Sample-and-threshold differential privacy: Histograms and applications

Akash Bharadwaj
Facebook
akashb@fb.com

Graham Cormode
Facebook
gcormode@fb.com

Abstract

Federated analytics aims to compute accurate statistics from distributed datasets. A "Differential Privacy" (DP) guarantee is usually desired by the users of the devices storing the data. In this work, we prove a strong (ε, δ)-DP guarantee for a highly practical sampling-based procedure to derive histograms. We also provide accuracy guarantees and show how to apply the procedure to estimate quantiles and modes.

1 Introduction

Building private histograms is a task that underpins a variety of machine learning and data analytics tasks. Histograms enable building usable discrete representations, distributions and marginals. Materializing histograms is thus a core subroutine in instantiating graphical models for synthetic data generation (McKenna et al., 2021), and hence they support numerous statistical analyses and inference tasks. The problem has been heavily studied in the setting of differential privacy, with a number of results shown under various models, such as the central model (Dwork, 2006; Xu et al., 2012; Dwork and Roth, 2014), local model (Bassily and Smith, 2015; Wang et al., 2017; Acharya et al., 2019) and shuffle model (Erlingsson et al., 2020; Balcer and Cheu, 2020; Li et al., 2020). All prior work has fundamentally relied on adding noise to data as a means to ensure privacy.

In this paper, we revisit this foundational question, and show how differential privacy can be obtained via a simple sample-and-threshold mechanism, which can be readily implemented in a distributed setting. Importantly, privacy is derived from enforcing a minimum threshold on the exact counts evaluated on sampled data—there is no additional explicit addition of noise. This has the convenient characteristic of avoiding false positives in the derived histogram, i.e. no spurious counts are added to it. Moreover, it is intuitively appealing to know that data will not be released unless it occurs with some minimum frequency—similar to the motivations of k-anonymity. We also note that while building production systems for federated analytics (Bonawitz et al., 2019), sampling participating devices is often unavoidable because of underlying periodicity in their availability, throughput limitations of production systems, or the need for timely and inexpensive results. Thus, being able to extract a DP guarantee from the sampling operator is opportune. Equipped with an private and efficient mechanism for histogram computation, we can apply it to various core analytics tasks.

Our contributions. We introduce a histogram mechanism that extends prior work by showing that:

1. The sample-and-threshold approach gives an (ε, δ)-DP guarantee
2. The counts provide accurate frequency estimates for items in the federated data set
3. The resulting mechanism can also answer heavy hitter, quantile and range queries

The key is to choose a sampling rate that is not too large compared to population size, and to prune items with low frequency in the sample, so that the presence of an item in the pruned sample does not indicate exactly how many instances were in the original population. While prior work has considered

Preprint. Under review.

the ability of sampling to amplify DP privacy bounds from noise addition, we observe that sampling (when combined with a thresholding mechanism on histogram counts) is sufficient to ensure DP.

2 Preliminaries

We address the federated scenario where there are n users and the i^{th} user holds a value x_i. The collection of all user inputs defines a dataset D. Our goal is to construct a histogram of the frequency distribution according to a fixed set of buckets B. We assume each input x_i is already mapped into its corresponding bucket, so that $x_i \in [B]$. We will describe a randomized mechanism, M, that can process a dataset D to give a distribution over output histograms $H^\dagger$. The objective is to ensure that a sampled output histogram H from $H^\dagger$ is close to the true histogram H^*, while ensuring that the output meets (ε, δ)-differential privacy (Dwork and Roth, 2014). Formally, we require

$$\Pr[M(D) = H] \leq \exp(\varepsilon) \Pr\left[M\left(D'\right) = H\right] + \delta$$

for neighboring inputs D, D' that differ in the data held by one individual. As usual, we expect δ to be small, typically much less than $1/n$.

Privacy and computational model: Our mechanism is designed to operate in a federated (distributed) setting that uses a hub-and-spoke communication topology. A central server samples a set of users to privately disclose data held on their devices back to it. The server then combines this information before reporting results to an analyst. The server is assumed to be unbiased and benign in sampling these devices. For example, we do not consider differencing/sybil attacks by the server. However, we assume that the server is "honest-but-curious" about all disclosures made to it. If sampled users communicate with the central server, it is through an anonymous channel that obscures their identifying attributes. User disclosures are routed through a "secure aggregator" which performs simple aggregation (adding, thresholding etc.) before disclosing aggregates to the server. Such a secure aggregator typically provides the following guarantees:

1. Its computational state depends only on input disclosures from users and known code;
2. Its computational state is never observed;
3. Only the final aggregate is eventually disclosed as an output.

Such guarantees can either by made mathematically via carefully designed protocols, or hardware solutions like Intel's SGX. In this work, disclosures are aggregated to build their frequency histogram. Buckets with frequency below a threshold are reset to zero. Final aggregates have formal DP guarantees and are disclosed to the analyst. This model sits between the shuffle and central DP model: it securely prunes histograms using a threshold, but no noise is added and the server is untrusted.

Related work. Histograms are one of the most heavily studied tasks in differential privacy (DP). Initial DP results added independent Laplace noise to each entry of the exact histogram (Dwork, 2006; Dwork and Roth, 2014). For multi-dimensional data, histograms of low-degree marginal distributions can be created via noise addition to the Hadamard transform of the data (Barak et al., 2007). In the local model of DP for histograms, users add noise to their data independently before disclosure. Here "frequency oracles" are used, and frequent items are extracted from the input (Bassily and Smith, 2015; Wang et al., 2017; Erlingsson et al., 2014; Apple, 2017; Bassily et al., 2020). In the shuffle model, messages from individuals are anonymized by a "shuffler", so the analyst sees only the multiset of messages received without attribution (Erlingsson et al., 2020; Ghazi et al., 2021). Under shuffling, for a fixed privacy level ε, accuracy comparable to the central DP case is achievable by introducing a little random noise per user (Balcer and Cheu, 2020; Li et al., 2020). Similarly, quantiles are found via "hierarchical histograms" in local and shuffle models (Qardaji et al., 2013; Xiao et al., 2010; Cormode et al., 2019; Ghazi et al., 2021).

It is well-known that sampling can amplify DP guarantees when combined with a DP mechanism on the sample: Balle et al. (2020) show results for Poisson sampling, and fixed-size sampling with and without replacement, while Imola and Chaudhuri (2021) study privacy amplification when sampling according to differentially private parameters. By contrast, we consider mechanisms where sampling in isolation (with a threshold) provides the DP guarantee directly. The closest work to ours is recent work on Federated Heavy Hitters discovery (Zhu et al., 2020), which uses sampling and thresholding to discover the set of heavy hitters. The key advance in our work is to show that we can output the

sampled frequencies as well as the discovered items, and hence produce private histograms. We also provide accuracy guarantees. Our work complements other efforts in the federated setting to achieve privacy guarantees with a restricted set of operations—for instance, Kairouz et al. (2021) seek to perform federated learning via noise addition *without* sampling.

3 Histograms

In the (B-bucket) histogram problem, each client i holds a single item x_i corresponding to a single bucket $b_i \in [B]$, and our aim is to produce a private histogram of item frequencies, such that a frequency associated with b in the private histogram approximates its frequency over the federated dataset.

The algorithm is based on Bernoulli sampling. Each client out of n is sampled with probability $p_s = m/n$, so the expected size of the sample is m (in the appendix, we discuss different ways to implement this sampling). Our subsequent analysis will set an upper bound on the sample size m in order to give a required privacy guarantee. The algorithm makes use of a threshold θ, so that buckets whose sampled counts are at least θ are reported in the histogram, and the rest are omitted from the histogram. Note then that the mechanism introduces no spurious items into the output: any item which is not present in the input can not appear in the output histogram. In addition, the costs of the algorithm are independent of the dimensionality of the underlying histogram, B.

Theorem 1. *The resulting histogram obeys (ε, δ)-differential privacy, for $\delta = O(\exp(-\theta))$ and $\varepsilon = O(\frac{m}{n}\ln(1/\delta)) \leq 1$.*

We provide formal proofs in the appendix, for completeness. The histogram produced by the mechanism is ultimately based on sampling and pruning, so for any prefix whose frequency is sufficiently above the pruning threshold, then its frequency within the histogram is an (almost) unbiased estimate of its true frequency. There is a small gap, since even for an item 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 (in which case we do not report the item).

Probability of omitting a heavy hitter: We first consider the probability that a frequent item is not reported by the algorithm.

Lemma 2. *The sample and threshold histogram protocol omits an item whose true frequency is W (where $\frac{Wm}{n} > \theta$) with probability at most $\exp(-(\frac{Wm}{n} - \theta)^2 \cdot n/2Wm)$.*

When $\frac{Wm}{n}$ is sufficiently bigger than θ, this gives a very strong probability. For example, consider the case $n = 10^6$, $\varepsilon = 1$, and we set $\theta = 20$ to obtain a δ of 10^{-8}. Under these settings, the probability that an item which occurs 0.1% of the time fails to be detected with probability less than 10^{-7}.

Frequency estimation bounds. More generally, we can use the frequency of any item in the histogram to estimate its true occurrence rate.

Lemma 3. *We can estimate the frequency of any item whose proportion in the federated dataset is ϕ within γ relative error with probability $O(\exp(-\gamma^2\phi m))$.*

Remark. It is instructive to compare these bounds to those that hold for the shuffle model. We observe that in practical federated computing settings, the server can contact only a fixed size cohort of m clients out of a much larger (unknown) population n. If this sampling is uniform, we obtain error $O(1/\sqrt{m})$ to estimate the proportion of the population in each bucket. A shuffle-based approach to histograms due to Balcer and Cheu (2020) adds appropriately parameterized Bernoulli random noise to reports from m clients yields (ε, δ)-DP, with additional error that scales as $O(\frac{1}{\varepsilon^2 m}\log(1/\delta))$ for $\varepsilon \leq 1$, provided m is large enough. For our mechanism, we obtain a bound on the estimate of any frequency with error from rounding small values down to zero, which is bounded by $O(\theta/m) = O(\ln(1/\delta)/m) = O(\ln^2(1/\delta)/(\varepsilon n))$. We observe that results in both the shuffle and sample-and-threshold paradigms *both* incur the same sampling error of $O(1/\sqrt{m})$. Then shuffling introduces additional noise of $O(\frac{1}{\varepsilon^2 m}\log(1/\delta))$, whereas sample-and-threshold incurs zero additional noise on items that exceed the θ threshold, and at most $O(\ln(1/\delta)/m)$ on small items. Hence, we argue that when shuffling implicitly samples m users from the population, the sample-and-threshold approach has superior error guarantees.

4 Heavy hitters and Quantiles

We sketch how to use the basic histogram protocol to find the heavy hitters and quantiles of the input.

Heavy hitters. A result for heavy hitters follows immediately from Theorem 1: we materialize the histogram of the input items, and report those items as heavy whose frequency in the same exceeds a given threshold fraction ϕ. This compares favourably with the approach of Zhu et al. (2020), which proceeds over L rounds, and incurs a cost of L in the privacy parameters ε and δ. In addition, the histogram-based protocol provides estimated frequencies for each heavy hitter, whose accuracy is guaranteed by Lemma 3. The motivation for having L rounds given by Zhu et al. (2020) is to reduce the exposure of the server to private information: they only observe prefixes from clients that extend shorter prefixes that are already known to be popular. However, this does not impact on the formal DP properties of the output.

Single quantiles via interactive search. Given client inputs which fall in the range $[0,1]$, we seek a value f such that the fraction of clients whose value is below f is (approximately) ϕ. The quantile query can be carried out by a binary search: we begin by creating a histogram with buckets $[0,\frac{1}{2}],[\frac{1}{2},1]$, and recursively try different split points $[0,t],[t,1]$ until we obtain a result with approximately a ϕ fraction of points in the first bucket, at which point we can report t as the ϕ-quantile. Provided ϕ is sufficiently larger than θ/m (and smaller than $1-\theta/m$), then we are unlikely to hit any cases where a bucket count is removed. As a result, the error will primarily the error from sampling, which is $O(1/\sqrt{m})$, plus the error from rounding, which is 2^{-h} if we perform h steps of binary search. That is, we find a result t such that there is a point $t \pm 2^{-h}$ that dominates a $\phi \pm O(1/\sqrt{m})$ fraction of the input. The privacy guarantee is $\varepsilon = O(hm\ln(1/\delta)/n)$, from the composition of h queries. This approach is very effective for single queries, but is less desirable when we have a large number of quantile queries to answer in parallel, in which case the hierarchical histogram approach is preferred.

Quantiles via hierarchical histograms. A common technique to answer quantile and range queries in one-dimension is to make use of hierarchical histograms: histograms with geometrically decreasing bucket sizes, so that any range can be expressed as the union of a small number of buckets. In more detail, assume again that each client has an input value in the range $[0,1]$ (say), and we can interpret these as prefixes, corresponding to subranges. Each of L histograms partitions the range $[0,1]$ with increasingly fine buckets, so that the histogram at level $\ell+1$ refines the buckets of the histogram at level ℓ by a constant factor. The privacy guarantee for this summary follows from the the L-fold composition of a differentially private mechanism, giving $(L\varepsilon, Lp_\theta)$-differential privacy via basic composition. Any range query can then be answered by greedily summing buckets that are fully contained in the range, so any range query is answered by combining the counts of $O(L)$ buckets. The error in answering a range query is bounded by $O(L\theta/m)$ additive error, arising from omitting at most $O(L)$ counts due to thresholding. Quantile queries are then addressed by binary searching for the prefix range that contains the target quantile value.

5 Concluding Remarks

In this paper, we have shown how the sample-and-threshold approach can be applied to the fundamental problem of private histogram computation, and related tasks like heavy hitter and quantile estimation. As with other works on private histograms, we assume that the bucket boundaries of the histogram are given. Adaptive division of the histogram buckets is possible, with privacy guarantees resulting from basic (or advanced) composition results. Nevertheless, this approach can give poor results in extreme cases, such as when the bulk of the data resides in a very small fraction of the input domain. In the full version of this work, we will show experimental results that confirm the predicted accuracy of this approach.

It is natural to consider what other computations might benefit from this sample-and-threshold approach. Direct application of the sample-and-threshold technique makes sense when many users hold copies of the same value. Hence, it is not well-suited to questions like finding sums and means of general distributions, unless we additionally apply some rounding and noise addition to input values. The approach may be of value for more complex computations, such as clustering or outlier removal, where dropping rare items is a benefit, or tasks where we seek to discover descriptions of patterns in the data that have large support.

References

Acharya, J., Sun, Z., and Zhang, H. (2019). Hadamard response: Estimating distributions privately, efficiently, and with little communication. In *The 22nd International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1120–1129. PMLR.

Apple (2017). Apple differential privacy technical overview. `https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf`, last accessed 19/07/21.

Balcer, V. and Cheu, A. (2020). Separating local & shuffled differential privacy via histograms. In *1st Conference on Information-Theoretic Cryptography, ITC 2020, June 17-19, 2020, Boston, MA, USA*, volume 163 of *LIPIcs*, pages 1:1–1:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Balle, B., Barthe, G., and Gaboardi, M. (2020). Privacy profiles and amplification by subsampling. *J. Priv. Confidentiality*, 10(1).

Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., and Talwar, K. (2007). Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 273–282. ACM.

Bassily, R., Nissim, K., Stemmer, U., and Thakurta, A. (2020). Practical locally private heavy hitters. *J. Mach. Learn. Res.*, 21:16:1–16:42.

Bassily, R. and Smith, A. D. (2015). Local, private, efficient protocols for succinct histograms. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 127–135. ACM.

Bonawitz, K. A., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Overveldt, T. V., Petrou, D., Ramage, D., and Roselander, J. (2019). Towards federated learning at scale: System design. In *Proceedings of Machine Learning and Systems*. mlsys.org.

Cormode, G., Kulkarni, T., and Srivastava, D. (2019). Answering range queries under local differential privacy. *Proc. VLDB Endow.*, 12(10):1126–1138.

Dwork, C. (2006). Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Song, S., Talwar, K., and Thakurta, A. (2020). Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *CoRR*, abs/2001.03618.

Erlingsson, Ú., Pihur, V., and Korolova, A. (2014). RAPPOR: randomized aggregatable privacy-preserving ordinal response. In Ahn, G., Yung, M., and Li, N., editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1054–1067. ACM.

Ghazi, B., Golowich, N., Kumar, R., Pagh, R., and Velingker, A. (2021). On the power of multiple anonymous messages: Frequency estimation and selection in the shuffle model of differential privacy. In *Advances in Cryptology - EUROCRYPT*, volume 12698 of *Lecture Notes in Computer Science*, pages 463–488. Springer.

Imola, J. and Chaudhuri, K. (2021). Privacy amplification via bernoulli sampling. *CoRR*, abs/2105.10594.

Kairouz, P., McMahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. (2021). Practical and private (deep) learning without sampling or shuffling. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5213–5225. PMLR.

Li, X., Liu, W., Chen, Z., Huang, K., Qin, Z., Zhang, L., and Ren, K. (2020). DUMP: A dummy-point-based framework for histogram estimation in shuffle model. *CoRR*, abs/2009.13738.

McKenna, R., Miklau, G., and Sheldon, D. (2021). Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *CoRR*, abs/2109.04978.

Qardaji, W. H., Yang, W., and Li, N. (2013). Understanding hierarchical methods for differentially private histograms. *Proc. VLDB Endow.*, 6(14):1954–1965.

Wang, T., Blocki, J., Li, N., and Jha, S. (2017). Locally differentially private protocols for frequency estimation. In Kirda, E. and Ristenpart, T., editors, *26th USENIX Security Symposium, USENIX Security*, pages 729–745. USENIX Association.

Xiao, X., Wang, G., and Gehrke, J. (2010). Differential privacy via wavelet transforms. In *Proceedings of the 26th International Conference on Data Engineering, ICDE*, pages 225–236. IEEE Computer Society.

Xu, J., Zhang, Z., Xiao, X., Yang, Y., and Yu, G. (2012). Differentially private histogram publication. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, pages 32–43. IEEE Computer Society.

Zhu, W., Kairouz, P., McMahan, B., Sun, H., and Li, W. (2020). Federated heavy hitters discovery with differential privacy. 108:3837–3847.

A Omitted Proofs

We provide proofs for the claimed properties of the sample-and-threshold histogram mechanism. We begin with some bounds on the number of sampled copies of an item.

Lemma 4. *The probability that the number of samples of an item 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

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

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

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:

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

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.

Lemma 5. *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

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

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

Proof of Theorem 1. Consider the treatment of an item x between two neighboring inputs D and D'. If $f_x = f'_x$, i.e., the number of copies of x is the same in both inputs, then x is treated identically in both cases. Otherwise, wlog we are looking at an x such that $f_x = f'_x + 1 = k+1$. We condition on the event that the number of samples of the item x is not more than θ times its expectation. Call this event E. By Lemma 4, event E holds except with probability $p_\theta = \exp(-(\theta-1)^2/(\theta+1)) = O(\exp(-\theta))$. We condition on E holding, and just account for this probability in our final reckoning.

Suppose that the count of x for D is less than n/m. Then, by our assumption of E, D will not sample θ copies of x, 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 x ($k+1$ for D) is at least n/m, and by our assumption D samples at most $\tau \leq \theta m(k+1)/n$ copies of x. Then, by Lemma 5, we can state that for the mechanism M, the probability of seeing a given output histogram H satisfies:

$$\begin{aligned}\frac{\Pr[M(D)=H|E]}{\Pr[M(D')=H|E]} &\leq \frac{k+1}{k+1-\tau} \leq \frac{k+1}{k+1-\theta(k+1)m/n} \\ &= \frac{n}{n-\theta m}\end{aligned} \tag{3}$$

We will assume that $m = c_\varepsilon \frac{n}{\theta}$ for a constant $c_\varepsilon < 1-1/e$ that depends on ε. The effect is to ensure that the sample size m is a small fraction of n. Substituting this assumption in (3), we conclude

$$\frac{\Pr[M(D)=H|E]}{\Pr[M(D')=H|E]} = \frac{n}{n-c_\varepsilon n} = \frac{1}{1-c_\varepsilon} := \exp(\varepsilon) \tag{4}$$

That is, except with probability p_θ, we have ε-differential privacy. Rearranging, we set $c_\varepsilon = 1-\exp(-\varepsilon)$. For small ε, we can approximate $c_\varepsilon = \varepsilon$. We can also write

$$\varepsilon = \ln\frac{1}{1-c_\varepsilon} = \ln\left(1+\frac{c_\varepsilon}{1-c_\varepsilon}\right) = \ln\left(1+\frac{m\theta/n}{1-c_\varepsilon}\right) \leq \ln\left(1+\frac{em\theta}{n}\right)$$

where the last step uses $c_\varepsilon \leq 1-1/e$. This proves $(O(\frac{m\theta}{n}), O(\exp(-\theta))$-differential privacy, as claimed. □

Proof of Lemma 2. For an item with (absolute) frequency W out of the n input items, it is reported 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

item. Now, the probability of each sample picking the item 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]

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

□

Proof of Lemma 3. 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 $\mu = \phi m \geq \phi\frac{\varepsilon n}{\theta} = \frac{3}{\gamma^2}\ln(1/2\beta)$. □

Fixed sized sampling. For practical efficiency, we would often like to work with a fixed size sample. However, the above histogram protocol performs Poisson sampling instead. The reason is that if the fixed size of the sample, m, 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 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.

Lemma 6. *Sampling $m + O(\sqrt{m})$ clients is sufficient to apply the sample-and-threshold mechanism, with high probability.*

Proof. Observe that, with high probability, the size of the (Poisson) 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 the aggregator. □

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