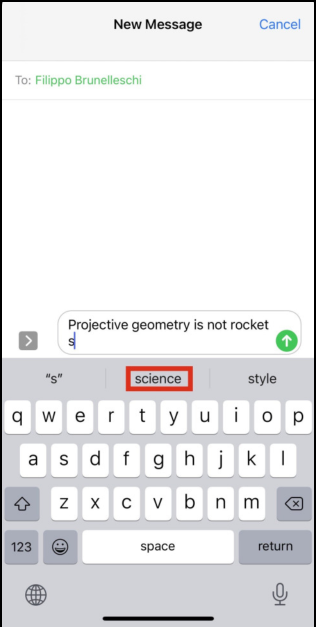
Private frequency estimation via Projective Geometry

Jelani Nelson

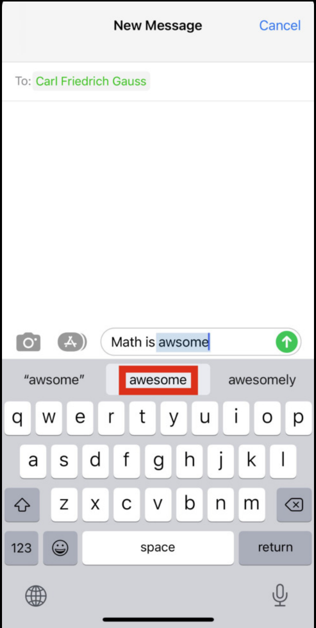
UC Berkeley

October 3, 2022

(joint work with Vitaly Feldman, Huy Le Nguyen, Kunal Talwar)



autocomplete



spell-correct

(12) **United States Patent**
Thakurta et al.

(10) **Patent No.:** **US 9,594,741 B1**
(45) **Date of Patent:** **Mar. 14, 2017**

(54) **LEARNING NEW WORDS**

(71) Applicant: **Apple Inc.**, Cupertino, CA (US)

(72) Inventors: **Abhradeep Guha Thakurta**, San Jose, CA (US); **Andrew H. Vyrros**, San Francisco, CA (US); **Umesh S. Vaishampayan**, Santa Clara, CA (US); **Gaurav Kapoor**, Santa Clara, CA (US); **Julien Freudiger**, Mountain View, CA (US); **Vivek Rangarajan Sridhar**, Sunnyvale, CA (US); **Doug Davidson**, Palo Alto, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **15/275,356**

(22) Filed: **Sep. 24, 2016**

Related U.S. Application Data

(60) Provisional application No. 62/348,988, filed on Jun. 12, 2016, provisional application No. 62/371,657, filed on Aug. 5, 2016.

(51) **Int. Cl.**
G06F 17/27 (2006.01)
G06N 99/00 (2010.01)

(52) **U.S. Cl.**
CPC ***G06F 17/2765*** (2013.01); ***G06F 17/2705***

(58) **Field of Classification Search**
USPC 704/1–10, 257, 270.1
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,140,332 B2 *	3/2012	Itoh	G06F 17/2735	704/10
8,185,376 B2 *	5/2012	Chu	G06F 17/275	704/8
2005/0256715 A1 *	11/2005	Okimoto	G06F 17/2715	704/257
2014/0278357 A1 *	9/2014	Horton	G06F 17/277	704/9

* cited by examiner

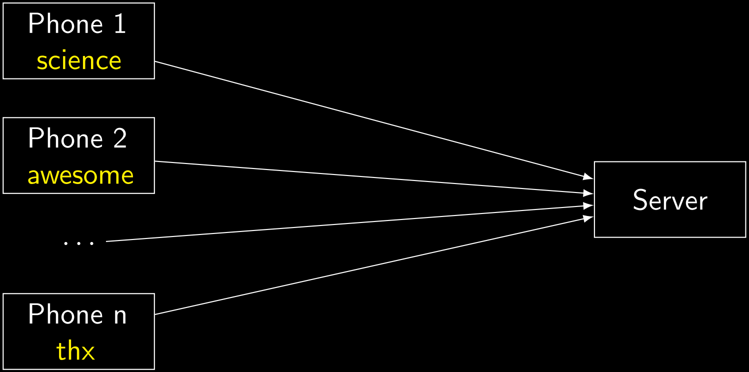
Primary Examiner — Abul Azad
(74) *Attorney, Agent, or Firm* — Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

Systems and methods are disclosed for a server learning new words generated by user client devices in a crowdsourced manner while maintaining local differential privacy of client devices. A client device can determine that a word typed on the client device is a new word that is not contained in a dictionary or asset catalog on the client device. New words can be grouped in classifications such as entertainment, health, finance, etc. A differential privacy system on the client device can comprise a privacy budget for each classification of new words. If there is privacy budget available for the classification, then one or more new terms in a classification can be sent to new term learning server, and the privacy budget for the classification reduced. The privacy budget can be periodically replenished.

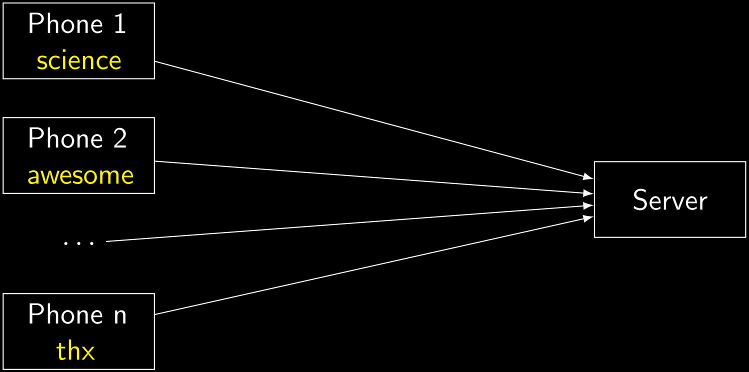
(57) **ABSTRACT**

Systems and methods are disclosed for a server learning new words generated by user client devices in a crowdsourced manner while maintaining local differential privacy of client devices. A client device can determine that a word typed on the client device is a new word that is not contained in a dictionary or asset catalog on the client device. New words can be grouped in classifications such as entertainment,



Server wants to know word distribution amongst phones/devices

$f_x :=$ how many devices just texted the word "x"?



Server wants to know word distribution amongst phones/devices
$f_x :=$ how many devices just texted the word "x"?

Simple (?). Each device sends a copy of all its texts to server.

Constraint: privacy

(do you really want phone manufacturers to read all your texts?)

(57) **ABSTRACT**

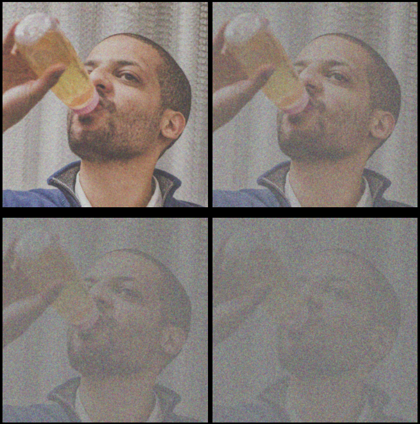
Systems and methods are disclosed for a server learning new words generated by user client devices in a crowdsourced manner while maintaining local differential privacy of client devices. A client device can determine that a word typed on the client device is a new word that is not contained in a dictionary or asset catalog on the client device. New words can be grouped in classifications such as entertainment, health, finance, etc. A differential privacy system on the client device can comprise a privacy budget for each classification of new words. If there is privacy budget available for the classification, then one or more new terms in a classification can be sent to new term learning server, and the privacy budget for the classification reduced. The privacy budget can be periodically replenished.

Basic idea

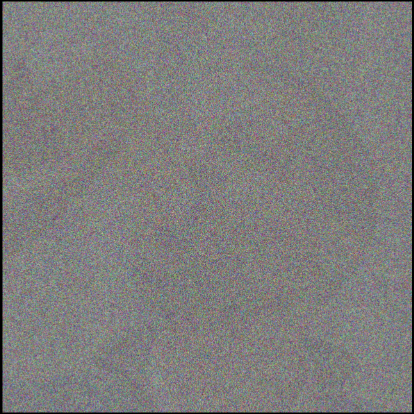
send randomized messages (e.g., add noise)!

Original Image

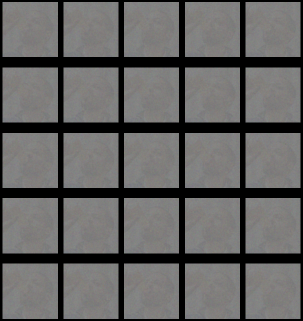
Noisified Versions



Now with lots of noise:



Heavily Noisified Copies



Averaged Image

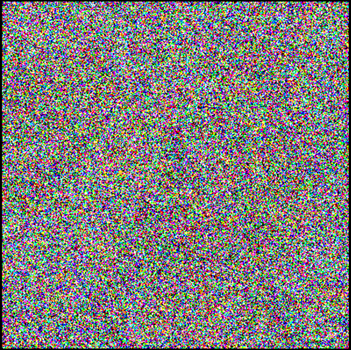
Moral of this story

can have each individual message look like garbage, thus protecting individual privacy, but server can extract useful knowledge by aggregating messages from all devices

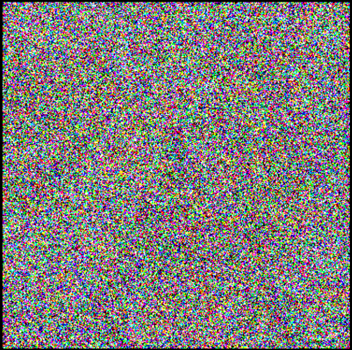
Moral of this story

can have each individual message look like garbage, thus protecting individual privacy, but server can extract useful knowledge by aggregating messages from all devices

But what *exactly* does privacy mean?



Above, applied 'wavelet denoising' to a single noised image
Maybe this isn't so private after all?

Above, applied 'wavelet denoising' to a single noised image
Maybe this isn't so private after all?

Must be careful with the definition!

Local Differential Privacy

Idea: Device i sends *random* message M_i that is only weakly correlated with its data (e.g., its word, or an image, etc.) x_i

Local Differential Privacy

Idea: Device i sends *random* message M_i that is only weakly correlated with its data (e.g., its word, or an image, etc.) x_i

- One individual device's message almost looks like random noise, but server can extract signal from many such messages from different devices in aggregate

Local Differential Privacy

Idea: Device i sends *random* message M_i that is only weakly correlated with its data (e.g., its word, or an image, etc.) x_i

- One individual device's message almost looks like random noise, but server can extract signal from many such messages from different devices in aggregate
- Privacy definition: scheme provides ε-differential privacy [Dwork-McSherry-Nissim-Smith'06] if for all devices i and all possible msgs M, and for all $x \neq x'$,

$$\frac{\mathbb{P}(M_i = M | x_i = x)}{\mathbb{P}(M_i = M | x_i = x')} \leq e^{\varepsilon}.$$

  ε is called the privacy loss ($\varepsilon = 0$ is perfectly private)

  (informally: device would have been almost as likely to send the same exact message even if their data were different)

Two regimes to keep in mind ...

- ε small ($\varepsilon < 1$): $e^{\varepsilon} \approx 1 + \varepsilon$
- ε large (what's usually deployed in practice)

Two regimes to keep in mind . . .

- ε small ($\varepsilon < 1$): $e^{\varepsilon} \approx 1 + \varepsilon$
- ε large (what's usually deployed in practice)

Large ε means *worse privacy*, so why deploy large ε?

Two regimes to keep in mind ...

- ε small ($\varepsilon < 1$): $e^{\varepsilon} \approx 1 + \varepsilon$
- ε large (what's usually deployed in practice)

Large ε means *worse privacy*, so why deploy large ε?

Fundamental **tradeoff** between ...

- **Utility:** quality of the knowledge the server extracts
- **Privacy:** defined in terms of privacy loss ε

Two regimes to keep in mind ...

- ε small ($\varepsilon < 1$): $e^{\varepsilon} \approx 1 + \varepsilon$
- ε large (what's usually deployed in practice)

Large ε means *worse privacy*, so why deploy large ε?

Fundamental **tradeoff** between ...

- **Utility:** quality of the knowledge the server extracts
- **Privacy:** defined in terms of privacy loss ε

Small ε requires too much utility loss to be usable. Silver lining: *shuffling* improves privacy [BEM+17], [CSU+19], [EFM+19], [BBGN19], [BKM+20], [FMT21].

Before going further: our *particular* problem for today

Before going further: our *particular* problem for today

Each device holds i some data x_i from a set $\{1, \ldots, k\}$.
This implies a *frequency histogram*, $f_x := (\#\text{devices with } x_i = x)$

Before going further: our *particular* problem for today

Each device holds i some data x_i from a set $\{1, \dots, k\}$.
This implies a *frequency histogram*, $f_x := (\#\text{devices with } x_i = x)$

Server wants to recover $\tilde{f}$ that is *close* to f
(e.g., small **Mean Squared Error (MSE)** $\frac{1}{k}\sum_{x=1}^{k}(f_x - \tilde{f}_x)^2$)

Things to optimize

Privacy and utility are just two things to consider; the full list:

- Privacy: defined already (ε = privacy loss)
- Utility: if `query`(x) returns $\tilde{f}_x$, want $|f_x - \tilde{f}_x|$ small

  (we define utility loss as the **MSE**, $\frac{1}{k}\,\mathbb{E}\,\|f - \tilde{f}\|_2^2$)
- Communication: devices each send $b = |M_i|$ bits
- Server time: time server takes to produce $\tilde{f}$ given messages
- Device time: device takes to produce M_i given x_i

Things to optimize

Privacy and utility are just two things to consider; the full list:

- Privacy: defined already ($\varepsilon =$ privacy loss)
- Utility: if $\mathtt{query}(x)$ returns $\tilde{f}_x$, want $|f_x - \tilde{f}_x|$ small
  (we define utility loss as the **MSE**, $\frac{1}{k}\,\mathbb{E}\,\|f - \tilde{f}\|_2^2$)
- Communication: devices each send $b = |M_i|$ bits
- Server time: time server takes to produce $\tilde{f}$ given messages
- Device time: device takes to produce M_i given x_i

Ideally want all five of the above to be small simultaneously.

A simple scheme

RandomizedResponse. Each device sends its true item x with probability $e^{\varepsilon} p$; otherwise sends a uniformly random other item (so that any other item is sent with probability p)

A simple scheme

RandomizedResponse. Each device sends its true item x with probability $e^{\varepsilon}p$; otherwise sends a uniformly random other item (so that any other item is sent with probability p)

$\mathbb{P}(\text{send } \textit{something}) = 1$, so $e^{\varepsilon}p + (k-1)p = 1$
solves to $p = \frac{1}{e^{\varepsilon}+k-1}$

A simple scheme

RandomizedResponse. Each device sends its true item x with probability $e^{\varepsilon} p$; otherwise sends a uniformly random other item (so that any other item is sent with probability p)

$\mathbb{P}(\text{send } \textit{something}) = 1$, so $e^{\varepsilon} p + (k-1)p = 1$
solves to $p = \frac{1}{e^{\varepsilon}+k-1}$

How does the server estimate $\tilde{f}_x \approx f_x$?

A simple scheme

RandomizedResponse. Each device sends its true item x with probability $e^{\varepsilon} p$; otherwise sends a uniformly random other item (so that any other item is sent with probability p)

$\mathbb{P}(\text{send } \textit{something}) = 1$, so $e^{\varepsilon} p + (k-1)p = 1$
solves to $p = \frac{1}{e^{\varepsilon}+k-1}$

How does the server estimate $\tilde{f}_x \approx f_x$?
For each message M_i, add $\alpha + \beta$ to estimate if $M_i = x$, else add β

A simple scheme

RandomizedResponse. Each device sends its true item x with probability $e^{\varepsilon}p$; otherwise sends a uniformly random other item (so that any other item is sent with probability p)

$\mathbb{P}(\text{send } \textit{something}) = 1$, so $e^{\varepsilon}p + (k-1)p = 1$
solves to $p = \frac{1}{e^{\varepsilon}+k-1}$

How does the server estimate $\tilde{f}_x \approx f_x$?
For each message M_i, add $\alpha + \beta$ to estimate if $M_i = x$, else add β

If $x_i = x$: expected contribution is $\alpha e^{\varepsilon} p + \beta$
If $x_i \neq x$: expected contribution is $\alpha p + \beta$

A simple scheme

RandomizedResponse. Each device sends its true item x with probability $e^{\varepsilon}p$; otherwise sends a uniformly random other item (so that any other item is sent with probability p)

$\mathbb{P}(\text{send } \textit{something}) = 1$, so $e^{\varepsilon}p + (k-1)p = 1$
solves to $p = \frac{1}{e^{\varepsilon}+k-1}$

How does the server estimate $\tilde{f}_x \approx f_x$?
For each message M_i, add $\alpha + \beta$ to estimate if $M_i = x$, else add β

If $x_i = x$: expected contribution is $\alpha e^{\varepsilon}p + \beta$
If $x_i \neq x$: expected contribution is $\alpha p + \beta$
Thus want $\alpha e^{\varepsilon} + \beta = 1$, $\alpha p + \beta = 0$; two eqns and two unknowns, solves to $\alpha = \frac{e^{\varepsilon}+k-1}{e^{\varepsilon}-1}$, $\beta = -\frac{1}{e^{\varepsilon}-1}$

A simple scheme

RandomizedResponse. Each device sends its true item x with probability $e^{\varepsilon}p$; otherwise sends a uniformly random other item (so that any other item is sent with probability p)

$\mathbb{P}$(send *something*) $= 1$, so $e^{\varepsilon}p + (k-1)p = 1$
solves to $p = \frac{1}{e^{\varepsilon}+k-1}$

How does the server estimate $\tilde{f}_x \approx f_x$?
For each message M_i, add $\alpha + \beta$ to estimate if $M_i = x$, else add β

If $x_i = x$: expected contribution is $\alpha e^{\varepsilon} p + \beta$
If $x_i \neq x$: expected contribution is $\alpha p + \beta$
Thus want $\alpha e^{\varepsilon} + \beta = 1$, $\alpha p + \beta = 0$; two eqns and two unknowns, solves to $\alpha = \frac{e^{\varepsilon}+k-1}{e^{\varepsilon}-1}$, $\beta = -\frac{1}{e^{\varepsilon}-1}$

Pros: Low communication, and very fast for server and devices
Con: Terrible utility loss (can show)

Another simple scheme

SubsetSelection [Ye, Barg '17]. Each device sends a random subset $S \subset \{1, \ldots, k\}$ of size d. If $x \in S$, S is sent with probability $e^{\varepsilon} p$; else S sent with probability p

Another simple scheme

SubsetSelection [Ye, Barg '17]. Each device sends a random subset $S \subset \{1, \ldots, k\}$ of size d. If $x \in S$, S is sent with probability $e^{\varepsilon} p$; else S sent with probability p

$\mathbb{P}(\text{send } \textit{something}) = 1$, so $e^{\varepsilon} p \binom{k-1}{d-1} + p \binom{k-1}{d} = 1$
solves to $p = \frac{1}{e^{\varepsilon}\binom{k-1}{d-1} + \binom{k-1}{d}}$

Another simple scheme

SubsetSelection [Ye, Barg '17]. Each device sends a random subset $S \subset \{1, \ldots, k\}$ of size d. If $x \in S$, S is sent with probability $e^{\varepsilon}p$; else S sent with probability p

$\mathbb{P}$(send *something*) $= 1$, so $e^{\varepsilon}p\binom{k-1}{d-1} + p\binom{k-1}{d} = 1$
solves to $p = \frac{1}{e^{\varepsilon}\binom{k-1}{d-1}+\binom{k-1}{d}}$

How does the server estimate $\tilde{f}_x \approx f_x$?

Another simple scheme

SubsetSelection [Ye, Barg '17]. Each device sends a random subset $S \subset \{1, \ldots, k\}$ of size d. If $x \in S$, S is sent with probability $e^{\varepsilon} p$; else S sent with probability p

$\mathbb{P}$(send *something*) $= 1$, so $e^{\varepsilon} p \binom{k-1}{d-1} + p \binom{k-1}{d} = 1$
solves to $p = \frac{1}{e^{\varepsilon}\binom{k-1}{d-1} + \binom{k-1}{d}}$

How does the server estimate $\tilde{f}_x \approx f_x$?
For each message M_i, add $\alpha + \beta$ to estimate if $x \in M_i$, else add β

Another simple scheme

SubsetSelection [Ye, Barg '17]. Each device sends a random subset $S \subset \{1, \ldots, k\}$ of size d. If $x \in S$, S is sent with probability $e^{\varepsilon} p$; else S sent with probability p

$\mathbb{P}$(send *something*) $= 1$, so $e^{\varepsilon} p \binom{k-1}{d-1} + p \binom{k-1}{d} = 1$
solves to $p = \frac{1}{e^{\varepsilon} \binom{k-1}{d-1} + \binom{k-1}{d}}$

How does the server estimate $\tilde{f}_x \approx f_x$?
For each message M_i, add $\alpha + \beta$ to estimate if $x \in M_i$, else add β

If $x_i = x$: expected contribution is $\alpha e^{\varepsilon} p \binom{k-1}{d-1} + \beta$
If $x_i \neq x$: expected contribution is $\alpha(e^{\varepsilon} p \binom{k-2}{d-1} + p \binom{k-2}{d}) + \beta$

Another simple scheme

SubsetSelection [Ye, Barg '17]. Each device sends a random subset $S \subset \{1, \ldots, k\}$ of size d. If $x \in S$, S is sent with probability $e^{\varepsilon} p$; else S sent with probability p

$\mathbb{P}(\text{send } \textit{something}) = 1$, so $e^{\varepsilon} p \binom{k-1}{d-1} + p \binom{k-1}{d} = 1$
solves to $p = \frac{1}{e^{\varepsilon}\binom{k-1}{d-1} + \binom{k-1}{d}}$

How does the server estimate $\tilde{f}_x \approx f_x$?
For each message M_i, add $\alpha + \beta$ to estimate if $x \in M_i$, else add β

If $x_i = x$: expected contribution is $\alpha e^{\varepsilon} p \binom{k-1}{d-1} + \beta$
If $x_i \neq x$: expected contribution is $\alpha\left(e^{\varepsilon} p \binom{k-2}{d-1} + p \binom{k-2}{d}\right) + \beta$
As before want first equal 1, second equal 0; two eqns and two unknowns, and can solve for α, β. Gives low MSE for $d \approx \frac{k}{e^{\varepsilon}+1}$.

Another simple scheme

SubsetSelection [Ye, Barg '17]. Each device sends a random subset $S \subset \{1, \ldots, k\}$ of size d. If $x \in S$, S is sent with probability $e^{\varepsilon} p$; else S sent with probability p

$\mathbb{P}$(send *something*) $= 1$, so $e^{\varepsilon} p \binom{k-1}{d-1} + p \binom{k-1}{d} = 1$
solves to $p = \frac{1}{e^{\varepsilon}\binom{k-1}{d-1} + \binom{k-1}{d}}$

How does the server estimate $\tilde{f}_x \approx f_x$?
For each message M_i, add $\alpha + \beta$ to estimate if $x \in M_i$, else add β

If $x_i = x$: expected contribution is $\alpha e^{\varepsilon} p \binom{k-1}{d-1} + \beta$
If $x_i \neq x$: expected contribution is $\alpha(e^{\varepsilon} p \binom{k-2}{d-1} + p \binom{k-2}{d}) + \beta$
As before want first equal 1, second equal 0; two eqns and two unknowns, and can solve for α, β. Gives low MSE for $d \approx \frac{k}{e^{\varepsilon}+1}$.

Pro: Optimal privacy loss/utility loss tradeoff [Ye, Barg'06]
Cons: Terrible communication, server/device runtimes

A meta approach

[Acharya, Sun, Zhang'19]

A meta approach [Acharya, Sun, Zhang'19]

Suppose data $x_i \in \{1, \ldots, k\}$, and there is a "message space" $\mathcal{Y}$

A meta approach [Acharya, Sun, Zhang'19]

Suppose data $x_i \in \{1, \ldots, k\}$, and there is a "message space" $\mathcal{Y}$

- Associate with each x some $S_x \subset \mathcal{Y}, |S_x| = s$
- Suppose $\{S_x\}_{x \in \mathcal{X}}$ is such that $\forall x \neq x', |S_x \cap S_{x'}| = \ell$

A meta approach [Acharya, Sun, Zhang'19]

Suppose data $x_i \in \{1, \ldots, k\}$, and there is a "message space" $\mathcal{Y}$

- Associate with each x some $S_x \subset \mathcal{Y}, |S_x| = s$
- Suppose $\{S_x\}_{x \in \mathcal{X}}$ is such that $\forall x \neq x', |S_x \cap S_{x'}| = \ell$
- **Mechanism:** For any $y \in \mathcal{Y}$, send message $M = y$ with probability p if $y \notin S_x$, and with probability $e^{\varepsilon} p$ if $y \in S_x$ (call S_x the *preferred messages for* x)

A meta approach [Acharya, Sun, Zhang'19]

Suppose data $x_i \in \{1, \ldots, k\}$, and there is a "message space" $\mathcal{Y}$

- Associate with each x some $S_x \subset \mathcal{Y}, |S_x| = s$
- Suppose $\{S_x\}_{x \in \mathcal{X}}$ is such that $\forall x \neq x', |S_x \cap S_{x'}| = \ell$
- **Mechanism:** For any $y \in \mathcal{Y}$, send message $M = y$ with probability p if $y \notin S_x$, and with probability $e^{\varepsilon} p$ if $y \in S_x$ (call S_x the *preferred messages for* x)

  Note: $e^{\varepsilon} ps + p(|\mathcal{Y}| - s) = 1$, so $p = \frac{1}{s(e^{\varepsilon} - 1) + |\mathcal{Y}|}$

A meta approach [Acharya, Sun, Zhang'19]

Suppose data $x_i \in \{1, \ldots, k\}$, and there is a "message space" $\mathcal{Y}$

- Associate with each x some $S_x \subset \mathcal{Y}, |S_x| = s$
- Suppose $\{S_x\}_{x \in \mathcal{X}}$ is such that $\forall x \neq x', |S_x \cap S_{x'}| = \ell$
- **Mechanism:** For any $y \in \mathcal{Y}$, send message $M = y$ with probability p if $y \notin S_x$, and with probability $e^\varepsilon p$ if $y \in S_x$ (call S_x the *preferred messages for x*)
  Note: $e^\varepsilon ps + p(|\mathcal{Y}| - s) = 1$, so $p = \frac{1}{s(e^\varepsilon - 1) + |\mathcal{Y}|}$
- Server estimates f_x as
  $\tilde{f}_x = \sum_{i=1}^n (\alpha \cdot [[M_i \in S_x]] + \beta)$ ($[[P]] = 1$ iff P is True; 0 o/w)

A meta approach [Acharya, Sun, Zhang'19]

Suppose data $x_i \in \{1, \ldots, k\}$, and there is a "message space" $\mathcal{Y}$

- Associate with each x some $S_x \subset \mathcal{Y}, |S_x| = s$
- Suppose $\{S_x\}_{x \in \mathcal{X}}$ is such that $\forall x \neq x', |S_x \cap S_{x'}| = \ell$
- **Mechanism:** For any $y \in \mathcal{Y}$, send message $M = y$ with probability p if $y \notin S_x$, and with probability $e^\varepsilon p$ if $y \in S_x$ (call S_x the *preferred messages for* x)

  Note: $e^\varepsilon ps + p(|\mathcal{Y}| - s) = 1$, so $p = \frac{1}{s(e^\varepsilon - 1) + |\mathcal{Y}|}$
- Server estimates f_x as

  $\tilde{f}_x = \sum_{i=1}^n (\alpha \cdot [[M_i \in S_x]] + \beta)$ ($[[P]] = 1$ iff P is True; 0 o/w)
- To have $\mathbb{E}\, \tilde{f}_x = f_x$ we just want to make sure:
  - $x_i = x \implies i$th summand has expectation 1
  - $x_i \neq x \implies i$th summand has expectation 0

A meta approach [Acharya, Sun, Zhang'19]

Suppose data $x_i \in \{1, \ldots, k\}$, and there is a "message space" $\mathcal{Y}$

- Associate with each x some $S_x \subset \mathcal{Y}, |S_x| = s$
- Suppose $\{S_x\}_{x \in \mathcal{X}}$ is such that $\forall x \neq x', |S_x \cap S_{x'}| = \ell$
- **Mechanism:** For any $y \in \mathcal{Y}$, send message $M = y$ with probability p if $y \notin S_x$, and with probability $e^\varepsilon p$ if $y \in S_x$ (call S_x the *preferred messages for* x)

  Note: $e^\varepsilon ps + p(|\mathcal{Y}| - s) = 1$, so $p = \frac{1}{s(e^\varepsilon - 1) + |\mathcal{Y}|}$
- Server estimates f_x as

  $\tilde{f}_x = \sum_{i=1}^n (\alpha \cdot [[M_i \in S_x]] + \beta)$ ($[[P]] = 1$ iff P is True; 0 o/w)
- To have $\mathbb{E}\,\tilde{f}_x = f_x$ we just want to make sure:
  - $x_i = x \implies i$th summand has expectation 1
  - $x_i \neq x \implies i$th summand has expectation 0
- In other words:
  - $\alpha e^\varepsilon ps + \beta = 1$
  - $\alpha(e^\varepsilon p\ell + p(s - \ell)) + \beta = 0$

A meta approach [Acharya, Sun, Zhang'19]

Suppose data $x_i \in \{1, \ldots, k\}$, and there is a "message space" $\mathcal{Y}$

- Associate with each x some $S_x \subset \mathcal{Y}, |S_x| = s$
- Suppose $\{S_x\}_{x \in \mathcal{X}}$ is such that $\forall x \neq x', |S_x \cap S_{x'}| = \ell$
- **Mechanism:** For any $y \in \mathcal{Y}$, send message $M = y$ with probability p if $y \notin S_x$, and with probability $e^\varepsilon p$ if $y \in S_x$ (call S_x the *preferred messages for* x)

  Note: $e^\varepsilon ps + p(|\mathcal{Y}| - s) = 1$, so $p = \frac{1}{s(e^\varepsilon - 1) + |\mathcal{Y}|}$
- Server estimates f_x as

  $\tilde{f}_x = \sum_{i=1}^n (\alpha \cdot [[M_i \in S_x]] + \beta)$ ($[[P]] = 1$ iff P is True; 0 o/w)
- To have $\mathbb{E}\, \tilde{f}_x = f_x$ we just want to make sure:
  - $x_i = x \implies i$th summand has expectation 1
  - $x_i \neq x \implies i$th summand has expectation 0
- In other words:
  - $\alpha e^\varepsilon ps + \beta = 1$
  - $\alpha(e^\varepsilon p\ell + p(s - \ell)) + \beta = 0$
- $\implies \alpha = \frac{1}{p(s-\ell)(e^\varepsilon - 1)}, \beta = -\frac{s + \ell(e^\varepsilon - 1)}{(s-\ell)(e^\varepsilon - 1)}$

Utility of meta approach

By independence,

- $Var[\tilde{f}_x] = \sum_{i=1}^{n} Var[(\alpha \cdot [[M_i \in S_x]] + \beta)$
  so $Var[\tilde{f}_x] = \alpha^2 \cdot \sum_{i=1}^{n} \mathbb{P}(M_i \in S_x)(1 - \mathbb{P}(M_i \in S_x))$

Utility of meta approach

By independence,

- $Var[\tilde{f}_x] = \sum_{i=1}^{n} Var[(\alpha \cdot [[M_i \in S_x]] + \beta)$
  so $Var[\tilde{f}_x] = \alpha^2 \cdot \sum_{i=1}^{n} \mathbb{P}(M_i \in S_x)(1 - \mathbb{P}(M_i \in S_x))$

If $x_i = x$, $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} ps$

Utility of meta approach

By independence,

- $Var[\tilde{f}_x] = \sum_{i=1}^{n} Var[(\alpha \cdot [[M_i \in S_x]] + \beta)$
  so $Var[\tilde{f}_x] = \alpha^2 \cdot \sum_{i=1}^{n} \mathbb{P}(M_i \in S_x)(1 - \mathbb{P}(M_i \in S_x))$

If $x_i = x$, $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} ps$

If $x_i \neq x$**,** $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} p\ell + p(s - \ell)$

Utility of meta approach

By independence,

- $Var[\tilde{f}_x] = \sum_{i=1}^{n} Var[(\alpha \cdot [[M_i \in S_x]] + \beta)$
  so $Var[\tilde{f}_x] = \alpha^2 \cdot \sum_{i=1}^{n} \mathbb{P}(M_i \in S_x)(1 - \mathbb{P}(M_i \in S_x))$

If $x_i = x$, $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} ps$

If $x_i \neq x$, $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} p\ell + p(s - \ell)$

Thus, $Var[\tilde{f}_x] \leq \alpha^2 \left(f_x e^{\varepsilon} ps + (n - f_x)(e^{\varepsilon} p\ell + p(s - \ell))\right)$

$$= n \cdot \frac{s + \ell(e^{\varepsilon} - 1)}{p(s - \ell)^2 (e^{\varepsilon} - 1)^2} + f_x \cdot \frac{1}{p(s - \ell)(e^{\varepsilon} - 1)}$$

$$= \frac{n(s + \ell(e^{\varepsilon} - 1))(s(e^{\varepsilon} - 1) + |\mathcal{Y}|)}{(s - \ell)^2 (e^{\varepsilon} - 1)^2} + \frac{f_x(s(e^{\varepsilon} - 1) + |\mathcal{Y}|)}{(s - \ell)(e^{\varepsilon} - 1)}$$

Utility of meta approach

By independence,

- $Var[\tilde{f}_x] = \sum_{i=1}^{n} Var[(\alpha \cdot [[M_i \in S_x]] + \beta)$
  so $Var[\tilde{f}_x] = \alpha^2 \cdot \sum_{i=1}^{n} \mathbb{P}(M_i \in S_x)(1 - \mathbb{P}(M_i \in S_x))$

If $x_i = x$, $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} ps$

If $x_i \neq x$, $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} p\ell + p(s - \ell)$

Thus, $Var[\tilde{f}_x] \leq \alpha^2 \left(f_x e^{\varepsilon} ps + (n - f_x)(e^{\varepsilon} p\ell + p(s - \ell)) \right)$

$$= n \cdot \frac{s + \ell(e^{\varepsilon} - 1)}{p(s - \ell)^2 (e^{\varepsilon} - 1)^2} + f_x \cdot \frac{1}{p(s - \ell)(e^{\varepsilon} - 1)}$$

$$= \frac{n(s + \ell(e^{\varepsilon} - 1))(s(e^{\varepsilon} - 1) + |\mathcal{Y}|)}{(s - \ell)^2 (e^{\varepsilon} - 1)^2} + \frac{f_x(s(e^{\varepsilon} - 1) + |\mathcal{Y}|)}{(s - \ell)(e^{\varepsilon} - 1)}$$

MSE is $\frac{1}{k} \mathbb{E} \|f - \tilde{f}_x\|_2^2 = \frac{1}{k} \sum_x Var[\tilde{f}_x]$, which is

$$\frac{n(1 + \frac{\ell}{s}(e^{\varepsilon} - 1))((e^{\varepsilon} - 1) + \frac{|\mathcal{Y}|}{s})}{(1 - \frac{\ell}{s})^2 (e^{\varepsilon} - 1)^2} + \frac{n((e^{\varepsilon} - 1) + \frac{|\mathcal{Y}|}{s})}{k(1 - \frac{\ell}{s})(e^{\varepsilon} - 1)}$$

Utility of meta approach

By independence,

- $Var[\tilde{f}_x] = \sum_{i=1}^{n} Var[(\alpha \cdot [[M_i \in S_x]] + \beta)$
  so $Var[\tilde{f}_x] = \alpha^2 \cdot \sum_{i=1}^{n} \mathbb{P}(M_i \in S_x)(1 - \mathbb{P}(M_i \in S_x))$

If $x_i = x$, $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} ps$

If $x_i \neq x$, $\mathbb{P}(M_i \in S_x) = e^{\varepsilon} p\ell + p(s - \ell)$

Thus, $Var[\tilde{f}_x] \leq \alpha^2 \left(f_x e^{\varepsilon} ps + (n - f_x)(e^{\varepsilon} p\ell + p(s - \ell))\right)$

$$= n \cdot \frac{s + \ell(e^{\varepsilon} - 1)}{p(s-\ell)^2(e^{\varepsilon}-1)^2} + f_x \cdot \frac{1}{p(s-\ell)(e^{\varepsilon}-1)}$$

$$= \frac{n(s + \ell(e^{\varepsilon}-1))(s(e^{\varepsilon}-1) + |\mathcal{Y}|)}{(s-\ell)^2(e^{\varepsilon}-1)^2} + \frac{f_x(s(e^{\varepsilon}-1) + |\mathcal{Y}|)}{(s-\ell)(e^{\varepsilon}-1)}$$

MSE is $\frac{1}{k}\mathbb{E}\,\|f - \tilde{f}_x\|_2^2 = \frac{1}{k}\sum_x Var[\tilde{f}_x]$, which is

$$\frac{n(1 + \frac{\ell}{s}(e^{\varepsilon}-1))((e^{\varepsilon}-1) + \frac{|\mathcal{Y}|}{s})}{(1-\frac{\ell}{s})^2(e^{\varepsilon}-1)^2} + \frac{n((e^{\varepsilon}-1) + \frac{|\mathcal{Y}|}{s})}{k(1-\frac{\ell}{s})(e^{\varepsilon}-1)}$$

Punchline: MSE increases as $\frac{\ell}{s}$, $\frac{|\mathcal{Y}|}{s}$ increase; want these small

Now reduces to a combinatorial question

Idea:

- Pick prime $q \approx e^{\varepsilon}$ and define message space $\mathcal{Y} := \mathbb{F}_q^t$
- Pick t large enough so $|\mathcal{Y}| \geq k$, and view x_i as in $\mathbb{F}_q^t$

Idea:

- Pick prime $q \approx e^{\varepsilon}$ and define message space $\mathcal{Y} := \mathbb{F}_q^t$
- Pick t large enough so $|\mathcal{Y}| \geq k$, and view x_i as in $\mathbb{F}_q^t$
- Define S_x as $(t-1)$-dimensional subspace orthogonal to x

Idea:

- Pick prime $q \approx e^{\varepsilon}$ and define message space $\mathcal{Y} := \mathbb{F}_q^t$
- Pick t large enough so $|\mathcal{Y}| \geq k$, and view x_i as in $\mathbb{F}_q^t$
- Define S_x as $(t-1)$-dimensional subspace orthogonal to x
- Then $S_x \cap S_y$ is $(t-2)$-dim subspace, so $s = q^{t-1}$, $\ell = q^{t-2}$

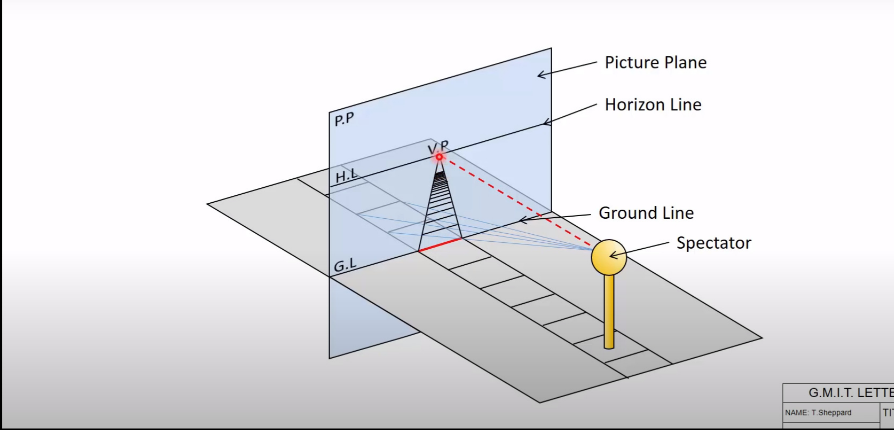
  $\frac{\ell}{s} = \frac{s}{|\mathcal{Y}|} = \frac{1}{q}$???

Idea:

- Pick prime $q \approx e^{\varepsilon}$ and define message space $\mathcal{Y} := \mathbb{F}_q^t$
- Pick t large enough so $|\mathcal{Y}| \geq k$, and view x_i as in $\mathbb{F}_q^t$
- Define S_x as $(t-1)$-dimensional subspace orthogonal to x
- Then $S_x \cap S_y$ is $(t-2)$-dim subspace, so $s = q^{t-1}$, $\ell = q^{t-2}$
  $\frac{\ell}{s} = \frac{s}{|\mathcal{Y}|} = \frac{1}{q}$???
- Not so fast: what if y is a multiple of x?
  $x = (1, 0, 0), y = (2, 0, 0)$

The fix: projective geometry

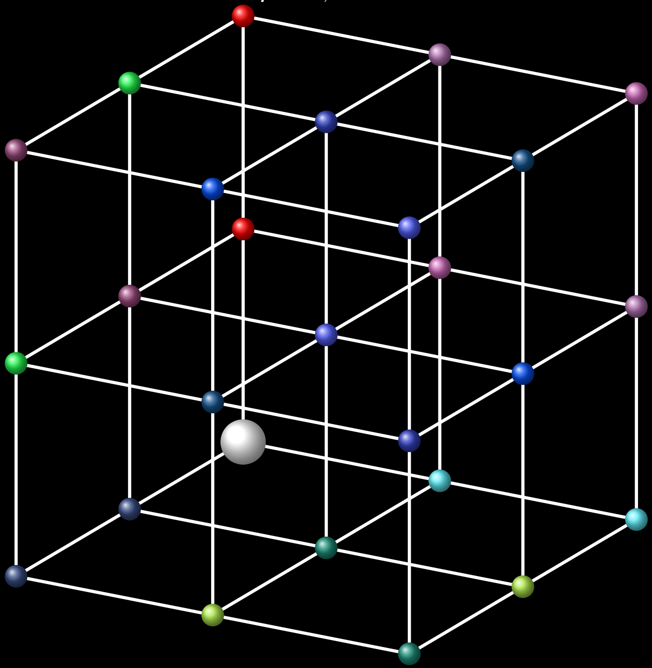
For all $x \in \mathbb{F}_q^t$, all points on line through 0 and x are equivalent.



comes from perspective drawing ("0" is spectator's eye)

(known idea in combinatorics; thanks to Noga Alon for pointing this out)

$q = 3, t = 3$



Projective geometry

Finite field projective geometry: Define *projective points* in $\mathbb{F}_q^t$ as nonzero vectors in $\mathbb{F}_q^t$ whose first nonzero is a 1 ("canonical").

Projective geometry

Finite field projective geometry: Define *projective points* in $\mathbb{F}_q^t$ as nonzero vectors in $\mathbb{F}_q^t$ whose first nonzero is a 1 ("canonical"). Can show #projective points is $\frac{q^t-1}{q-1}$; identify $[k]$ with projective points, and preferred set S_x is projective subspace "orthogonal" to x, i.e., all projective points u s.t. $\langle x, u\rangle = 0 \mod q$.

Projective geometry

Finite field projective geometry: Define *projective points* in $\mathbb{F}_q^t$ as nonzero vectors in $\mathbb{F}_q^t$ whose first nonzero is a 1 ("canonical"). Can show #projective points is $\frac{q^t-1}{q-1}$; identify $[k]$ with projective points, and preferred set S_x is projective subspace "orthogonal" to x, i.e., all projective points u s.t. $\langle x, u\rangle = 0 \mod q$.

Easy to compute s, ℓ since just amounts to counting size of a subspace of $\mathbb{F}_q^t$ of some dimension d ($d = t - 1$ or $t - 2$).

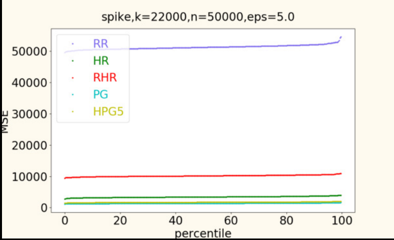
Bottom line: can get the nice $s, \ell, |\mathcal{Y}|$ we wanted!

scheme name	communication	utility loss	server time
RandomizedResponse	$\lceil \log_2 k \rceil$	$\frac{n(2e^{\varepsilon}+k)}{(e^{\varepsilon}-1)^2}$	$n+k$
RAPPOR	$O(\log k \cdot \frac{k}{e^{\varepsilon}})$	$\frac{4ne^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$n\frac{k}{e^{\varepsilon}}$
SubsetSelection	$\frac{k}{e^{\varepsilon}}(\varepsilon + O(1))$	$\frac{4ne^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$n\frac{k}{e^{\varepsilon}}$
PI-RAPPOR	$\lceil \log_2 k \rceil + O(\varepsilon)$	$\frac{4ne^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$\min(n+k^2, n\frac{k}{e^{\varepsilon}})$, or $n + ke^{2\varepsilon} \log k$ (*this work*)
HadamardResponse	$\lceil \log_2 k \rceil$	$\frac{36ne^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$n + k \log k$
RecursiveHadamardResponse	$\lceil \log_2 k \rceil$	$\frac{8ne^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$n + k \log k$
ProjectiveGeometryResponse	$\lceil \log_2 k \rceil$	$\frac{4ne^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$n + ke^{\varepsilon} \log k$
HybridProjectiveGeometryResponse	$\lceil \log_2 k \rceil$	$(1+\frac{1}{q-1})\frac{4ne^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$n + kq \log k$

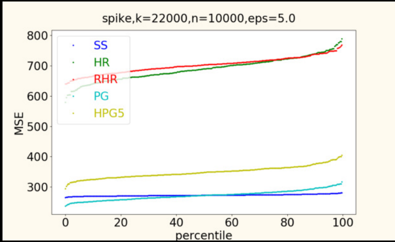
For HPG, $q \in [2, \exp(\varepsilon)+1]$ is a prime that can be chosen arbitrarily to trade off utility for runtime

PGR and HPGR are our new schemes [Feldman, Nelson, Nguyen, Talwar'22]

Experiments



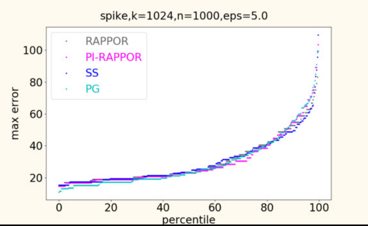
(a)



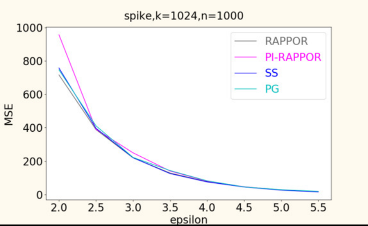
(b)

Figure: RR has significantly worse error than other algorithms, even for moderately large universes, followed by HR and RHR, which have roughly double the error of state-of-the-art algorithms. HPG trades off having slightly worse error than state-of-the-art for faster runtime.

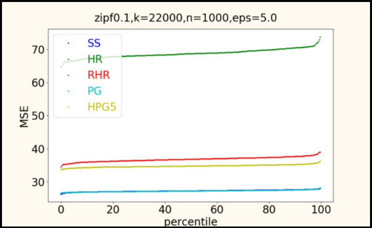
Experiments



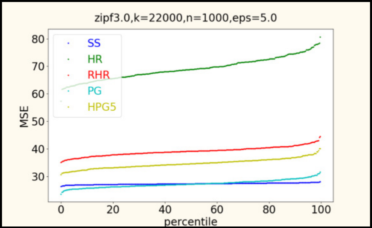
(a)



(b)



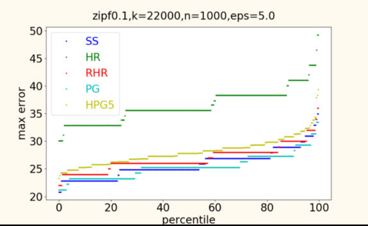
(c)



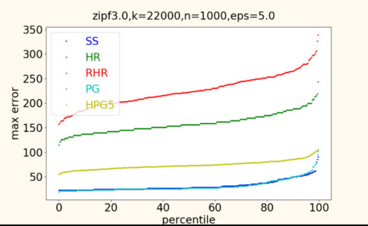
(d)

Figure: Error distributions from experiments.

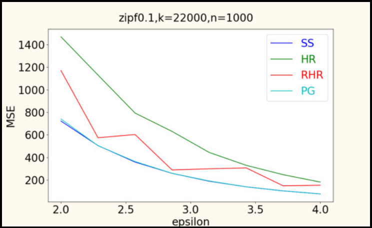
Experiments



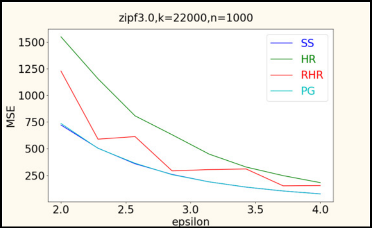
(a)



(b)



(c)



(d)

Figure: Error distributions from experiments.

Experiments

Timing:

scheme name	**runtime (in seconds)**
PI-RAPPOR	1,893.82 (approximately 31.5 minutes)
PG	36.92
HPG3	5.94
RHR	1.20
HR	0.64
RR	0.02

Table: Server runtimes for $\varepsilon = 5$, $k = 3{,}307{,}948$. For HPG, we chose the parameters $h = 50, q = 3, t = 11$, so that the mechanism rounded up the universe size to $h(q^t - 1)/(q - 1)$, which is about 34% larger than k.

Making our scheme fast

Making our scheme fast

Idea: find a recurrence relation; use dynamic programming + one more trick

Reconstruction

$$\tilde{f}_x = \sum_{i=1}^{n} (\alpha \cdot [[M_i \in S_x]] + \beta) = \alpha \cdot \left(\sum_{i=1}^{n} [[M_i \in S_x]] \right) + \beta n$$

Reconstruction

$$\tilde{f}_x = \sum_{i=1}^{n} (\alpha \cdot [[M_i \in S_x]] + \beta) = \alpha \cdot \left(\sum_{i=1}^{n} [[M_i \in S_x]] \right) + \beta n$$

Recalling the definition of S_x, this is,

$$\tilde{f}_x = \alpha \cdot \left(\sum_{\text{canonical } u : \langle x, u \rangle = 0} y_u \right) + \beta n,$$

where y_u is the number of messages M_i equal to u.

Reconstruction

$$\tilde{f}_x = \sum_{i=1}^{n} (\alpha \cdot [[M_i \in S_x]] + \beta) = \alpha \cdot \left(\sum_{i=1}^{n} [[M_i \in S_x]] \right) + \beta n$$

Recalling the definition of S_x, this is,

$$\tilde{f}_x = \alpha \cdot \left(\sum_{\text{canonical } u : \langle x, u \rangle = 0} y_u \right) + \beta n,$$

where y_u is the number of messages M_i equal to u.

Naively computing the above would take $\approx k/q$ time per x, and there are k values of x, so $\frac{k^2}{q} = \frac{k^2}{e^\varepsilon + 1}$ time total
(plus an additional n time to form the vector y)

Faster reconstruction

Can reconstruct $\tilde{f}$ faster: Dynamic programming

Faster reconstruction

Can reconstruct $\tilde{f}$ faster: Dynamic programming
For $a \in \mathbb{F}_q^j$, $b \in \mathbb{F}_q^{t-j}$, $z \in \mathbb{F}_q$, where a is further restricted to have its first nonzero entry be a 1 (it may also be the all-zeroes vector), and b is restricted to be a canonical vector when $j = 0$, define

$$F(a, b, z) = \sum_{\substack{\mathrm{pref}_j(u)=a \\ \langle \mathrm{suff}_{t-j}(u), b \rangle = z}} y_u$$

Faster reconstruction

Can reconstruct $\tilde{f}$ faster: Dynamic programming
For $a \in \mathbb{F}_q^j, b \in \mathbb{F}_q^{t-j}, z \in \mathbb{F}_q$, where a is further restricted to have its first nonzero entry be a 1 (it may also be the all-zeroes vector), and b is restricted to be a canonical vector when $j = 0$, define

$$F(a, b, z) = \sum_{\substack{\mathrm{pref}_j(u)=a \\ \langle \mathrm{suff}_{t-j}(u), b\rangle = z}} y_u$$

Then, $\tilde{f}_v = \alpha \cdot F(\perp, v, 0) + \beta n$

Faster reconstruction

Can reconstruct $\tilde{f}$ faster: Dynamic programming
For $a \in \mathbb{F}_q^j, b \in \mathbb{F}_q^{t-j}, z \in \mathbb{F}_q$, where a is further restricted to have its first nonzero entry be a 1 (it may also be the all-zeroes vector), and b is restricted to be a canonical vector when $j = 0$, define

$$F(a, b, z) = \sum_{\substack{\mathrm{pref}_j(u)=a \\ \langle \mathrm{suff}_{t-j}(u), b\rangle = z}} y_u$$

Then, $\tilde{f}_v = \alpha \cdot F(\perp, v, 0) + \beta n$

F satisfies a recurrence relation, and we can use DP

Faster reconstruction

Let $j \in [0, t)$ denote the length of the vector a. Let $\mathrm{suff}_{-1}(b)$ denote the vector b but with the first entry removed (so it is a vector of length one shorter). Then

$$F(a, b, z) = \begin{cases} y_a, & \text{if } j = t, a \neq 0, z = 0 \\ 0, & \text{if } j = t, \text{ and } a = 0 \text{ or } z \neq 0 \\ \sum_{w=0}^{1} F(a \circ w, \mathrm{suff}_{-1}(b), z - b_1 w \bmod q), & \text{if } j \neq t, a = 0 \\ \sum_{w=0}^{q-1} F(a \circ w, \mathrm{suff}_{-1}(b), z - b_1 w \bmod q), & \text{if } j \neq t, a \neq 0 \end{cases}$$

Faster reconstruction

Let $j \in [0, t)$ denote the length of the vector a. Let $\mathrm{suff}_{-1}(b)$ denote the vector b but with the first entry removed (so it is a vector of length one shorter). Then

$$F(a, b, z) = \begin{cases} y_a, & \text{if } j = t, a \neq 0, z = 0 \\ 0, & \text{if } j = t, \text{ and } a = 0 \text{ or } z \neq 0 \\ \sum_{w=0}^{1} F(a \circ w, \mathrm{suff}_{-1}(b), z - b_1 w \bmod q), & \text{if } j \neq t, a = 0 \\ \sum_{w=0}^{q-1} F(a \circ w, \mathrm{suff}_{-1}(b), z - b_1 w \bmod q), & \text{if } j \neq t, a \neq 0 \end{cases}$$

Dynamic Programming gives $O(kq^2 t)$ time and $O(kq)$ space.

Faster reconstruction

Let $j \in [0, t)$ denote the length of the vector a. Let $\text{suff}_{-1}(b)$ denote the vector b but with the first entry removed (so it is a vector of length one shorter). Then

$$F(a, b, z) = \begin{cases} y_a, & \text{if } j = t, a \neq 0, z = 0 \\ 0, & \text{if } j = t, \text{ and } a = 0 \text{ or } z \neq 0 \\ \sum_{w=0}^{1} F(a \circ w, \text{suff}_{-1}(b), z - b_1 w \bmod q), & \text{if } j \neq t, a = 0 \\ \sum_{w=0}^{q-1} F(a \circ w, \text{suff}_{-1}(b), z - b_1 w \bmod q), & \text{if } j \neq t, a \neq 0 \end{cases}$$

Dynamic Programming gives $O(kq^2t)$ time and $O(kq)$ space.

Optimization: observe $F(a, b, z) = F(a, b\zeta^{-1}, z\zeta^{-1})$ for any $\zeta \in \mathbb{F}_q^*$. If we choose ζ so that $b\zeta^{-1}$ is either canonical or the zero vector, then we cut down on the possibilities for b by a factor of q.

Faster reconstruction

Let $j \in [0, t)$ denote the length of the vector a. Let $\mathrm{suff}_{-1}(b)$ denote the vector b but with the first entry removed (so it is a vector of length one shorter). Then

$$F(a, b, z) = \begin{cases} y_a, & \text{if } j = t, a \neq 0, z = 0 \\ 0, & \text{if } j = t, \text{ and } a = 0 \text{ or } z \neq 0 \\ \sum_{w=0}^{1} F(a \circ w, \mathrm{suff}_{-1}(b), z - b_1 w \bmod q), & \text{if } j \neq t, a = 0 \\ \sum_{w=0}^{q-1} F(a \circ w, \mathrm{suff}_{-1}(b), z - b_1 w \bmod q), & \text{if } j \neq t, a \neq 0 \end{cases}$$

Dynamic Programming gives $O(kq^2 t)$ time and $O(kq)$ space.

Optimization: observe $F(a, b, z) = F(a, b\zeta^{-1}, z\zeta^{-1})$ for any $\zeta \in \mathbb{F}_q^*$. If we choose ζ so that $b\zeta^{-1}$ is either canonical or the zero vector, then we cut down on the possibilities for b by a factor of q. Leads to $O(kqt)$ time and $O(k)$ space.

Code release

https://github.com/minilek/private_frequency_oracles/

```
vector<int> ProjectiveGeometryResponse::dp_bottom_up(vector<int> &y) {
  int N = K + 1;
  for (int l = 1; l < t; ++l)
    N = max(N, ((qpows[l]-1)/(q-1) + 1) * ((qpows[t-l]-1)/(q-1) + 1) * q);
  vector<int> last(N), next(N);

  for (int a = 1; a <= K; ++a)
    last[a] = y[a-1];

  int lastA = K+1, lastB = 1, curA = 0, curB = 0;
  vector<int> ret(K);

  for (int length = t - 1; length >= 0; --length) {
    curA = (qpows[length] - 1) / (q-1) + 1, curB = (qpows[t - length] - 1) / (q-1) + 1;
    fill(next.begin(), next.end(), 0);
    for (int b = 0; b < curB; ++b) {
      vector<int> decomp = Util::decompose_canonical_vector(b, t - length, q, qpows, qinv);
      int vb0 = decomp[0], ginv = qinv[decomp[1]], vbsuff_index = decomp[2];
      for (int a = 0; a < curA; ++a) {
        if (!length) {
          int calc = last[vbsuff_index*lastA*q + 0*q + 0];
          calc += last[vbsuff_index*lastA*q + 1*q + (((int64_t)q - vb0) * ginv) % q];
          next[b] = calc;
        } else {
          int extension = a ? (2 + (a-1)*q) : 0;
          for (int z = 0; z < q; ++z) {
            int calc = 0;
            for (int d = 0; d <= (a ? q-1 : 1); ++d) {
              int new_dot_prod = ((((int64_t)q + z - vb0*d) % q) * ginv) % q;
              if (length == t-1)
                calc += (new_dot_prod ? 0 : last[extension + d]);
              else
                calc += last[vbsuff_index*lastA*q + (extension+d)*q + new_dot_prod];
            }
            next[b*curA*q + a*q + z] = calc;
          }
        }
      }
    }
    swap(last, next);
    lastA = curA;
    lastB = curB;
  }
  for (int i = 0; i < K; ++i)
    ret[i] = last[i + 1];
  return ret;
}
```

Tradeoff

Also possible to trade off utility and time: for any prime $q \in [2, \exp(\varepsilon) + 1]$, can worsen utility by $1 + 1/q$ factor but speed up runtime by $\frac{\exp(\varepsilon)+1}{q}$ factor.

Tradeoff

Also possible to trade off utility and time: for any prime $q \in [2, \exp(\varepsilon)+1]$, can worsen utility by $1 + 1/q$ factor but speed up runtime by $\frac{\exp(\varepsilon)+1}{q}$ factor.

Basic idea: Break up universe $[k]$ into h blocks of size k/h each. Each local randomizer first reveals its true block with some probability (basically RandomizedResponse) then does PGR inside the block, else just sends a totally random message.

We call this scheme HybridProjectiveGeometryResponse.

What next?

What next?

- Find a way to get around k having to be a power of $q \approx e^{\varepsilon} + 1$ (if it isn't, we round up to next power of q, which has costs)
- Finding $\tilde{f}$ so $\|f - \tilde{f}\|$ small is related to locally differentially private *heavy hitters*. Can we get sublinear-time heavy hitters algorithm with the *optimal constant* in the error $\|f - \tilde{f}\|$?