Federated Heavy Hitters revisited

Graham Cormode

July 14, 2021

1 Introduction

The problem of finding the most frequent items from a collection is a core analytics task that supports a range of objectives, from simple popularity charts, to instantiating complex language models. Due to the sensitivity of data used within these applications, it is necessary to apply strong privacy protections to the data.

Recent wok on Federated Heavy Hitters discovery [1] describes an algorithm to collect information from a set of distributed clients, who each hold a (private) item. We can treat these items as strings of characters over a fixed alphabet. The algorithm proceeds in a series of L rounds to build up a trie describing the frequent items among the client population. In each round, the server contacts a random sample of m clients, and shares the current trie with them. Each client replies if its item extends the trie, and if so the client "votes" for the prefix that its item extends, along with the next character.

The server receives these votes, and tallies them. If there are more than a threshold θ of votes for a particular extension of the trie, then this extension is added to the trie as part of the next level. We assume that the procedure stops after the trie has been built out to L levels, or if the trie cannot be extended beyond a certain level.

In the original paper, the protocol is analyzed, and it is shown that the trie of prefixes built in this fashion meets the requirements of (ε, δ)-Differential privacy. In this note, we extend this approach to heavy hitters as follows:

- We show that we can augment the trie that is output with counts of the items received from the clients and still meet (ε, δ)-DP.
- We show that these counts provide accurate frequency estimates for items and prefixes from the input
- We show that the resulting data structure can also answer quantile and range queries
- Our proofs are compact and self-contained.

2 Privacy Analysis including counts

We adopt the algorithm and notation from the TrieHH paper [1], with some modifications.

The algorithm proceeds over L levels. At each level, we sample clients to report on their items that extend the current trie. In our version, each client out of n is sampled with probability $p_s = m/n$, so the expected size of the sample is m. (We later discuss different ways to implement this sampling). We add items to the current trie based on an (absolute) threshold θ. We augment the output of TrieHH by including the observed count of prefix for each node in the trie, provided it is more than a fixed threshold θ.

We can view the action at each level as materializing a histogram of the sampled clients. The core of the argument is that the histogram (after rounding counts less than θ to zero) is (ε, δ)-differentially private.

Lemma 1. *The probability that the number of samples of a prefix in a round is more than θ times its expectation is at most δ, for $\theta = 3 + \ln 1/\delta$.*

Proof. Given a prefix that occurs k times in the input, each occurrence has probability $p_s = m/n$ of being picked. The expected number of sampled occurrences is then $kp_s = km/n$.

Let X denote the random variable that counts the number of successes (times the prefix is picked) out of the k trials, so $\mathsf{E}[X] = km/n$. Then, X is a sum of k Bernoulli random variables with parameter p_s. We do a case split on p_s:

Case: $p_s \leq 1/k$**.** If $p_s \leq 1/k$, we apply an (additive) Chernoff-Hoeffding bound to the mean of the k trials:

$$\Pr[X \geq \theta] = \Pr\left[\frac{1}{k}X - \frac{1}{k}\mathsf{E}[X] \geq (\theta p_s - p_s)\right] \leq \exp\left(-D\left(\frac{\theta}{k}\middle\|\frac{1}{k}\right)k\right).$$

Here, $D(p\|q)$ denotes the K-L divergence (relative entropy) between the (Bernoulli) distributions with parameters p and q. We have

$$\begin{aligned}
-D(p\|q)k &= -\theta \ln\left(\frac{\theta}{k}\cdot\frac{k}{1}\right) - (k-\theta)\ln\left(\frac{k-\theta}{k}\cdot\frac{k}{k-1}\right)\\
&= -\theta\ln\theta - (k-\theta)\ln\left(1 - \frac{\theta-1}{k-1}\right)\\
&= -\theta\ln\theta + (k-\theta)\ln\left(\frac{k-1}{k-\theta}\right)\\
&= -\theta\ln\theta + (k-\theta)\ln\left(1+\frac{\theta-1}{k-\theta}\right)\\
&\leq -\theta\ln\theta + \theta - 1
\end{aligned}$$

Hence,

$$\Pr[X \geq \theta] \leq \exp(-\theta\ln\theta + \theta - 1) \tag{1}$$

For this case, to achieve a target error bound δ, we rearrange to obtain $\frac{\theta}{e}\ln\frac{\theta}{e} = \frac{1}{e}\ln(1/e\delta)$, and apply Lambert's W function. This gives $\frac{\theta}{e} = W(\frac{1}{e}\ln(1/e\delta)$, i.e., $\theta = eW(\frac{1}{e}\ln\frac{1}{e\delta})$. Note that this case corresponds to the scenario where we do not publish the counts, but only indicate which items occurred more than θ times in the sample.

Case: $p_s > 1/k$**.** If $p_s > 1/k$, we apply a (multiplicative) Chernoff bound:

$$\Pr[X \geq \theta\mathsf{E}[X]] \leq \exp(-(\theta-1)^2\mathsf{E}[X]/(1+\theta)) = \exp(-(\theta-1)^2 kp_s/(1+\theta)) \leq \exp(-(\theta-1)^2/(1+\theta))$$

In this case, to achieve a target error bound δ, we can pick $\theta = 3+\ln(1/\delta)$, and obtain $\exp(-(2+\ln 1/\delta)^2/(4+\ln 1/\delta)) < \exp(-\ln(1/\delta)) = \delta$.

The second case is stricter for all $\theta > 1$, so we will use this setting of θ in what follows. □

We next give a bound on the ratio of probabilities of seeing the same output on neighboring inputs. We can view the TrieHH protocol as materializing a histogram at each level, with progressively finer cells. In the protocol as originally described, cells whose ancestor in a previous level did not exceed the θ threshold are not eligible for consideration. However, the proof still applies if we do not enforce such restrictions.

Lemma 2. *Given two neighboring inputs D, D', such that D differs in one item from D', the ratio of probabilities of seeing a cell with a given value τ is bounded by $\frac{k+1}{k+1-\tau}$, where $k+1$ is the number of copies of the given item in input D and $k > \tau$.*

Proof. The case to focus on is when input D has one extra copy of a particular item compared to D', at some intermediate stage of the algorithm. For notation, we will write $S_k(n,s,\tau)$ to denote the number of ways to succeed in collecting exactly τ instances of the target item while picking s items out of n, when there are k total instances of the item. We can observe that there is a simple combinatorial expression for this quantity: we count the number of combinations where we pick a particular subset of size τ from the k instances, and a particular subset of size $s-\tau$ from the remaining $n-k$ examples.

$$S_k(n,s,\tau) = \binom{k}{\tau}\binom{n-k}{s-\tau} \tag{2}$$

Our goal is to bound the ratio of probabilities of seeing a count of τ copies of the item in the output of D, who has $k+1$ copies of the item, and of D' who holds k copies. The probability that the sample size is exactly s is given by $P_s = p_s^s(1-p_s)^{n-s}$. For a given sample size s, the probability for D is $S_{k+1}(n,s,\tau)P_s$, and for D' it is $S_k(n,s,\tau)P_s$. Then this ratio of probabilities is given by

$$\frac{S_{k+1}(n,m,\tau)}{S_k(n,m,\tau)} = \frac{\binom{k+1}{\tau}\binom{n-k-1}{m-\tau}}{\binom{k}{\tau}\binom{n-k}{m-\tau}} = \frac{(k+1)(n-k-m-\tau)}{(n-k)(k+1-\tau)} = \left(1-\frac{m+\tau}{n-k}\right)\left(\frac{k+1}{k+1-\tau}\right) \leq \frac{k+1}{k+1-\tau}$$

Then we can bound this ratio across all sample sizes as simply $\sum_{s=0}^{n} \frac{k+1}{k+1-\tau} P_s = \frac{k+1}{k+1-\tau}$ □

Lemma 3. *The Trie-HH protocol satisfies (ε,δ)-DP for $\delta = L\exp(-(\theta-1)^2/(1+\theta))$ and $\varepsilon = O(Lm\ln(1/\delta)/n)$.*

Proof. The protocol proceeds in rounds. We can view the protocol as publishing a histogram at each level, where the granularity of the cells is refined in each round. The protocol enforces that if a prefix is not included at a particular level, then none of its extensions are published in any subsequent level. However, we can view this as "post-processing", and analyze the simpler algorithm that does not enforce this constraint.

We can consider two inputs D and D', such that D has one more copy of a particular string than D', and argue that the DP guarantee holds on the output of the protocol. Denote this string as t. We aim to show that the probability that the mechanism, M, applied to input D, behaves similarly to the same mechanism applied to D'. Specifically, we want to show that

$$\Pr[M(D) \in R] \leq \exp(\varepsilon)\Pr[M(D') \in R] + \delta$$

for any set of feasible outputs R.

First, we consider a single round i, and let t_i denote the string t truncated to level i. Observe that for any string $t' \neq t$, if $t'_i \neq t_i$, then at level i, t'_i has the same number of copies in D and D', and so is treated identically in both cases. Hence, we can focus only on the treatment of t_i in the two cases. We condition on the event that the number of samples of the prefix t_i is not more than θ times its expectation. Call this event E_i. By Lemma 1, event E_i holds except with probability $p_\theta = \exp(-(\theta-1)^2/(\theta+1))$. We condition on E_i holding, and just account for this probability in our final reckoning.

Suppose at round i, the count of t_i for D is less than n/m. Then, by our assumption on θ, D will not sample θ copies of t at this level, and so both D and D' would output the same histogram. Hence, the probability of all outputs are equal on D and on D'.

Otherwise, the count of t_i ($k+1$ for D) is at least n/m, and by our assumption D samples at most $\tau \leq \theta m(k+1)/n$ copies of t_i. Then, by Lemma 2, we can state that for the mechanism at level i, M_i, the probability of seeing a given output histogram T_i satisfies:

$$\frac{\Pr[M_i(D) = T_i|E_i]}{\Pr[M_i(D') = T_i|E_i]} \leq \frac{k+1}{k+1-\tau} \leq \frac{k+1}{k+1-\theta(k+1)m/n} = \frac{n}{n-\theta m} \tag{3}$$

We will assume that $m \leq \frac{n}{10\theta}$. The effect is to ensure that the sample size m is a small fraction of n. Substituting this assumption in (3), we conclude

$$\frac{\Pr[M_i(D) = T_i|E_i]}{\Pr[M_i(D') = T_i|E_i]} \leq 1 + \frac{m\theta}{n-n/10} = 1 + \frac{10m\theta}{9n} := \exp(\varepsilon_i) \tag{4}$$

Then we can write

$$\begin{aligned}
\Pr[M_i(D) \in R] &\leq \Pr[E_i]\Pr[M(D) \in R|E_i] + (1-\Pr[E_i]) \\
&\leq \Pr[E_i]\sum_{T \in R}\Pr[M_i(D) = T|E_i] + \Pr[\sim E_i] \\
&\leq \Pr[E_i]\exp(\varepsilon_i)\sum_{T \in R]}\Pr[M_i(D') = T|E_i] + \Pr[\sim E_i] \\
&= \exp(\varepsilon_i)\Pr[E_i]\Pr[M(D') \in R|E_i] + \Pr[\sim E_i] \\
&\leq \exp(\varepsilon_i)\Pr[M_i(D') \in R] + p_\theta
\end{aligned}$$

Consequently, we can claim that the mechanism M_i satisfies $(\varepsilon_i, p_\theta)$ differential privacy.

Finally, we argue that the output of the full protocol is the L-fold composition of the mechanisms M_i. Let $\varepsilon = \sum_{i=1}^{L}\varepsilon_i$. Using basic composition, we obtain a bound of $(L\varepsilon', Lp_\theta)$-differential privacy, leading to the result stated in the theorem claim. For $\varepsilon_i = \varepsilon' < 1$, we can obtain a tighter bound, of $(L\varepsilon'^2 + \varepsilon'\sqrt{L\log 1/(p_\theta L)}, 2Lp_\theta)$. □

Remark. We remark that if the objective is only to find the heavy hitters, then the factor of L can be dropped from these bounds. That is, instead of proceeding in rounds, we simply apply the basic histogram protocol to the full inputs, and report the items which survive the thresholding process (along with their associated counts if desired). Following the above analysis, the resulting output is (ε,δ)-differentially private, when setting $\theta = 3+\ln 1/\delta$ and $m \leq \frac{9n}{10\theta}$ to get $\varepsilon = 10\theta m/n$.

2.1 Fixed sized sampling

In the original TrieHH paper, the analysis proceeds for a fixed size sample (the sample size is exactly m). In our setting, we perform Poisson sampling instead of taking a fixed sample. The reason is that if the size of the sample is known, then we are effectively also releasing the number of samples that were suppressed by the θ threshold (by adding up the released counts, and subtracting from m). This potentially leaks information. Consider the case where D' contains n copies of the same item, while D contains $n-1$ copies of the same item, and one unique item. With probability m/n, the mechanism on input D samples the unique item along with $m-1$ other items, and so produces a sample of size $m-1$. But on input D', there is zero probability of producing a sample smaller than m. This forces $\delta \geq m/n$, which is typically too large for (ε,δ)-DP (we usually seek $\delta \ll 1/n$).

Performing Poisson sampling with p_s addresses this problem: the expected sample size is the same, but we no longer leak the true size of the sample before censoring. Indeed, we can see that the (observable) size of the sample is differentially private: given two inputs D and D' such that D has one additional unique item, the distribution of sample sizes are close, up to a factor of $1+p_s/(1-p_s) = 1+m/(n-m)$, which is at below $\exp(\varepsilon_i)$ by (4).

Implementing Poisson sampling may appear costly: the server needs to contact n clients instead of m, where we expect $n \gg m$. However, we can perform this more efficiently. Observe that, with high probability, the size of the sample will be close to expected value of m. In particular, by a Chernoff bound, the probability that the sample size is more than $c\sqrt{m}$ larger than m is

$$\Pr[s > (1+c/\sqrt{m})m] \leq \exp(-c^2/3).$$

Hence, for c a suitable constant (say, 10), this probability is negligibly small. To realize this sampling, we contact a fixed size number of clients $s = m + c\sqrt{m}$, and then have each client perform a Bernoulli test on whether to participate: with probability m/s, it participates, otherwise it abstains. An abstaining client can, for example, vote for a unique element (e.g., an item based on a hash of its identifier), and so be automatically discounted from the protocol, without revealing this information to an aggregator.

3 Accuracy Bounds

The trieHH algorithm is ultimately based on sampling and pruning, so for any prefix whose frequency is sufficiently above the pruning threshold, then its frequency within the trie is an (almost) unbiased estimate of its true frequency. There is a small gap, since even for a prefix with high frequency, there is a small chance that it is not sampled often enough, and so its estimate will fall below the threshold θ (otherwise, we do not materialize the node).

We first consider the probability that a frequent item is not reported by the algorithm. For a prefix with (absolute) frequency W out of the n input items, it is reported at level i (conditioned on its length $i-1$ prefix being reported at level $i-1$) if the number of sampled occurrences exceeds θ. Similar to the analysis above, we can apply a Chernoff-Hoeffding bound to the random variable X that counts the number of occurrences of the prefix. Now, the probability of each sample picking the prefix is W/n, and the expected number in the sample is $Wm/n > \theta$. For convenience, we will write $w = Wm/n$ for this expectation. We have that[1]

$$\Pr[X \leq \theta] = \Pr\left[X \leq \frac{\theta}{w} w\right] = \Pr\left[X \leq \left(1 - \frac{w-\theta}{w}\right)\mathsf{E}[X]\right] = \exp\left(-\frac{(w-\theta)^2}{2w}\right) \tag{5}$$

When w is sufficiently bigger than θ, this gives a very strong probability. For example, consider the case $n = 10^6$, and we set $\theta = 15$ to obtain a modest δ of 10^{-6}. The sample size $m = 12{,}000$, and for an item that occurs 1% of the time in the input, we expect to sample it $w = 120$ times. This gives a bound of $\exp(-45) = 10^{-20}$ that such an item is not detected at any level. Over $L = 10$ levels, a union bound ensures that the chance of missing it remains at most 10^{-19}.

[1] Here we are sampling without replacement. However, using bounds for sampling with replacement are still valid here.

More generally, we can use the frequency of any prefix in the trie as an estimate for its true occurrence rate. Applying the same Chernoff-Hoeffding bound as above, we have for $\gamma < 1$,

$$\Pr[|X-\mu| > \gamma\mu] = 2\exp(-\gamma^2\mu/3) = \beta$$

Rearranging, we obtain $\mu = \frac{3}{\gamma^2}\ln 1/2\beta$. Suppose we aim to find all items whose frequency is at least ϕ, and estimate their frequency with relative error at most γ. Then we have $\mu = \phi m \geq \phi\frac{\varepsilon n}{L\theta} = \frac{3}{\gamma^2}\ln(1/2\beta)$.

We can substitute values into this expression to explore the space. For example, if we set $\varepsilon = 3$, $L = 10$, $\ln 1/2\beta = 10$, $\theta = 10$ and $\gamma = 1/\sqrt{10}$, then we obtain $\phi = 10^4/n$ — in other words, provided $n > 10^6$, we can accurately find estimates of frequencies that occur 1% of the time (except with vanishingly small probability).

4 Quantiles via TrieHH

We can observe that the Trie built during the TrieHH protocol can also be used to answer quantile queries, with the same privacy (and similar accuracy) guarantees.

Now each client has an input value in the range $[0,1]$ (say), and we can interpret these as prefixes, corresponding to subranges. For example, if we set the “alphabet size”, α to 4, then the input value $\frac{1}{3}$ falls in the range $[0.25, 0.5]$ for a prefix of length 1; and in the range $[\frac{5}{16}, \frac{6}{16}]$ for a prefix of length 2. With this mapping of values to prefixes, the algorithm proceeds as before, and outputs the (DP) trie with weights on nodes.

To answer a range query $[0,r]$, we decompose the range greedily into chunks that can be answered by the trie. For example, if $\alpha = 4$, and we want the range $[0, 0.7]$, we find the chunks $[0, \frac{1}{4}], [\frac{1}{4}, \frac{2}{4}]$ at level 1; $[\frac{8}{16}, \frac{9}{16}], [\frac{9}{16}, \frac{10}{16}], [\frac{10}{16}, \frac{11}{16}]$ at level 2; and so on.

Due to the pruning, we will not have information on any ranges whose sampled weight is less than θ, corresponding to a θ/m fraction of mass. This will give an error bound of $(\alpha-1)\theta/m$ per level, and so $L(\alpha-1)\theta/m$ over all levels. Based on our setting of m proportional to $\varepsilon n/(L\theta)$, we obtain a total error of $(\alpha-1)(L\theta)^2/\varepsilon n$.

Picking similar test values as above shows that this can give reasonable accuracy for n large enough. For $\theta = 10$, $L = 10$, $\alpha = 4$, $\varepsilon = 2$, the error bound yields $\frac{3}{2}10^4/n$. So for $n > 10^6$, we obtain rank queries (and quantiles) in this space with error around 0.015.

References

[1] Wennan Zhu, Peter Kairouz, Haicheng Sun, Brendan McMahan, and Wei Li. Federated heavy hitters discovery with differential privacy. *CoRR*, abs/1902.08534, 2019.