Making Data Sketches Accurate and Fast by Filtering the Cold and Aggregating Items

Yang Zhou¹,², Tong Yang², Jie Jiang², Bin Cui², Omid Alipoufard³, Minlan Yu¹, Xiaoming Li², Steve Uhlig⁴

Harvard University¹, Peking University², Yale University³, Queen Mary University of London⁴
Data Streams are Pervasive

Network traffic  Video streaming  Sensor data  Web click data (etc.)

In many applications, some statistical information is needed!

Applications: Network measurement, DBMS optimization, Search engine design, Security, etc.

Information required: flow size, heavy hitters, heavy changes, quantiles, etc.
Accurate and Fast Data Stream Analysis is Challenging

Challenges:
1. Memory constraint
   - Fit into cache to boost speed
   - Hardware on-chip memory limited
2. Single-pass requirement
   - Data is of huge volume and fast speed: Dumping into disk is hard
   - Some applications need online analysis

Exact statistics (e.g., by using hash tables) are difficult to obtain (and often unnecessary)!
## Data Sketches can Help

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Data Sketch Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency estimation</td>
<td>Count-Min, CM-CU, Count, ASketch</td>
</tr>
<tr>
<td>Top-k Hot items</td>
<td>Count-Min, CM-CU, Space-Saving ASketch, FlowRadar, UnivMon</td>
</tr>
<tr>
<td>Heavy changes</td>
<td>RevSketch, FlowRadar, UnivMon, Space-Saving</td>
</tr>
<tr>
<td>Superspreader /DDoS detection</td>
<td>TwoLevel</td>
</tr>
<tr>
<td>Frequency distribution</td>
<td>MRAC, FlowRadar</td>
</tr>
<tr>
<td>Cardinality</td>
<td>FM, LC, UnivMon</td>
</tr>
<tr>
<td>Entropy</td>
<td>FlowRadar, UnivMon</td>
</tr>
</tbody>
</table>
## Data Sketches can Help

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Data Sketch Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency estimation</td>
<td>Count-Min, CM-CU, Count, ASketch</td>
</tr>
<tr>
<td>Top-k Hot items</td>
<td>Count-Min, CM-CU, <strong>Space-Saving</strong></td>
</tr>
<tr>
<td></td>
<td>ASketch, FlowRadar, UnivMon</td>
</tr>
<tr>
<td>Heavy changes</td>
<td>RevSketch, FlowRadar, UnivMon, Space-Saving</td>
</tr>
<tr>
<td>Superspreader /DDoS detection</td>
<td>TwoLevel</td>
</tr>
<tr>
<td>Frequency distribution</td>
<td>MRAC, FlowRadar</td>
</tr>
<tr>
<td>Cardinality</td>
<td>FM, LC, UnivMon</td>
</tr>
<tr>
<td>Entropy</td>
<td>FlowRadar, UnivMon</td>
</tr>
</tbody>
</table>
Count-Min Sketch — Estimating Frequencies

Insertion

Query

frequency: \textbf{18} = \text{Min}\{19, 24, 26, 18\}
Space-Saving — Finding Top-k Hot Items

- Maintaining a heap-like data structure.
- If Space-Saving is full, the smallest item will be replaced by the new item, whose frequency is initialized to be $f_{\text{min}} + 1$.
Limitations of Conventional Data Sketches

Cold & hot items

Sketch

Real Data Streams:
Highly skewed
-> Majority: Cold items
-> Minority: Hot items

Count-Min:
All items use large counters
-> A waste of memory

Space-Saving:
A great many of replacements caused by cold items are unnecessary
-> poor accuracy
Methodology of Cold Filter*

**Count-Min:**
- Use small counters in CF
  - record cold items
- Use large counters in sketch
  - record hot items

**Space-Saving:**
- CF filters many cold items
  - reduce # unnecessary replacements

Agg-Evict: Optimizing Speed

Ideally, $\frac{8}{3}=2.67$ speed-up

$\rightarrow$ How to design an efficient Aggregator?
Design of Agg-Evict

1. Using SIMD to query continuous cells in a K-V pair array
2. Using Random Eviction for simplicity and speed

Accuracy Improvement

Frequency estimation: Varying the CF size

All algorithms use the same memory size

Finding Top-k hot items: Varying k
Speed Improvement

![Bar chart showing throughput (Mpps) for various systems with and without Agg-Evict.](image)
Conclusion

- **Cold Filter**: Improving accuracy by filtering the cold
- **Agg-Evict**: Improving speed by aggregating items
- **Generic**: Applicable to many different data sketches
Thanks!

Source Code:  https://github.com/zhouyangpkuer/ColdFilter,