Differentially Private Hierarchical Heavy Hitters

Ari Biswas [1] Graham Cormode [2] Yaron Kanza [3] Divesh Srivastava [3] Zhengyi Zhou [3]

[1]University Of Warwick [2]Meta AI [3]AT&T Research

Hierarchical Heavy Hitters

The nodes of the tree describe the elements of the hierarchy $\mathcal{H}$. Use notation $e \succeq p$ to describe that p is a parent of node e.

- Heavy Hitters (HH) tells us if an element is heavy, but Hierarchical Heavy Hitters (HHH) tells us how that element is heavy.
- HHH allows us to distinguish between an element that is heavy because it has a heavy child (or a few heavy children) and an element that is heavy because it has many light children that are cumulatively heavy.
- HHH generalises HH. Given HHH, we can compute HH, but not the other way round.

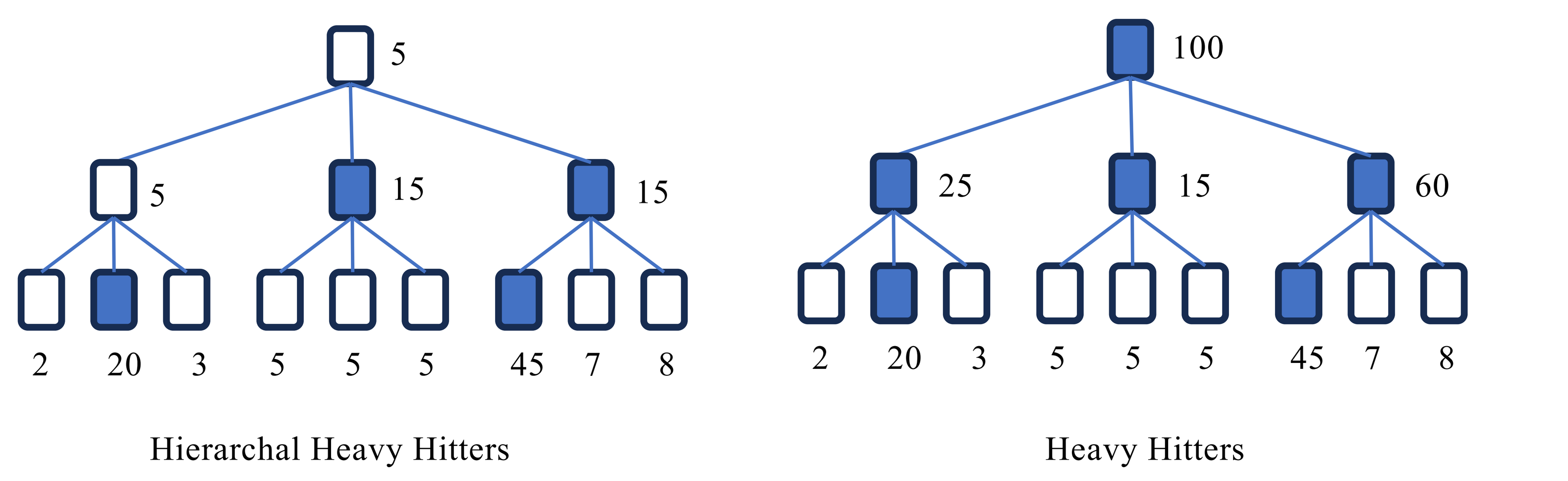

Figure 1. A dataset of 100 elements over a hierarchy with residual counts (left) and unconditional counts (right).

Unconditional Frequency

The unconditional frequency of any element $p \in \mathcal{H}$, denoted by $f_X(p)$, is the number of elements in X that generalise to p.

$$f_X(p) = \sum_{e \in X} \mathbb{1}[e \succeq p]$$

Conditional/Residual Frequency

Given a dataset X, and a set $\mathcal{S} \subseteq \mathcal{H}$, we say $x \not\succ \mathcal{S}$ if $\nexists q \in \mathcal{S}$ such that $x \succeq q$. We define the conditional or residual count $F_{\mathcal{S}}(p)$ of a prefix p with respect to $\mathcal{S}$ as the sum of all fully specified elements who do not have a parent already in $\mathcal{S}$.

$$F_{\mathcal{S}}(p) = \sum_{e \in X \wedge e \succeq p \wedge e \not\succ \mathcal{S}} f_X(e)$$

Relative Error Vs Absolute Error

With probability $1 - \eta$, we want

- Simultaneous Absolute Error

$$\max_{p \in \mathcal{H}} \left| f_X(p) - \widetilde{f}_X(p) \right| \leq \Delta'$$

- Simultaneous Relative Error[a]

$$\max_{p \in \mathcal{H}} \left| \frac{f_X(p) - \widetilde{f}_X(p)}{f_X(p)} \right| \leq \Delta''$$

where $\Delta' \in \mathbb{R}$ and $\Delta'' \in [0, 1]$

[a] Some works, such as [2] use an additive version for relative error.

Problem Statement + Results

Input to Algorithm

- Database X of size n fully specified (leaves) elements from a hierarchy $\mathcal{H}$ with height h.
- Privacy parameter $\varepsilon \in (0, \log n)$, $\delta = o\left(1/n^2\right)$
- Threshold $\tau > \frac{8}{\varepsilon}\log(2h/\delta) + 1$.
- Confidence $\eta \in (0, 1/2)$

Algorithm 1 DP-HHH Detection With No Memory Constraint

$\gamma \xleftarrow{\$}$ Laplace$(\frac{2}{\varepsilon})$
$\mathcal{S} = \{\}$
for $i = h, \ldots, 1$ do
  $\mathcal{A}_i = \{p \in \mathcal{H} | \text{Level}(p) = i\}$
  for $p \in \mathcal{A}_i$ do
    if $F_{\mathcal{S}}(p) = 0$ then
      continue to next iteration
    end if
    $w_p \xleftarrow{\$}$ Laplace$(\frac{4}{\varepsilon})$
    if $F_{\mathcal{S}}(p) + w_p + \gamma \geq \tau$ then
      $\mathcal{S} = \mathcal{S} \cup \{p\}$
      $\widetilde{F}_{\mathcal{S}}(p) = F_{\mathcal{S}}(p) +$ Laplace$(\frac{4}{\varepsilon})$
    end if
  end for
end for
$\widetilde{f}_X(p) = \sum_{q \in \mathcal{S} \wedge q \succeq p} \widetilde{F}_{\mathcal{S}}(q)$
Output $\mathcal{S}$ and $\{\widetilde{f}_X(p)\}_{p \in \mathcal{S}}$.

Output Of Algorithm

Hierarchical Heavy Hitters $\mathcal{S} \subseteq \mathcal{H}$, and, their approximate unconditional frequencies $\{\widetilde{f}_X(p)\}_{p \in \mathcal{S}}$ such that for some error parameter $\Delta \in \mathbb{R}^+$

- **Privacy**: The Algorithm is (ε, δ)-DP.
- **Coverage**: With probability $1 - \eta$, for *any* element $p \notin \mathcal{S}$, $F_{\mathcal{S}}(p) \leq \tau - \Delta$.
- **Simultaneous Relative Error**: With probability $1 - \eta$,

$$\max_{p \in \mathcal{H}} \left| \frac{f_X(p) - \widetilde{f}_X(p)}{f_X(p)} \right| \leq \frac{\Delta}{\tau}$$

Coverage And Error Guarantee

Algorithm 1 is (ε, δ)-DP and satisfies simultaneous relative error and coverage guarantees for any

$$\Delta \geq \frac{8}{\varepsilon}\left(\log\frac{1}{\delta} + \log\frac{2h}{\eta}\right)$$

Privacy Proof Sketch

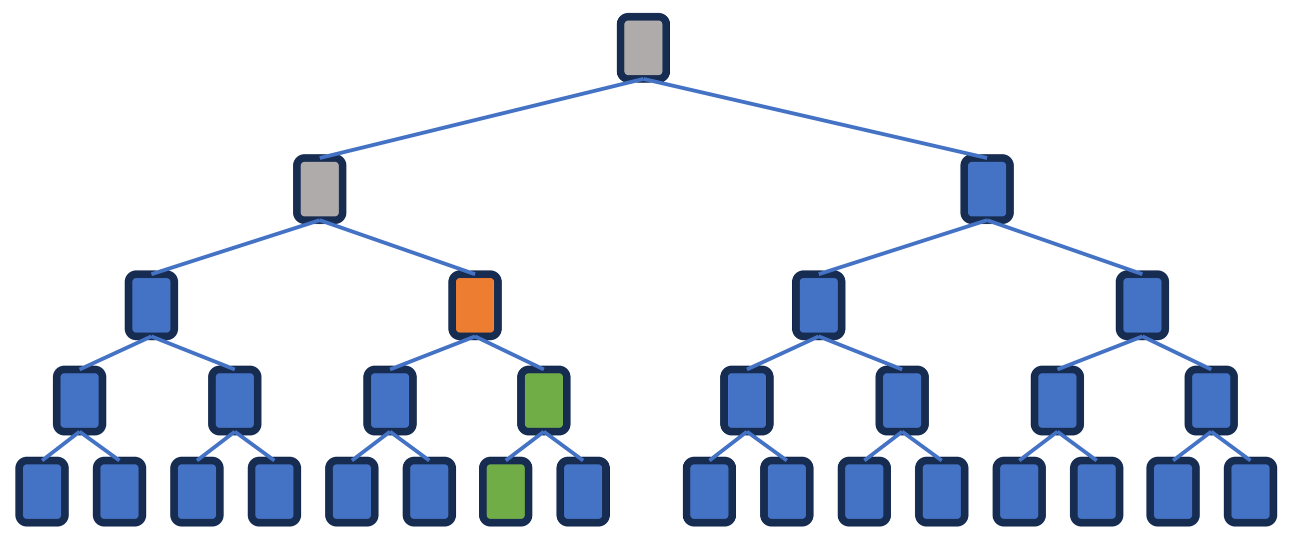

Figure 2. We can partition $\mathcal{H}$ into 4 disjoint sets. Despite there being an exponentially many nodes we only pay for 2

Given $u^\star$ (orange node), observe that we can partition $\mathcal{H}$ into 3 sets.

1. $\mathcal{I}_{\text{Unrelated}} = \{v \in \mathcal{H} | x' \not\succeq v\}$ (shown in blue in Figure 2)
2. $\mathcal{I}_{\text{After}} = \{v \in \mathcal{H} | u^\star \succ v\}$ (shown in grey in Figure 2).
3. $\mathcal{I}_{\text{Active}} = \mathcal{H} \setminus (\mathcal{I}_{\text{Unrelated}} \cup \mathcal{I}_{\text{After}})$ (denoted by all nodes that are green and orange in Figure 2).

Edge case handled by Stability Histogram.

Related Work + Other Results

- Non-private HHH problem first introduced by Cormode *et al.*[1].
- Any (ε, δ)-DP algorithm must incur $\widetilde{\Omega}\left(\frac{h}{\varepsilon}\right)$ *absolute* error to estimate the count for *any* element in a hierarchy of height h [2].
- Also show a privatised version of the Misra Gries (MG) sketch by Mitzenmacher *et al.*[4] where the *absolute* error is independent of the size of the sketch, despite the sensitivity of the MG sketch being linear in the size of the sketch.
- Proof builds on the private MG sketch by Lebeda and Tatek [3].
- There is a gap between the memory constrained problem, and the unlimited memory solution - the dependence on h cannot be removed.

References

[1] Graham Cormode, Flip Korn, Shanmugavelayutham Muthukrishnan, and Divesh Srivastava. Finding hierarchical heavy hitters in data streams. In *Proceedings 2003 VLDB Conference*, pages 464–475. Elsevier, 2003.

[2] Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, and Kewen Wu. On differentially private counting on trees. *arXiv preprint arXiv:2212.11967*, 2022.

[3] Christian Janos Lebeda and Jakub Tetek. Better differentially private approximate histograms and heavy hitters using the misra-gries sketch. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 79–88, 2023.

[4] Michael Mitzenmacher, Thomas Steinke, and Justin Thaler. Hierarchical heavy hitters with the space saving algorithm. In *2012 Proceedings of the Fourteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 160–174. SIAM, 2012.