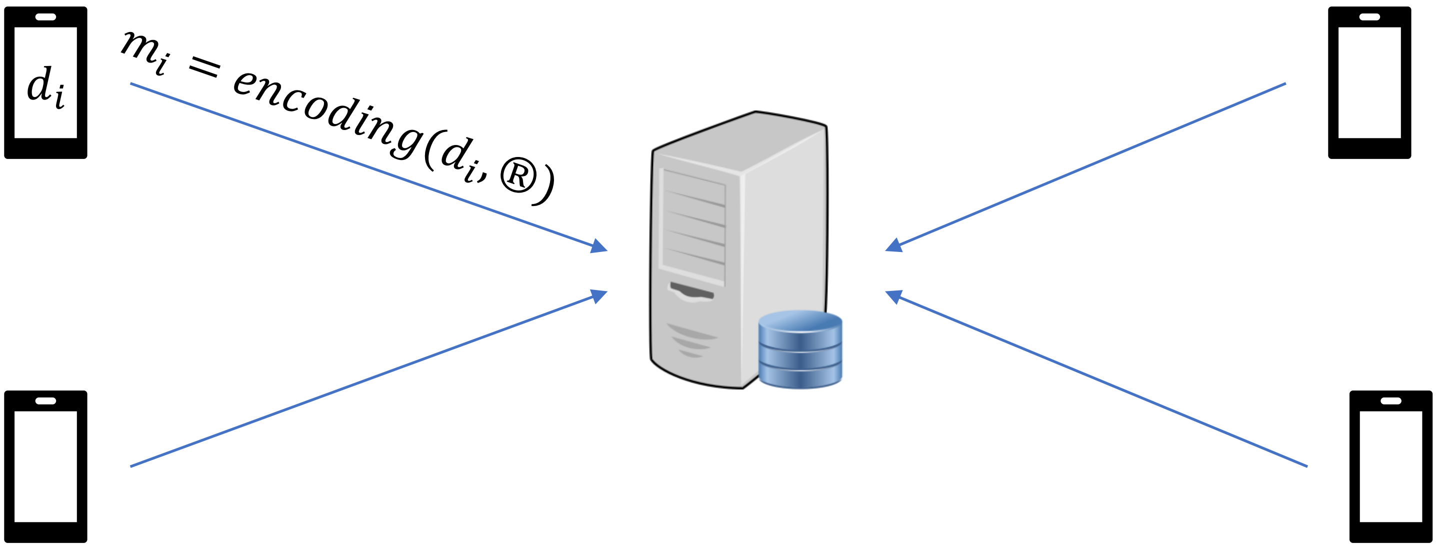
Private frequency estimation

- n users, each holding an item from a universe of size k
- Items: words in private documents, browser settings
- Goal: construct the frequency histogram (how many times each value occurs) as accurately as possible while satisfying ϵ-local privacy

Private frequency estimation



- ϵ-private if
  $\Pr[m_i = m | d_i = d] \leq e^{\epsilon} \Pr[m_i = m | d_i = d'] \ \forall m, d, d'$
- Minimize error due to privacy requirement
  - Our metric: sum of variance of all histogram entries
- Minimize communication
- Efficient encoding for users and decoding for server

Previous works

Algorithm	Communication	Error/Variance	Server time
Randomized Resp. [Warner '65]	$\log k$	$\frac{n(2e^{\epsilon}+k)}{(e^{\epsilon}-1)^2}$	$n+k$
RAPPOR [EPK '14]	k	$\frac{4ne^{\epsilon}}{(e^{\epsilon}-1)^2}$	nk
Subset Selection [YB '17, WHN+ '19]	$\frac{k\epsilon}{e^{\epsilon}+1}$	$\frac{4ne^{\epsilon}}{(e^{\epsilon}-1)^2}$	$\frac{kn}{e^{\epsilon}+1}$
PI-RAPPOR [FT '21]	$\log k$	$\frac{4ne^{\epsilon}}{(e^{\epsilon}-1)^2}$	$\min\left(n+k^2, \frac{kn}{e^{\epsilon}+1}\right)$
Hadamard Resp. [ASZ '19]	$\log k$	$\frac{36ne^{\epsilon}}{(e^{\epsilon}-1)^2}$	$n+k\log k$
Recursive Hadamard Resp. [CKO '20]	$\log k$	$\frac{8ne^{\epsilon}}{(e^{\epsilon}-1)^2}$	$n+k\log k$

Previous works

Great	Good	Poor

Algorithm	Communication	Error/Variance	Server time
Randomized Resp. [Warner '65]	$\log k$	$\frac{n(2e^\epsilon + k)}{(e^\epsilon - 1)^2}$	$n + k$
RAPPOR [EPK '14]	k	$\frac{4ne^\epsilon}{(e^\epsilon - 1)^2}$	nk
Subset Selection [YB '17, WHN+ '19]	$\frac{k\epsilon}{e^\epsilon + 1}$	$\frac{4ne^\epsilon}{(e^\epsilon - 1)^2}$	$\frac{kn}{e^\epsilon + 1}$
PI-RAPPOR [FT '21]	$\log k$	$\frac{4ne^\epsilon}{(e^\epsilon - 1)^2}$	$\min\left(n + k^2, \frac{kn}{e^\epsilon + 1}\right)$
Hadamard Resp. [ASZ '19]	$\log k$	$\frac{36ne^\epsilon}{(e^\epsilon - 1)^2}$	$n + k \log k$
Recursive Hadamard Resp. [CKO '20]	$\log k$	$\frac{8ne^\epsilon}{(e^\epsilon - 1)^2}$	$n + k \log k$

Why asymptotic optimal error is not enough?

- Cannot improve error with more resources (more time, more memory)
- In many applications (finding new words, malicious homepage domains, etc), frequency estimation is used to identify popular items above the error floor
- Frequency distribution tends to have heavy tail
- Halving the error can result in a constant factor more items being discovered

Our contribution

Algorithm	Communication	Error/Variance	Server time
Projective Geometry Response (PGR)	$\log k$	$\frac{4ne^{\epsilon}}{(e^{\epsilon}-1)^2}$	$n + ke^{\epsilon}\log k$
Hybrid Projective Geometry Resp.	$\log k$	$\left(1+\frac{1}{q-1}\right)\frac{4ne^{\epsilon}}{(e^{\epsilon}-1)^2}$	$n + kq\log k$

- PGR has optimal error, low communication, and fast decoding
- Hybrid PGR gives tradeoff between error and decoding time

Framework for the local randomizer [ASZ ’19]

- Input-message matrix (k inputs, M messages)

	Message 1	Message 2	Message 3	Message 4
Input 1				
Input 2				
Input 3				

- Each input v corresponds to a set of “preferred” messages $S_v \subset U$ (U: set of all messages)
  - Each $m \in S_v$ is sent with probability $e^{\epsilon} p$
  - Each $m \notin S_v$ is sent with probability p
- Trivially satisfy privacy constraints

Construction using projective geometry

- Field F of size q, vector space F^t
- A vector is canonical if the first non-zero is 1
- Inputs and messages are canonical vectors in F^t
- $k = \frac{q^t - 1}{q - 1}$ different input values, same for messages
- For input vector u, the set S_u consists of canonical vectors orthogonal to u (a subspace of F^t)
- Each set has size $c_{set} = \frac{q^{t-1} - 1}{q - 1}$
- Two different sets have intersection of size $c_{int} = \frac{q^{t-2} - 1}{q - 1}$
- The regular sizes and $\frac{k}{c_{set}} \approx \frac{c_{set}}{c_{int}}$ make the variance near optimal

Reconstructing frequency histogram

- To estimate the number of users with input u, need to count the number of messages in S_u (canonical vectors in a subspace)
- Structure of subspaces allows for fast dynamic programming algorithm

Experiment

- Zipfian distribution: PMF(i) $\sim 1/i^s$
- HPG: "hybrid" version of algorithm trading off time and error