r \{C[r, h_r(i)] \cdot s_r(i)\}$
\textbf{UPDATE}(C, i)
\begin{align*}
\text{for } r \in [1 : t] \\
    C[r, h_r(i)] &\leftarrow C[r, h_r(i)] + s_r(i) \\
\end{align*}
\textbf{end for}

\textbf{ESTIMATE}(C, i)
\begin{align*}
\text{return } \text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \}
\end{align*}
\textbf{UPDATE}(C, i)

\textbf{for } r \in [1: t] \\
\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)
\textbf{end for}

\textbf{ESTIMATE}(C, i)

\textbf{return } \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
**UPDATE**(C, i)

for \( r \in [1 : t] \)

\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]

end for

**ESTIMATE**(C, i)

return \( \text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \} \)
UPDATE(C, i)
for $r \in [1 : t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \}$
**UPDATE**\((C, i)\)

```latex
\textbf{for } r \in [1 : t] \\
\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)
```

**ESTIMATE**\((C, i)\)

```latex
\textbf{return } \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
```
**UPDATE**($C$, $i$)

for $r \in [1 : t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

**ESTIMATE**($C$, $i$)

return $\text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \}$

---

### Diagram

![Diagram showing the steps of the algorithm](image-url)

---

### List

1 4 6 1 2 10 1 5
\textbf{UPDATE}(C, i)

\textbf{for } r \in [1 : t] \\
\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)

\textbf{end for}

\textbf{ESTIMATE}(C, i)

\textbf{return } \text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \}
\textbf{UPDATE}(C, i)
\begin{align*}
\text{for } r & \in [1 : t] \\
&C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)
\end{align*}
\textbf{end for}

\textbf{ESTIMATE}(C, i)
\begin{align*}
\text{return } \text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \}
\end{align*}
**UPDATE(C, i)**

```plaintext
for r ∈ [1 : t]
    C[r, h_r(i)] ← C[r, h_r(i)] + s_r(i)
end for
```

**ESTIMATE(C, i)**

```plaintext
return \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
```
**UPDATE**($C, i$)

for $r \in [1 : t]$

$$C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$$

end for

**ESTIMATE**($C, i$)

return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$
UPDATE(C, i) for $r \in [1 : t]$
\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]
end for

ESTIMATE(C, i)
return median_r \{ C[r, h_r(i)] \cdot s_r(i) \}
\textbf{UPDATE}(C, i)
\begin{align*}
\text{for } r &\in [1 : t] \\
C[r, h_r(i)] &\leftarrow C[r, h_r(i)] + s_r(i)
\end{align*}
\textbf{end for}

\textbf{ESTIMATE}(C, i)
\begin{align*}
\text{return } \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
\end{align*}
UPDATE($C, i$)

for $r \in [1 : t]$

\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]

end for

ESTIMATE($C, i$)

return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$
UPDATE(C, i)
for $r \in [1 : t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$
UPDATE(C, i)
for r ∈ [1 : t]
    C[r, h_r(i)] ← C[r, h_r(i)]−s_r(i)
end for

ESTIMATE(C, i)
return median_r \{C[r, h_r(i)] \cdot s_r(i)\}
UPDATE(C, i)

for $r \in [1 : t]$

\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]

end for

ESTIMATE(C, i)

return median

\[ \left\{ C[r, h_r(i)] \cdot s_r(i) \right\} \]
\textbf{UPDATE}(\mathbf{C}, \ i)\\
\textbf{for} \ r \in [1: t]\\
\quad \mathbf{C}[r, h_r(i)] \leftarrow \mathbf{C}[r, h_r(i)] + s_r(i)\\
\textbf{end for}\\

\textbf{ESTIMATE}(\mathbf{C}, \ i)\\
\textbf{return} \ \text{median}_r \{\mathbf{C}[r, h_r(i)] \cdot s_r(i)\}\\
UPDATE(C, i)
for $r \in [1 : t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return median$_r \{C[r, h_r(i)] \cdot s_r(i)\}$
**UPDATE**(C, i)

for $r \in [1 : t]$

\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]

end for

**ESTIMATE**(C, i)

return $\text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \}$
UPDATE(C, i)
for $r \in [1 : t]$
\[ C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i) \]
end for

ESTIMATE(C, i)
return $\text{median}_r \{ C[r, h_r(i)] \cdot s_r(i) \}$
\textbf{UPDATE}(C, i)

\textbf{for } r \in [1 : t]
\hspace{1cm} C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)
\textbf{end for}

\textbf{ESTIMATE}(C, i)
\hspace{1cm} \textbf{return} \ \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
\begin{align*}
\text{UPDATE}(C, i) \\
\text{for } r \in [1 : t] \\
\quad C[r, h_r(i)] &\leftarrow C[r, h_r(i)] + s_r(i) \\
\text{end for} \\
\text{ESTIMATE}(C, i) \\
\text{return } \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
\end{align*}
UPDATE(C, i)
for $r \in [1 : t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return median$_r \{C[r, h_r(i)] \cdot s_r(i)\}$
UPDATE(C, i)
for $r \in [1 : t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return median$_r \{ C[r, h_r(i)] \cdot s_r(i) \}$
\text{UPDATE}(C, i)\
\text{for } r \in [1 : t]\
\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)\
\text{end for}\

\text{ESTIMATE}(C, i)\
\text{return } \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}
UPDATE($C$, $i$)
for $r \in [1 : t]$
\[
C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)
\]
end for

ESTIMATE($C$, $i$)
return median$_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Sketching: take (randomized) linear measurements of the input
\[
S \cdot x = b
\]
sketching matrix

space = number of rows

Easy to maintain sketch in dynamic streams (insertions and deletions)
Sparse recovery

Let $S$ be a COUNTSKETCH matrix with $O(k \log n)$ rows

**Lemma**

For every $x \in \mathbb{R}^n$ if $\hat{x} = \text{EST}(Sx)$, then whp

$$\|x - \hat{x}\|_\infty \leq \frac{1}{\sqrt{k}} \|x_{TAIL}\|_2.$$  

($x_{TAIL} - x$ with largest $k$ elements zeroed out)
Sparse recovery

Let $S$ be a COUNTSKETCH matrix with $O(k \log n)$ rows.

Lemma

For every $x \in \mathbb{R}^n$ if $\hat{x} = \text{EST}(Sx)$, then whp

$$\|x - \hat{x}\|_\infty \leq \frac{1}{\sqrt{k}} \|x_{\text{TAIL}}\|_2.$$

($x_{\text{TAIL}} = x$ with largest $k$ elements zeroed out)
Sparse recovery

Let $S$ be a COUNTSKETCH matrix with $O(k \log n)$ rows

Lemma

For every $x \in \mathbb{R}^n$ if $\hat{x} = \text{EST}(Sx)$, then whp

$$
\| x - \hat{x} \|_\infty \leq \frac{1}{\sqrt{k}} \| x^{\text{TAIL}} \|_2.
$$

($x^{\text{TAIL}} = x$ with largest $k$ elements zeroed out)

Observation 1: # of measurements is optimal for $\ell_\infty/\ell_2$ guarantee above

(capacity of the Gaussian channel, i.e. Shannon-Hartley theorem, – see Do Ba, Indyk, Price, Woodruff’10)
Sparse recovery

Let $S$ be a \textsc{CountSketch} matrix with $O(k \log n)$ rows.

**Lemma**

For every $x \in \mathbb{R}^n$ if $\hat{x} = \text{EST}(Sx)$, then whp

$$\|x - \hat{x}\|_\infty \leq \frac{1}{\sqrt{k}} \|x_{\text{TAIL}}\|_2.$$

($x_{\text{TAIL}} - x$ with largest $k$ elements zeroed out)

**Observation 1:** # of measurements is optimal for $\ell_\infty/\ell_2$ guarantee above

(capacity of the Gaussian channel, i.e. Shannon-Hartley theorem, – see Do Ba, Indyk, Price, Woodruff’10)

**Observation 2:** $\ell_2/\ell_2$ sparse recovery guarantee follows:

$$\|x - \hat{x}\|_2 = O(1) \cdot \|x_{\text{TAIL}}\|_2.$$
Sparse recovery
Let $S$ be a COUNTSKETCH matrix with $O(k \log n)$ rows

**Lemma**
*For every $x \in \mathbb{R}^n$ if $\hat{x} = \text{EST}(Sx)$, then whp*

$$
\| x - \hat{x} \|_\infty \leq \frac{1}{\sqrt{k}} \| x_{\text{T A I L}} \|_2.
$$

(*$x_{\text{T A I L}}$ – $x$ with largest $k$ elements zeroed out)*)

**Observation 1:** # of measurements is optimal for $\ell_\infty/\ell_2$ guarantee above

(capacity of the Gaussian channel, i.e. Shannon-Hartley theorem, – see Do Ba, Indyk, Price, Woodruff’10)
Sparse recovery

Let $S$ be a COUNTSKETCH matrix with $O(k \log n)$ rows.

Lemma

For every $x \in \mathbb{R}^n$ if $\hat{x} = \text{EST}(Sx)$, then whp

$$\|x - \hat{x}\|_\infty \leq \frac{1}{\sqrt{k}} \|x_{\text{TAIL}}\|_2.$$  

($x_{\text{TAIL}} - x$ with largest $k$ elements zeroed out)

Observation 1: # of measurements is optimal for $\ell_\infty/\ell_2$ guarantee above (capacity of the Gaussian channel, i.e. Shannon-Hartley theorem, – see Do Ba, Indyk, Price, Woodruff’10)

Observation 2: $\ell_2/\ell_2$ sparse recovery guarantee follows:

$$\|x - \hat{x}\|_2 = O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2.$$
Sparse recovery

Let $S$ be a COUNTSKETCH matrix with $O\left(\frac{1}{\epsilon^2} k \log n\right)$ rows

Lemma

For every $x \in \mathbb{R}^n$ if $\hat{x} = \text{EST}(Sx)$, then whp

$$\|x - \hat{x}\|_\infty \leq \frac{1}{\sqrt{k}} \|x_{\text{TAIL}}\|_2.$$ 

($x_{\text{TAIL}} - x$ with largest $k$ elements zeroed out)

Observation 1: # of measurements is optimal for $\ell_\infty/\ell_2$ guarantee above

(capacity of the Gaussian channel, i.e. Shannon-Hartley theorem, – see Do Ba, Indyk, Price, Woodruff’10)

Observation 2: $\ell_2/\ell_2$ sparse recovery guarantee follows:

$$\|x - \hat{x}\|_2 = (1 + \epsilon) \cdot \min_{k-\text{sparse } x'} \|x - x'\|_2.$$
Sparse recovery

Let $S$ be a COUNTSKETCH matrix with $O(k \log n)$ rows

**Lemma**

*For every $x \in \mathbb{R}^n$ if $\hat{x} = \text{EST}(Sx)$, then whp*

$$
\| x - \hat{x} \|_\infty \leq \frac{1}{\sqrt{k}} \| x_{\text{TAIL}} \|_2.
$$

($x_{\text{TAIL}} = x$ with largest $k$ elements zeroed out)

**Observation 3:** if $\| x \|_0 \leq k$, then $x_{\text{TAIL}} = 0$ and $\text{EST}(Sx) = x$ whp

Exact sparse recovery: a $k$-sparse vector can be recovered from $O(k)$ linear measurements
CountSketch for matrices?

Input: $r$ parties hold vectors $x_1, \ldots, x_r \in \mathbb{R}^n$

each party sends $O(B \log n)$ bits to coordinator

(assume shared randomness)
CountSketch for matrices?

**Input:** $r$ parties hold vectors $x_1, \ldots, x_r \in \mathbb{R}^n$

each party sends $O(B \log n)$ bits to coordinator

(assume shared randomness)

**Output:** find largest entries in $A = \sum_{i=1}^{r} x_i x_i^T$

more precisely, output approximation $\hat{A}$

$$\left| \hat{A}_{ij} - A_{ij} \right| = \frac{O(1)}{\sqrt{B}} \| A \|_F.$$
CountSketch for matrices?

Input: \( r \) parties hold vectors \( x_1, \ldots, x_r \in \mathbb{R}^n \)

each party sends \( O(B \log n) \) bits to coordinator

(assume shared randomness)

Output: find largest entries in \( A = \sum_{i=1}^{r} x_i x_i^T \)

more precisely, output approximation \( \hat{A} \)

\[
\left| \hat{A}_{ij} - A_{ij} \right| = \frac{O(1)}{\sqrt{B}} \|A\|_F.
\]

\( x_1 \in \mathbb{R}^n \)

\( x_2 \in \mathbb{R}^n \)

\( x_3 \in \mathbb{R}^n \)

\( \vdots \)

\( x_r \in \mathbb{R}^n \)

Coordinator \( \rightarrow \hat{A} \)
CountSketch for matrices?

Input: \( r \) parties hold vectors \( x_1, \ldots, x_r \in \mathbb{R}^n \)

each party sends \( O(B \log n) \) bits to coordinator
(assume shared randomness)

Output: find largest entries in \( A = \sum_{i=1}^{r} x_i x_i^T \)
more precisely, output approximation \( \hat{A} \)

\[
|\hat{A}_{ij} - A_{ij}| = \frac{O(1)}{\sqrt{B}} \|A\|_F.
\]

\( x_1 \in \mathbb{R}^n \)
\( x_2 \in \mathbb{R}^n \)
\( x_3 \in \mathbb{R}^n \)
\( \vdots \)
\( x_r \in \mathbb{R}^n \)

Every party \( i \) sends \( \text{COUNTSKETCH}(x_i x_i^T) \) into \( O(B) \) buckets
(slow)
Define

\[ A = \sum_{i=1}^{r} x_i x_i^T \in \mathbb{R}^{n\times n}. \]
Define
\[ A = \sum_{i=1}^{r} x_i x_i^T \in \mathbb{R}^{n \times n}. \]

Hash function
\[ h: [n] \times [n] \rightarrow [B] \]

and random signs
\[ s: [n] \times [n] \rightarrow \{-1, +1\}. \]
Define

\[ A = \sum_{i=1}^{r} x_i x_i^T \in \mathbb{R}^{n \times n}. \]

Hash function

\[ h : [n] \times [n] \rightarrow [B] \]

and random signs

\[ s : [n] \times [n] \rightarrow \{-1, +1\}. \]

\[ (Sx)_b = \sum_{i, j \in [n] : h(i, j) = b} s(i, j) \cdot x_i x_j. \]
Define
\[ A = \sum_{i=1}^{r} x_i x_i^T \in \mathbb{R}^{n \times n}. \]

Hash function
\[ h : [n] \times [n] \rightarrow [B] \]

and random signs
\[ s : [n] \times [n] \rightarrow \{-1, +1\}. \]

\[ (Sx)_b = \sum_{i,j \in [n]: h(i,j) = b} s(i,j) \cdot x_i x_j. \]

\text{COUNTSKETCH}(x_i x_i^{T}) \text{ takes } n^2 \text{ time to compute...}
Define

\[ A = \sum_{i=1}^{r} x_i x_i^T \in \mathbb{R}^{n \times n}. \]

Hash function

\[ h : [n] \times [n] \rightarrow [B] \]

and random signs

\[ s : [n] \times [n] \rightarrow \{-1, +1\}. \]

\[
(Sx)_b = \sum_{i,j \in [n]: h(i,j) = b} s(i,j) \cdot x_i x_j.
\]

\textsc{CountSketch}(x_i x_i^T)\ takes \ n^2 \ time\ to\ compute...

\textbf{Make hash functions ‘separable’?}
CountSketch for matrices?

Input: \( r \) parties hold vectors \( x_1, \ldots, x_r \in \mathbb{R}^n \)
each party sends \( O(B \log n) \) bits to coordinator
(assume shared randomness)

Output: find largest entries in \( A = \sum_{i=1}^{r} x_i x_i^T \)
more precisely, output approximation \( \hat{A} \)

\[
\left| \hat{A}_{ij} - A_{ij} \right| = O(1) \frac{\|A\|_F}{\sqrt{B}}.
\]

\( x_1 \in \mathbb{R}^n \)
\( x_2 \in \mathbb{R}^n \)
\( x_3 \in \mathbb{R}^n \)
\vdots
\( x_r \in \mathbb{R}^n \)

Coordinator \( \rightarrow \hat{A} \)
CountSketch for matrices?

Input: \( r \) parties hold vectors \( x_1, \ldots, x_r \in \mathbb{R}^n \)
each party sends \( O(B \log n) \) bits to coordinator
(assume shared randomness)

Output: find largest entries in \( A = \sum_{i=1}^{r} x_i x_i^T \)
more precisely, output approximation \( \hat{A} \)

\[
|\hat{A}_{ij} - A_{ij}| = \frac{O(1)}{\sqrt{B}} \|A\|_F.
\]

Every party \( i \) sends \text{SOMESKETCH}(x_i) into \( B \) buckets?
(fast)
Take two independent instances of COUNTSKETCH: hash functions

\[ h_1, h_2 : [n] \rightarrow [B], \]

random signs

\[ s_1, s_2 : [n] \rightarrow \{-1, +1\} \]

Tensor COUNTSKETCH_1 and COUNTSKETCH_2!
Define tensoring of COUNTSKETCH\(_1\) and COUNTSKETCH\(_2\):

\[ h(i, j) = h_1(i) + h_2(j) \pmod{B}. \]
Define tensoring of COUNTSKETCH$_1$ and COUNTSKETCH$_2$:

$$h(i, j) = h_1(i) + h_2(j) \pmod{B}.$$ 

and $s(i, j) = s_1(i) \cdot s_2(j)$. 
Define tensoring of $\text{COUNTSKETCH}_1$ and $\text{COUNTSKETCH}_2$:

$$h(i,j) = h_1(i) + h_2(j) \pmod{B}.$$  

and $s(i,j) = s_1(i) \cdot s_2(j)$.

$$\left( Sx \right)_b = \sum_{i,j \in [n]: h(i,j)=b} s(i,j) \cdot x_i x_j$$  

$$= \sum_{i,j \in [n]: h_1(i)+h_2(j)=b} s_1(i) \cdot s_2(j) \cdot x_i x_j$$
\[(Sx)_b = \sum_{i,j \in [n]: h_1(i) + h_2(j) = b} s_1(i) \cdot s_2(j) \cdot x_i x_j.\]

Can find \(Sx\) from \textit{COUNTSKETCH}_1(x) and \textit{COUNTSKETCH}_2(x) fast! (exercise)
Stronger analysis of **COUNTSKETCH**

The bound

$$\|x - \hat{x}\|_\infty \leq \frac{1}{\sqrt{k}} \|x_{TAIL}\|_2$$

is optimal for sketches with $O(k \log n)$ rows, for worst case $x$. 

---

Minton-Price'14 assumes uniformly random hashing. A very recent improvement:
Stronger analysis of **COUNTSKETCH**

The bound

\[ \| x - \hat{x} \|_\infty \leq \frac{1}{\sqrt{k}} \| x^{TAIL} \|_2 \]

is optimal for sketches with \( O(k \log n) \) rows, for worst case \( x \)

If \( x \) is drawn from a distribution (e.g., power law, Zipfian), one can do better by about \( \log n \) factor: Minton-Price’14
Stronger analysis of COUNTSKETCH

The bound
\[ \| x - \hat{x} \|_\infty \leq \frac{1}{\sqrt{k}} \| x^{TAIL} \|_2 \]

is optimal for sketches with \( O(k \log n) \) rows, for worst case \( x \)

If \( x \) is drawn from a distribution (e.g., power law, Zipfian), one can do better by about \( \log n \) factor: Minton-Price’14

Minton-Price’14 assumes uniformly random hashing. A very recent improvement:

Pseudorandom Hashing for Space-bounded Computation with Applications in Streaming

Praneeth Kacham* Rasmus Pagh† Mikkel Thorup‡ David P. Woodruff§

Abstract

We revisit Nisan’s classical pseudorandom generator (PRG) for space-bounded computation (STOC 1990) and its applications in streaming algorithms. We describe a new generator, HashPRG, that can be thought of as a symmetric version of Nisan’s generator over larger alphabets.
Non-asymptotic measurement complexity?

Good constants are achieved by \( \ell_1 \)-minimization and related (non-sublinear) methods. Get best of both worlds?
Non-asymptotic measurement complexity?

Good constants are achieved by $\ell_1$-minimization and related (non-sublinear) methods. Get best of both worlds?

Ex: in LDPC codes, the Tanner graphs needs to be irregular to achieve capacity – similar effects here?
Non-asymptotic measurement complexity?

Good constants are achieved by \( \ell_1 \)-minimization and related (non-sublinear) methods. Get best of both worlds?

Ex: in LDPC codes, the Tanner graphs needs to be irregular to achieve capacity – similar effects here?

**Information Theoretic Limits of Cardinality Estimation: Fisher Meets Shannon**

Seth Pettie  
pettie@umich.edu  
University of Michigan  
Computer Science and Engineering  
Ann Arbor, MI, USA

Dingyu Wang  
wangdy@umich.edu  
University of Michigan  
Computer Science and Engineering  
Ann Arbor, MI, USA

**ABSTRACT**

Estimating the cardinality (number of distinct elements) of a large multiset is a classic problem in streaming and sketching, dating back to Flajolet and Martin’s classic Probabilistic Counting (PCSA) algorithm from 1983.

In this paper we study the intrinsic tradeoff between the space complexity of the sketch and its estimation error in the RANDOM ORACLE model. We define a new measure of efficiency for cardinality estimators called the Fisher-Shannon (Fish) number \( \mathcal{H}/I \). It captures the tension between the limiting Shannon entropy \( \mathcal{H} \) of the sketch and its normalized Fisher information \( I \), which characterizes the variance of a statistically efficient, asymptotically unbiased estimator.

**KEYWORDS**

cardinality estimation, streaming algorithm

ACM Reference Format:

1 INTRODUCTION

Cardinality Estimation (aka Distinct Elements or \( F_0 \)-estimation) is a fundamental problem in streaming/sketching, with widespread industrial deployments in databases, networking, and sensing.
Non-asymptotic measurement complexity?

Good constants are achieved by $\ell_1$-minimization and related (non-sublinear) methods. Get best of both worlds?

**Ex:** in LDPC codes, the Tanner graphs needs to be irregular to achieve capacity – similar effects here?

---

**Information Theoretic Limits of Cardinality Estimation:**

*Fisher Meets Shannon*

Seth Pettie  
pettie@umich.edu  
University of Michigan  
Computer Science and Engineering  
Ann Arbor, MI, USA

Dingyu Wang  
wangdy@umich.edu  
University of Michigan  
Computer Science and Engineering  
Ann Arbor, MI, USA

**ABSTRACT**

Estimating the cardinality of a multiset is a classic problem, back to Flajolet and Martin.  
In this paper we consider the complexity of the **oracle** model.  
Width estimation for the cardinality estimator captures the tens of sketches and characterizes the unbiased estimate.

**KEYWORDS**

Peeling Close to the Orientability Threshold – Spatial Coupling in Hashing-Based Data Structures

Stefan Walzer*

---

Abstract  

1 Introduction
Non-asymptotic measurement complexity?

Good constants are achieved by $\ell_1$-minimization and related (non-sublinear) methods. Get best of both worlds?

**Ex:** in LDPC codes, the Tanner graphs needs to be irregular to achieve capacity – similar effects here?

**Information Theoretic Limits of Cardinality Estimation: Fisher Meets Shannon**

Seth Pettie  
pettie@umich.edu  
University of Michigan  
Computer Science and Engineering  
Ann Arbor, MI, USA

Dingyu Wang  
wangdy@umich.edu  
University of Michigan  
Computer Science and Engineering  
Ann Arbor, MI, USA

**ABSTRACT**

Estimating the cardinality of a multiset is a classic problem in the theory of data streams. In this paper, we consider the problem in the oracle model. We introduce a new notion of complexity, called the complexity of the oracle model. Our main result is a new algorithm that achieves sublinear complexity in the oracle model. We also provide an upper bound on the complexity of the oracle model and characterize the complexity of the oracle model for different families of functions. Our results give a complete characterization of the complexity of the oracle model for a wide range of functions.

**KEYWORDS**

Peeling Close to the Orientability Threshold – Spatial Coupling in Hashing-Based Data Structures

Stefan Walzer*

**Simple Set Sketching**
**Learning-augmented** sketching: learn the hash function $h$ in \textsc{CountSketch} (and more!) from data

**Adversarially robust** sketching: what if $x$ is chosen by an adversary with (partial) knowledge of the data structure?
Learning-augmented sketching: learn the hash function $h$ in COUNTSKETCH (and more!) from data

Adversarially robust sketching: what if $x$ is chosen by an adversary with (partial) knowledge of the data structure?

---

ON THE ROBUSTNESS OF COUNTSKETCH TO ADAPTIVE INPUTS

---

Edith Cohen* Xin Lyu† Jelani Nelson‡ Tamás Sarlós§ Moshe Shechner¶ Uri Stemmer‖

March 1, 2022

ABSTRACT

COUNTSKetch is a popular dimensionality reduction technique that maps vectors to a lower dimension using randomized linear measurements. The sketch supports recovering $\ell_2$-heavy hitters of a vector (entries with $|v|^2 \geq \frac{1}{k} \|v\|^2$). We study the robustness of the sketch in adaptive settings where input vectors may depend on the output from prior inputs. Adaptive settings arise in processes with feedback or with adversarial attacks. We show that the classic estimator is not robust, and can be attacked with a number of queries of the order of the sketch size. We propose a robust estimator (for a slightly modified sketch) that allows for quadratic number of queries in the sketch size, which is an improvement factor of $\sqrt{k}$ (for $k$ heavy hitters) over prior work.

1 Introduction
Take \textit{(randomized) linear measurements} of the input

\[
S \cdot x = b
\]

sketching matrix

\textit{space} = \text{number of rows}

Distribution of the sketching matrix?
Distribution of the sketching matrix?

\[ +1 \ 0 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ +1 \ 0 \\
0 \ 0 \ 0 \ 0 \ 0 \ +1 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ 0 \\
0 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ +1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ 0 \ 0 \ +1 \ 0 \ 0 \ 0 \ -1 \\
0 \ 0 \ 0 \ 0 \ +1 \ 0 \ 0 \ +1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \]

Bernoulli(±1) or Gaussian linear measurements on random subsets of the universe

The nonzeros are specified by the hash function \( h: [n] \rightarrow [B] \)

Can compute \( Sx \) in time \( \text{nnz}(x) \)!
Distribution of the sketching matrix?

\[
\begin{array}{cccccccccccccc}
+1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & +1 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & +1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

Bernoulli(±1) or Gaussian linear measurements on random subsets of the universe

The nonzeros are specified by the hash function \( h : [n] \rightarrow [B] \)

Can compute \( Sx \) in time \( \text{nnz}(x) \)!
Distribution of the sketching matrix?

\[
\begin{array}{ccccccccccccccc}
+1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & +1 & 0 & 0 & -1 \\
0 & 0 & 0 & +1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

Bernoulli(±1) or Gaussian linear measurements on random subsets of the universe.

The nonzeros are specified by the hash function \( h : [n] \rightarrow [B] \)

Can compute \( Sx \) in time \( \text{nnz}(x)! \)
Distribution of the sketching matrix?

\[
\begin{pmatrix}
+1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & +1 & 0 & 0 & -1 \\
0 & 0 & 0 & +1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

Bernoulli(±1) or Gaussian linear measurements on random subsets of the universe

The nonzeros are specified by the hash function \( h : [n] \rightarrow [B] \)

Can compute \( Sx \) in time \( \text{nnz}(x) \)!
Distribution of the sketching matrix?

\[
\begin{pmatrix}
  +1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
  0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
  0 & -1 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & +1 & 0 & 0 & 0 & -1 & 0 \\
  0 & 0 & 0 & +1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

Bernoulli(±1) or Gaussian linear measurements on random subsets of the universe

The nonzeros are specified by the hash function \( h : [n] \rightarrow [B] \)

Can compute \( Sx \) in time \( \text{nnz}(x) \)!
Distribution of the sketching matrix?

\[
\begin{bmatrix}
+1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & +1 & 0 & 0 & -1 \\
0 & 0 & 0 & +1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Bernoulli(±1) or Gaussian linear measurements on random subsets of the universe

The nonzeros are specified by the hash function \( h : [n] \rightarrow [B] \)

Can compute \( Sx \) in time \( \text{nnz}(x) \)!
Random restrictions (hashing)

What can we learn from $Sx$, where $S$ is just random restrictions?

<table>
<thead>
<tr>
<th>+1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>+1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
</tr>
</tbody>
</table>

Can learn $\|x\|_0$, i.e. number of nonzeros in $x$
Johnson-Lindenstrauss transform

Sketching matrix $S$ = a row of i.i.d. Gaussians of unit variance
Johnson-Lindenstrauss transform

Sketching matrix \( S \) = a row of i.i.d. Gaussians of unit variance

Measures \( \ell_2^2 \) norm of \( x \) in expectation:

\[
\mathbb{E} \left[ \| Sx \|_2^2 \right] = \| x \|_2^2
\]
Johnson-Lindenstrauss transform

Sketching matrix $S = m$ rows of i.i.d. Gaussians of unit variance $\frac{1}{m}$
Johnson-Lindenstrauss transform

Sketching matrix $S = m$ rows of i.i.d. Gaussians of unit variance $1/m$

Measures $\ell_2^2$ norm of $x$ with high probability:

$$
P \left[ \| Sx \|_2^2 \neq \| x \|_2^2 \right] = 1 - \exp(-\Omega(\varepsilon^2 m))$$
Johnson-Lindenstrauss transform

Sketching matrix $S = m$ rows of i.i.d. Gaussians of unit variance $1/m$

Measures $\ell_2^2$ norm of $x$ with high probability:

$$\Pr \left[ \|Sx\|_2^2 \neq \|x\|_2^2 \right] = 1 - \exp(-\Omega(\varepsilon^2 m))$$

Downside: $Sx$ takes $m \cdot n$ time to compute
(Faster) Johnson-Lindenstrauss transform

Subsampled randomized Hadamard transform
Subsampled randomized Hadamard transform

Sketching matrix $S = P \cdot H \cdot D$
(Faster) Johnson-Lindenstrauss transform

Subsampled randomized Hadamard transform

Sketching matrix  \( S = P \cdot H \cdot D \)

\( D = \) diagonal random sign matrix,  \( H = \) Hadamard transform,  \( P = \) random sampling matrix
(Faster) Johnson-Lindenstrauss transform

Subsampled randomized Hadamard transform

Sketching matrix $S = P \cdot H \cdot D$

$D$=diagonal random sign matrix, $H$=Hadamard transform, $P$=random sampling matrix

$Sx$ can be computed in $O(m + n\log n)$ time
Frequency moments

The $p$-th frequency moment

$$F_p = \sum_{i \in [n]} f_i^p$$
Frequency moments

The $p$-th frequency moment

$$F_p = \sum_{i \in [n]} f_i^p$$

Theorem

*Can approximate $F_p$ for all $p \in [0,2]$ in polylogarithmic space, but need $\Omega(n^{1-2/p})$ space for all $p > 2$*
Frequency moments

The $p$-th frequency moment

$$F_p = \sum_{i \in [n]} f_i^p$$

**Theorem**

*Can approximate $F_p$ for all $p \in [0,2]$ in polylogarithmic space, but need $\Omega(n^{1-2/p})$ space for all $p > 2$*

---

Bar-Yossef et al. Information complexity approach to data stream lower bounds