Frequency Estimation Under Multiparty Differential Privacy: One-shot and Streaming

Ziyue Huang, Yuan Qiu, Ke Yi, Graham Cormode
{zhuangbq, yqiuac, yike}@cse.ust.hk, g.cormode@warwick.ac.uk

Abstract

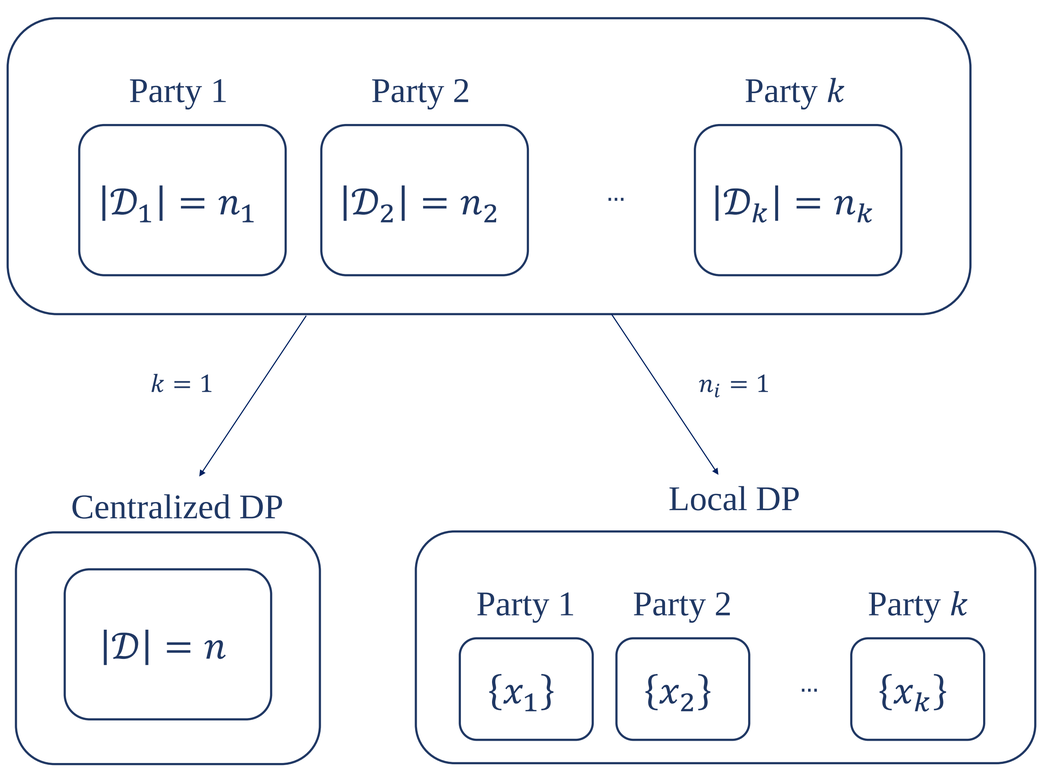
We study the fundamental problem of frequency estimation under both **privacy** and **communication** constraints, where the data is distributed among k parties. We consider two application scenarios:

(1) **one-shot**, where the data is static and the aggregator conducts a one-time computation; and

(2) **streaming**, where each party receives a stream of items over time and the aggregator continuously monitors the frequencies.

We adopt the model of multiparty differential privacy (MDP), which is more general than local differential privacy (LDP) and (centralized) differential privacy. Our protocols achieve optimality (up to logarithmic factors) permissible by the **more stringent** of the two constraints. In particular, when specialized to the ε-LDP model, our protocol achieves an error of $\sqrt{k}/(e^{\Theta(\varepsilon)}-1)$ using $O\left(k\max\left\{\varepsilon, \log\frac{1}{\varepsilon}\right\}\right)$ bits of communication and $O(k\log u)$ bits of public randomness, where u is the size of the domain.

Multiparty Differential Privacy



Results for One-shot Estimation

With $\tilde{O}(ks)$ total communication, our protocol achieves error

$$\tilde{O}\left(\frac{N}{\sqrt{k}s}+\frac{\sqrt{k}}{e^{\widetilde{\Theta}(\varepsilon)}-1}\right)$$

Communication-dependent | Privacy bound

$N=\sum_{i=1}^{k} n_i$ is the total number of items

s controls the communication-utility trade-off

Under LDP, we use less communication to achieve optimal error

$$\tilde{O}\left(\frac{\sqrt{k}}{e^{\varepsilon/4}-1}\right)$$

	Ours	Basic RAPPOR	PI-RAPPOR
Comm.	$k\max\left\{\varepsilon, \log\frac{1}{\varepsilon}\right\}$	ku	$k\max\left\{\log u, \varepsilon, \log\frac{1}{\varepsilon}\right\}$

General One-Shot Protocol

Party Side

- Build a count sketch of $s_i=\left\lceil\frac{ksn_i}{N}\right\rceil$ columns
- Perturb each counter by adding geometric noise
- Send the noisy count sketch to the aggregator

Aggregator Side

- Estimate the frequency from each count sketch, then taking sum

Improvements by Frequency Separation

Each party privately separates local heavy/light hitters

- For heavy hitters $j\in S^{hi}$
  - Perturb its frequency $x_{i,j}$ by adding a geometric noise
  - Send the perturbed count using **importance sampling**
    - Send $\left(j,\frac{\hat{x}_{i,j}}{p}\right)$ with probability $p=\min\left\{\frac{ks\cdot|\hat{x}_{i,j}|}{N},1\right\}$
- For light hitters $j\in S^{lo}$
  - Apply our general count sketch based method

Results for Streaming Estimation

With $\tilde{O}\left(ks\cdot\frac{n}{w}\right)$ total communication, our protocol achieves error

$$\tilde{O}\left(\frac{\sqrt{k}w}{s\cdot\min\{\varepsilon,1\}}+\frac{\sqrt{k}}{e^{\widetilde{\Theta}(\varepsilon)}-1}\right)$$

n is the total number of time steps

w is the length of the sliding window

When $\varepsilon=O(1)$, it matches the one-shot error ($w=n=N/k$)

When $\varepsilon=\Theta(1)$, there are two improvements over PDCH

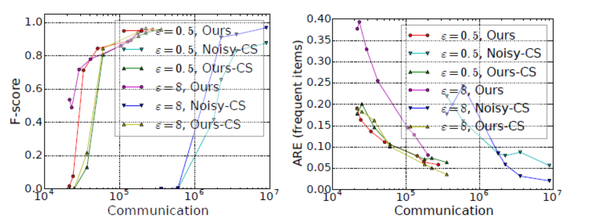
- The error improves from $\tilde{O}\left(\frac{kw}{s}\right)$ to $\tilde{O}\left(\frac{\sqrt{k}w}{s}\right)$
- There is no upper bound on s, setting $s:=w$ achieves $\tilde{O}(\sqrt{k})$

Full-Stream Protocol

Divide the stream into epochs, each epoch contains $\frac{n}{s}$ time steps

- Intra-epoch: Sample time steps and apply HRR
- Inter-epoch: Build a dyadic structure using the one-shot algorithm as building blocks

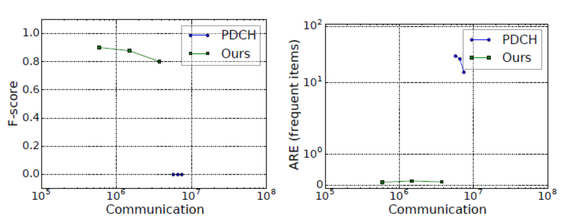
Experiments



(a) F-score (b) ARE (frequent items)

Figure 6: Accuracy vs. Communication on AOL.

One-shot



(a) F-score (b) ARE (frequent items)

Figure 13: Accuracy vs. communication on AOL

Streaming