Local Differential Privacy (Part 1)

- Definition
- Frequency Oracle
  - Tianhao Wang, Jeremiah Blocki, Ninghui Li, Somesh Jha: Locally Differentially Private Protocols for Frequency Estimation. USENIX Security Symposium 2017
- Heavy Hitter Identification
  - Tianhao Wang, Ninghui Li, Somesh Jha: Locally Differentially Private Heavy Hitter Identification. IEEE TDSC (2021)
- Frequent Itemset Mining
  - Tianhao Wang, Ninghui Li, Somesh Jha: Locally Differentially Private Frequent Itemset Mining. IEEE Symposium on Security and Privacy 2018

From DP to LDP: Formal Definition

Idea of DP: Any output should be about as likely regardless of whether or not I am in the dataset

A randomized algorithm $\boldsymbol{A}$ satisfies **ε-differential privacy**, iff for any two neighboring datasets $\boldsymbol{D}$ and $\boldsymbol{D'}$ and for any output $\boldsymbol{O}$ of $\boldsymbol{A}$,

$$\Pr[\boldsymbol{A}(\boldsymbol{D}) = \boldsymbol{O}] \leq \exp(\boldsymbol{\varepsilon}) \cdot \Pr[\boldsymbol{A}(\boldsymbol{D'}) = \boldsymbol{O}]$$

A randomized algorithm $\boldsymbol{A}$ satisfies **ε-local differential privacy**, iff for any two inputs $\boldsymbol{x}$ and $\boldsymbol{x'}$ and for any output $\boldsymbol{y}$ of $\boldsymbol{A}$,

$$\Pr[\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{y}] \leq \exp(\boldsymbol{\varepsilon}) \cdot \Pr[\boldsymbol{A}(\boldsymbol{x'}) = \boldsymbol{y}]$$

Run by

ε is also called privacy budget

Smaller ε → stronger privacy

person

Idea of LDP: Any output should be about as likely regardless of my secret

Properties of (Centralized) DP

A randomized algorithm $\boldsymbol{A}$ satisfies $\boldsymbol{\varepsilon}$-differential privacy, iff for any two neighboring datasets $\boldsymbol{D}$ and $\boldsymbol{D'}$ and for any output $\boldsymbol{O}$ of $\boldsymbol{A}$,

$$\Pr[\boldsymbol{A}(\boldsymbol{D}) = \boldsymbol{O}] \leq \exp(\boldsymbol{\varepsilon}) \cdot \Pr[\boldsymbol{A}(\boldsymbol{D'}) = \boldsymbol{O}]$$

- Post-processing (of the output) is free
  - does not

What about LDP?

- Parallel composition
  - partition the dataset into subsets, each applying an ε_i-DP algorithm, the overall result satisfies $\max(\varepsilon_i)$-DP
- Sequential composition
  - apply k DP algorithms, each using ε_i, result satisfies $\sum \varepsilon_i$-DP

Properties of LDP

A randomized algorithm $\boldsymbol{A}$ satisfies $\boldsymbol{\varepsilon}$-local differential privacy, iff for any two inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$ and for any output $\boldsymbol{y}$ of $\boldsymbol{A}$,

$$\Pr[\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{y}] \leq \exp(\boldsymbol{\varepsilon}) \cdot \Pr[\boldsymbol{A}(\boldsymbol{x}') = \boldsymbol{y}]$$

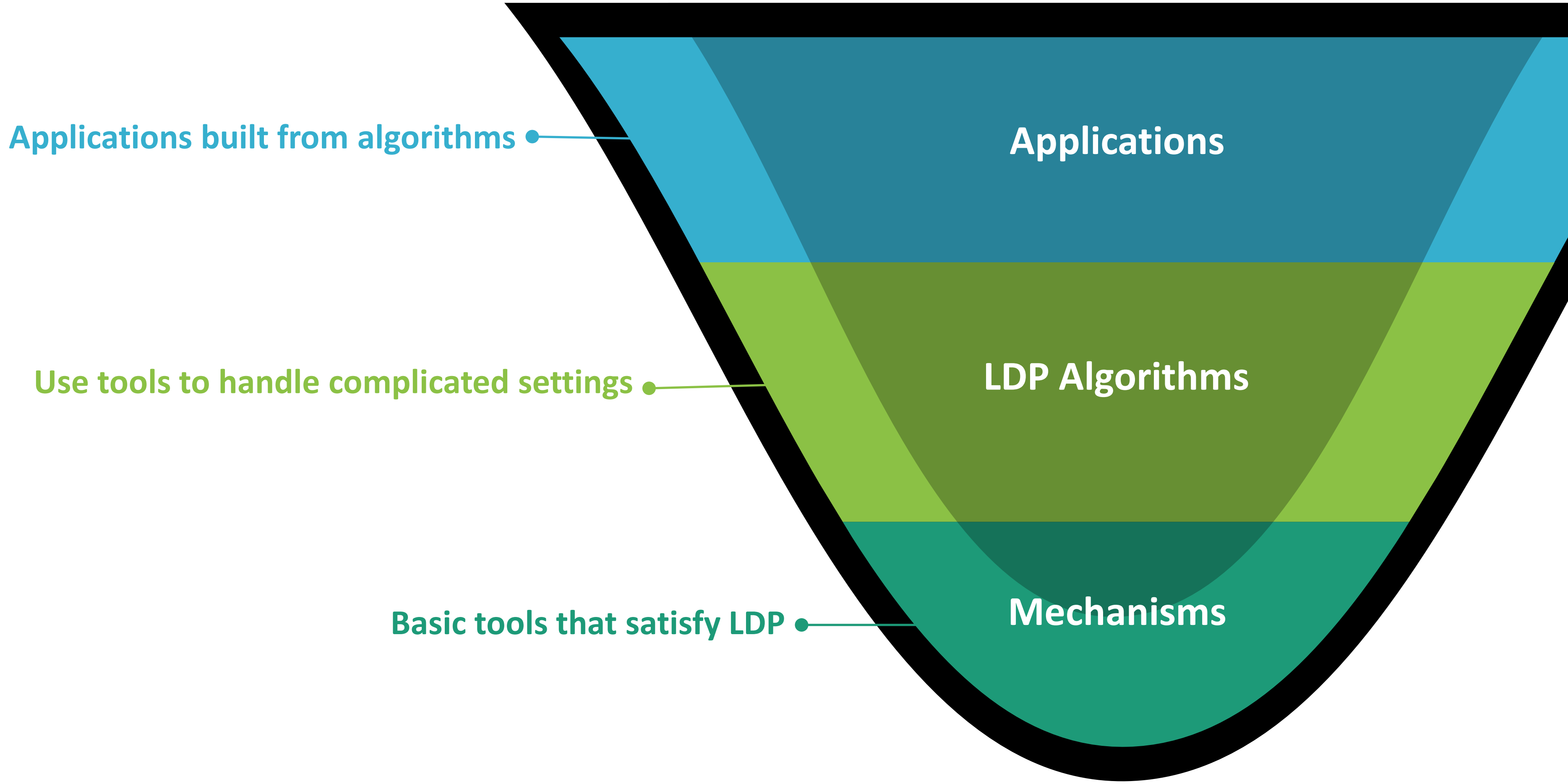
- Post-processing is also free
  - does not consume privacy budget
- No direct parallel composition
  - because each user only has one record, which cannot be partitioned
  - but one can apply different questions to different subsets of users
- Sequential composition
  - apply k LDP algorithms, each using ε_i, result satisfies $\sum \varepsilon_i$-LDP

Key difference between DP and LDP

- DP concerns two neighboring datasets
- LDP concerns any two values
- As a result, the amount of noise is different: In aggregated result for counting queries
  - Noise in DP is $\Omega(1)$ (sensitivity is constant)
  - But in LDP, even noise for each user is constant, the aggregated result is $\Omega(\sqrt{n})$ [1]
  - If the result is normalized (divide the result with n), noise is $\Omega\left(\frac{1}{n}\right)$ versus $\Omega\left(\frac{1}{\sqrt{n}}\right)$

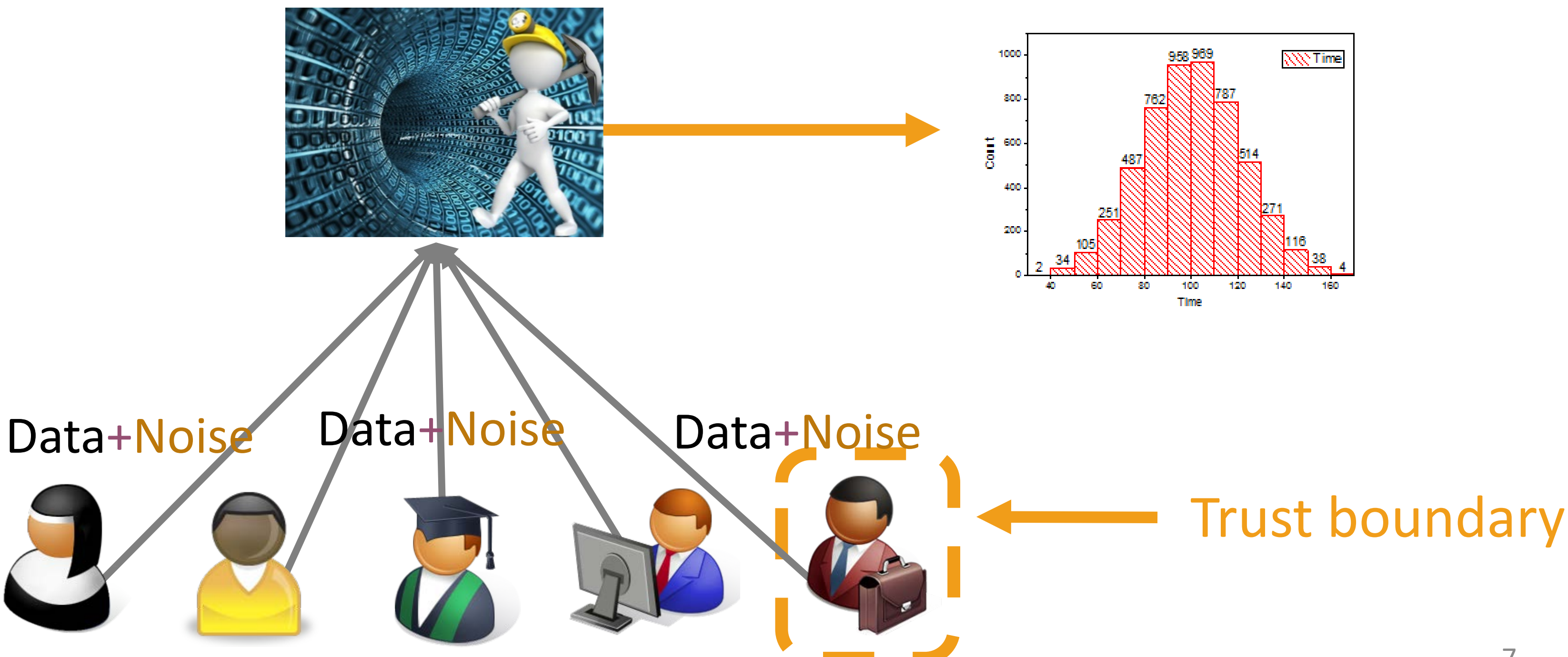
[1] Optimal lower bound for differentially private multi-party aggregation by T.-H. H. Chan, E. Shi, and D. Song

Overview


Applications built from algorithms
Use tools to handle complicated settings
Basic tools that satisfy LDP
Applications
LDP Algorithms
Mechanisms

Frequency Estimation

- Assumption: each user has a single value x from a categorical domain D
- Goal: Estimate the frequency of any value in D

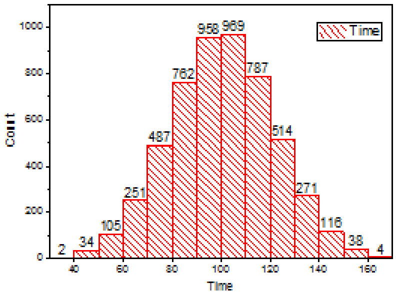


Frequency Oracle Framework

- $x \coloneqq E(v)$
takes input value v from domain D and outputs an encoded value x
- $y \coloneqq P(x)$
takes an encoded value x and outputs y.

y

- $c \coloneqq Est(\{y\})$
takes reports $\{y\}$ from all users and outputs estimations $c(v)$ for any value v in domain D



P is ε -LDP iff for any v and v' from D, and any valid output y,

$$\frac{\Pr[P(E(v))=y]}{\Pr[P(E(v'))=y]} \le e^{\varepsilon}$$

Random Response (Warner'65)

- Survey technique for private questions
- Survey peo
  - "Do you
- Each perso
  - Flip a secret coin 
  - Answer truth if head (w/p 0.5)
  - Answe
  - E.g., a ' w/p 25%
- To get unbiased estimation of the distribution:
  - If n_v out of n people have the disease, we expect to see

$$E[\,I_v] = 0.75n_v + 0.25(n - n_v) \text{ "yes" answers}$$

  - $c(n_v) = \frac{I_v - 0.25n}{0.5}$ is the unbiased estimation of number of patients

For any v and v' from "yes" and "no",

$$\frac{\Pr[P(v) = v]}{\Pr[P(v') = v]} \leq 3 = e^{\varepsilon}$$

certain about the secret.

This only handles binary attribute.
We will handle the more general setting.

Concrete Example (Let's do math)

A patient will answer "yes" w/p 75%, and "no" w/p 25%

	truth	->yes	->no
yes	80	40+20	0+20
no	20	0+5	10+5

$$c(n_v) = \frac{I_v - 0.25n}{0.5}$$

observed	65	35
estimate	80	20

(Simple) Proofs

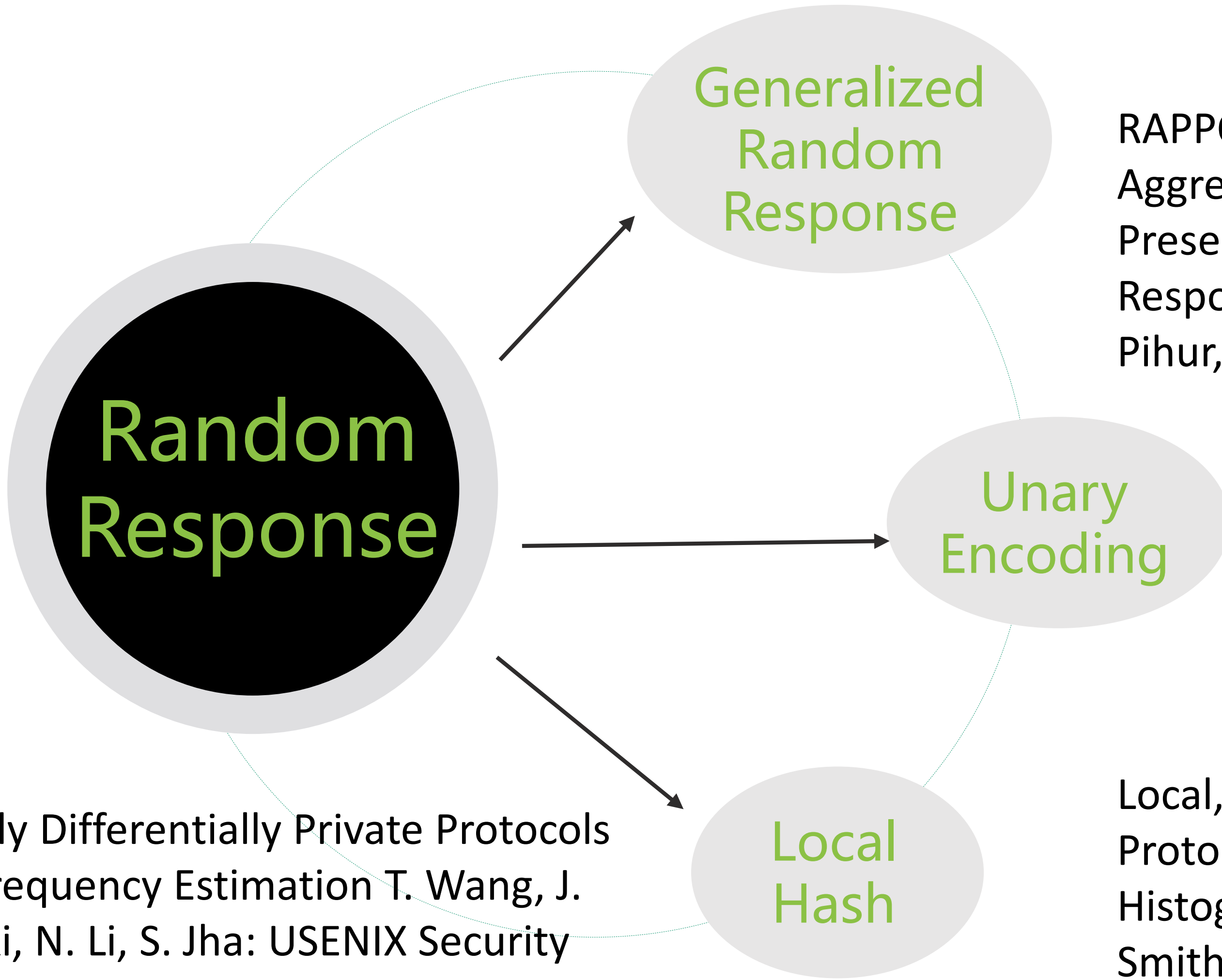
- $E[c(n_v)] = n_v$
- We have
  - $c(n_v) = \frac{I_v - 0.25n}{0.5}$
  - $E[\,I_v] = 0.75n_v + 0.25(n - n_v)$
- $E[c(n_v)] = \frac{E[I_v] - 0.25n}{0.5} = \frac{0.75n_v + 0.25(n - n_v) - 0.25n}{0.5} = n_v$
- Can be extended to other protocols
- Variance can be derived similarly

Probabilistic Analysis

Compare the result $c(v)$ with the ground truth n_v.

- $c(v)$ is a random variable
- Show that $c(v)$ is unbiased: $E[c(n_v)] = n_v$
- Compute the variance of $c(v)$: $Var[c(v)]$
- Use appropriate inequality to bound the error
  - Bernstein or Hoeffding inequalities
- Transform from variance to error bound
  - Since $c(v)$ is a binomial variable (sum of iid Bernoulli variables)

From Two to Any Categories



RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. Ú. Erlingsson, V. Pihur, A. Korolova, CCS 2014

Local, Private, Efficient Protocols for Succinct Histograms R. Bassily, A. Smith. STOC 2015.

Locally Differentially Private Protocols for Frequency Estimation T. Wang, J. Blocki, N. Li, S. Jha: USENIX Security 2017

Generalized Random Response (Direct Encoding)

- User:
  - Encode $x = v$ (suppose v from $D = \{1,2, \ldots, d\}$)
  - Toss a
  - If it is
  - Otherwise, report any other value with probability $q = \frac{1-p}{d-1}$
  - $E[I_v] = n_v \cdot p + (n - n_v) \cdot q$
  - Unbias

Intuitively, the higher p, the more accurate

However, when d is large, p becomes small (for the same ε)

ε	$p(d=2)$	$p(d=8)$	$p(d=128)$	$p(d=1024)$
0.1	0.52	0.13	0.016	0.001
1	0.73	0.27	0.027	0.002
2	0.88	0.51	0.057	0.007
4	0.98	0.88	0.307	0.05

To get rid of dependency on domain size, we move to the unary encoding protocols.

Unary Encoding (Basic RAPPOR)

- Encode the value v into a bit string $\boldsymbol{x} \coloneqq \vec{0}, \boldsymbol{x}[v] \coloneqq 1$
  - e.g., $D = \{1,2,3,4\}, v = 3,$ then $\boldsymbol{x} = [0,0,1,0]$
- Perturb each bit, preserving it with probability p
  - $p_{1\to 1} = p_{0\to 0} = p = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}$ $\quad p_{1\to 0} = p_{0\to 1} = q = \frac{1}{e^{\varepsilon/2}+1}$
  - $\Rightarrow \frac{\Pr[P(E(v))=\boldsymbol{x}]}{\Pr[P(E(v'))=\boldsymbol{x}]} \le \frac{p_{1\to 1}}{p_{0\to 1}} \times \frac{p_{0\to 0}}{p_{1\to 0}} = e^{\varepsilon}$
    - Since $\boldsymbol{x}$ is unary encoding of v, $\boldsymbol{x}$ and $\boldsymbol{x}'$ differ in two locations
- Intuition:
  - By unary encoding, each location can only be 0 or 1, effectively reducing d in each location to 2. (But privacy budget is halved.)
  - When d is large, UE is better than DE.
- To estimate frequency of each value, do it for each bit.

Truth is $[\mathbf{0}, \mathbf{3}, \mathbf{1}, \mathbf{1}]$

$c(v) = \frac{I_v - n \cdot q}{p - q}$

$\boldsymbol{c}$: $[\mathbf{0}, \frac{\mathbf{10}}{\mathbf{3}}, \frac{\mathbf{5}}{\mathbf{3}}, \mathbf{0}]$

$\boldsymbol{\Sigma y}$: $[\mathbf{1}, \mathbf{3}, \mathbf{2}, \mathbf{1}]$

Accuracy increase with number of users.

$\boldsymbol{y}$: $[\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}]$ $[\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}]$ $[\mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}]$ $[\mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}]$ $[\mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}]$

$p = \frac{4}{5}, q = \frac{1}{5}$

$\boldsymbol{x}$: $[\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}]$ $[\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}]$ $[\mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}]$ $[\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}]$ $[\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}]$

$d = 4$

$\boldsymbol{v}$: **2** **2** **3** **2** **4**

Laplacian (Gaussian)

- Instead of using randomize response for each bit, add Laplacian (Gaussian) noise to each bit.
  - Sensitivity is 2, because two vectors differ in two bits.
- It is equivalent to the centralized setting, but the number of record is only 1.
- The server aggregates the results.
- This is worse than UE.

Optimized Unary Encoding (UE)

- In UE, 1 and 0 are treated symmetrically
  - $p_{1\to 1} = p_{0\to 0} = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}$, $\quad p_{1\to 0} = p_{0\to 1} = \frac{1}{e^{\varepsilon/2}+1}$
- **Observation:** In the input, there are a lot more 0's than 1's when d is large.
- **Key Insight:** Perturb 0 and 1 differently and should reduce $p_{0\to 1}$ as much as possible
  - $p_{1\to 1} = \frac{1}{2}$, $\quad p_{1\to 0} = \frac{1}{2}$
  - $p_{0\to 0} = \frac{e^{\varepsilon}}{e^{\varepsilon}+1}$, $\quad p_{0\to 1} = \frac{1}{e^{\varepsilon}+1}$
    - $\frac{p_{1\to 1}}{p_{0\to 1}} \times \frac{p_{0\to 0}}{p_{1\to 0}} \leq e^{\epsilon}$

Binary Local Hash

Local, Private, Efficient Protocols for Succinct Histograms R. Bassily, A. Smith. STOC 2015.

- The original protocol uses a shared random matrix; this is an equivalent description
- Each user uses a random hash function H from D to $\{0,1\}$ (g=2)
- The user then perturbs the hashed bit (encode) with probabilities
  - $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1} = \frac{e^{\varepsilon}}{e^{\varepsilon}+1}, q = \frac{1}{e^{\varepsilon}+g-1} = \frac{1}{e^{\varepsilon}+1}$

$$\Rightarrow \frac{\Pr[P(E(v)) = H(v)]}{\Pr[P(E(v')) = H(v)]} = \frac{p}{q} \leq e^{\varepsilon}$$

- The user then reports the bit and the hash function
- The aggregator increments the reported group
- $E[I_v] = n_v \cdot p + (n - n_v) \cdot (\frac{1}{2}q + \frac{1}{2}p)$
- Unbiased Estimation: $c(v) = \frac{I_v - n \cdot \frac{1}{2}}{p - \frac{1}{2}}$

Example

$$\begin{array}{cccc} & +1 & & +1 \\ \end{array}$$
$$[\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}]$$

Group 1={2,4}

$v = 2$

Group 1

$\boldsymbol{D} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$

Group 0

Because of $\frac{1}{2}$, results is worse than UE

c: $[-\frac{5}{3}, \frac{5}{3}, -\frac{5}{3}, \frac{5}{3}]$

$$c(v) = \frac{I_v - n \cdot \frac{1}{2}}{p - \frac{1}{2}}$$

Σ: $[2, 3, 2, 3]$

Recover the vector: $[0, 1, 0, 1]$ $[0, 1, 0, 1]$ $[1, 0, 1, 0]$ $[0, 1, 0, 1]$ $[1, 0, 1, 0]$

y: 1 0 0 0 0

$p = \frac{4}{5}$

$x = (v + j)\%2$: 1 0 0 0 1

$d = 4$

v: 2 2 3 2 4

Optimized Local Hash (OLH)

- Observation: It is not necessary to hash into one bit.
- Conjecture: By hashing into a larger range, the result might be better.
- Technique: Optimize variance.
- Result: When $g = e^{\varepsilon} + 1$, we can achieve better accuracy.
- Intuition:
  - In original BLH, secret is **compressed** into a bit, **perturbed** and transmitted.
  - Balance between the two steps.

Comparison of Mechanisms

	DE	SHE	THE ($\theta = 1$)	SUE	OUE	BLH	OLH
Communication Cost	$O(\log d)$	$O(d)$	$O(d)$	$O(d)$	$O(d)$	$O(\log n)$	$O(\log n)$
$\mathrm{Var}[\tilde{c}(i)]/n$	$\frac{d-2+e^\varepsilon}{(e^\varepsilon-1)^2}$	$\frac{8}{\varepsilon^2}$	$\frac{2e^{\varepsilon/2}-1}{(e^{\varepsilon/2}-1)^2}$	$\frac{e^{\varepsilon/2}}{(e^{\varepsilon/2}-1)^2}$	$\frac{4e^\varepsilon}{(e^\varepsilon-1)^2}$	$\frac{(e^\varepsilon+1)^2}{(e^\varepsilon-1)^2}$	$\frac{4e^\varepsilon}{(e^\varepsilon-1)^2}$

Table 1: Comparison of Communication Cost, Computation Cost Incurred by the Aggregator, and Variances for different methods.

Direct Encoding has greater variance with larger d

OUE and OLH have the same variance

if $d < 3e^\varepsilon + 2$
    use DE
else
    if communication cost is important
        use OLH
    else
        use OUE

Two other protocols

- Subset Selection
  - S. Wang, L. Huang, P. Wang, Y. Nie, H. Xu, W. Yang, X. Li, and C. Qiao. Mutual information optimally local private discrete distribution estimation. arXiv 2016.
  - M. Ye and A. Barg. Optimal schemes for discrete distribution estimation under locally differential privacy. IEEE Transactions on Information Theory 2018.
- Hadamard Response
  - A. Jayadev, Z. Sun, and H. Zhang. Communication Efficient, Sample Optimal, Linear Time Locally Private Discrete Distribution Estimation. arXiv 2018.

Subset Selection

- Encode value v into a bit string $\boldsymbol{x} \coloneqq \vec{0}, \boldsymbol{x}[v] \coloneqq 1$
  - e.g., $D = \{1,2,3,4\}, v = 3,$ then $\boldsymbol{x}$ = [0,0,1,0]
- Instead of perturbing each bit independently, as in Unary Encoding, do the following things:
  - Randomly partition D into g subsets of equal size ($|D|$ is divided by g, $g = e^{\varepsilon} + 1$)
  - Report the subset that contains v w/p p, report any other subset w/p q
  - $\frac{p}{q} \leq e^{\varepsilon}$
- Variance is slightly better than OUE (by a constant, especially when $|D|$ is small).

Hadamard Response

- In Binary Local Hash, each user uses a random hash function H from D to $\{0,1\}$
- The original description uses a random matrix
  - Each user takes a random column
  - Each entry corresponds to one value
- In Hadamard Response, the Hadamard matrix is used (less random)
- Evaluation is asymptotically faster
- When $|D|$ is large, and one is only interested in a subset of D (as the case of heavy hitter identification), theoretical evaluation time is the same (but practically faster than evaluating hash functions).
- Not clear whether can be generalized to non-binary case

Summary of LDP Frequency Oracle Mechanisms

- Generalized Random Response
- Unary Encoding (SUE and OUE)
  - Can also be viewed as reporting random subsets
  - A variant to fix the size of reported subset
- Local Hashing Approach (BLH and OLH)
  - One way to implement BLH is to use Hadamard Response

On answering multiple questions

- Previously works (including centralized DP) suggest splitting privacy budget
- For example, when a user answers two questions, privacy budgets are $\varepsilon/2$ and $\varepsilon/2$ (assuming the two questions are of equal importance)
- In the centralized setting, there are sequential composition and parallel composition
  - By partitioning users, one uses to parallel composition
  - By split privacy budget, one uses sequential composition
  - The two can basically produce equivalent results
- What about the local setting?

On answering multiple questions

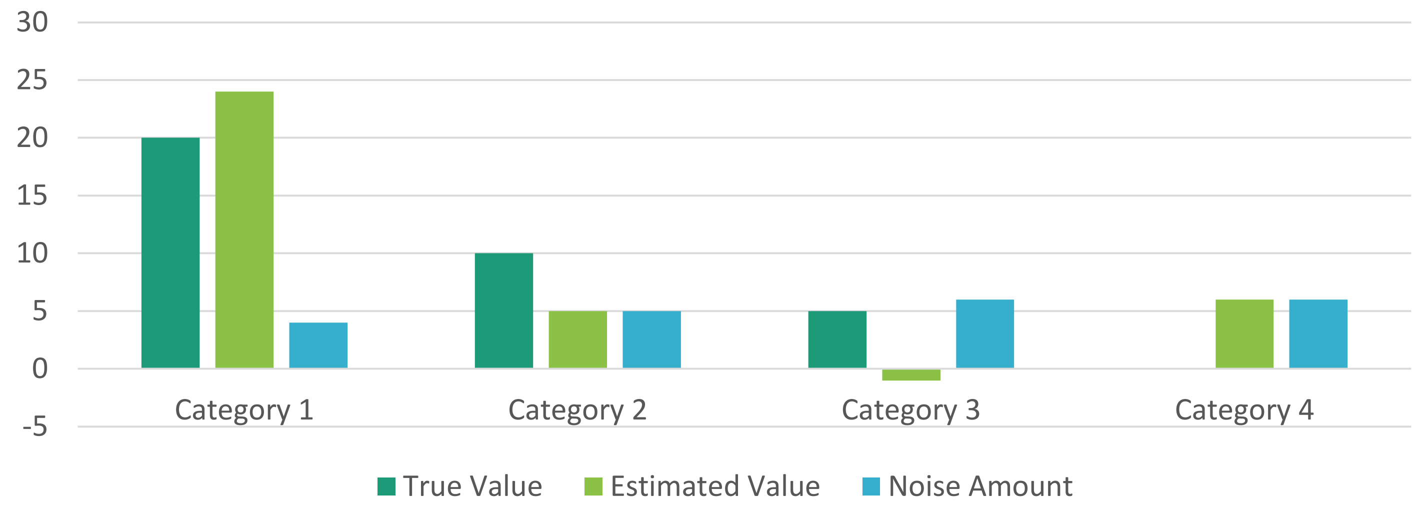
- Measure the frequency accuracy for one question
  - Assume OLH is used, for each question
    - $Var[c(v)/n] = \frac{q\cdot(1-q)}{n\cdot(p-q)^2} = \frac{4e^{\varepsilon}}{n\cdot(e^{\varepsilon}-1)^2}$
  - Assume sample variance is small
  - Normalize since two approach have different number of users
- Two settings:
  - Split privacy budget: $Var[c(v)/n] = \frac{4e^{\varepsilon/2}}{n\cdot\left(e^{\varepsilon/2}-1\right)^2}$
  - Partition users: $Var\left[c(v)/\frac{1}{2}n\right] = \frac{8e^{\varepsilon}}{n\cdot(e^{\varepsilon}-1)^2}$ ✓
- Algebra shows that it is better to partition users
- Can be generalized to $Q > 2$ questions

On answering multiple questions

- If one is interested in $K > 1$ questions
  - Partition users: $Var[c(v)/\{Q\ c\ K\}n] = \frac{4\{Q\ c\ K\}e^{\varepsilon}}{n \cdot (e^{\varepsilon}-1)^2}$
  - Split privacy budget: faster estimation algorithm
    - Appendix in Locally Differentially Private Heavy Hitter Identification. T. Wang, N. Li, S. Jha. arXiv 2017
    - CALM: Consistent Adaptive Local Marginal for Marginal Release under Local Differential Privacy. Z. Zhang, T. Wang, N. Li, S. He, J. Chen. CCS 2018
  - Variance is more complicated
    - Conjecture when $K > Q/2$, split privacy budget will be better

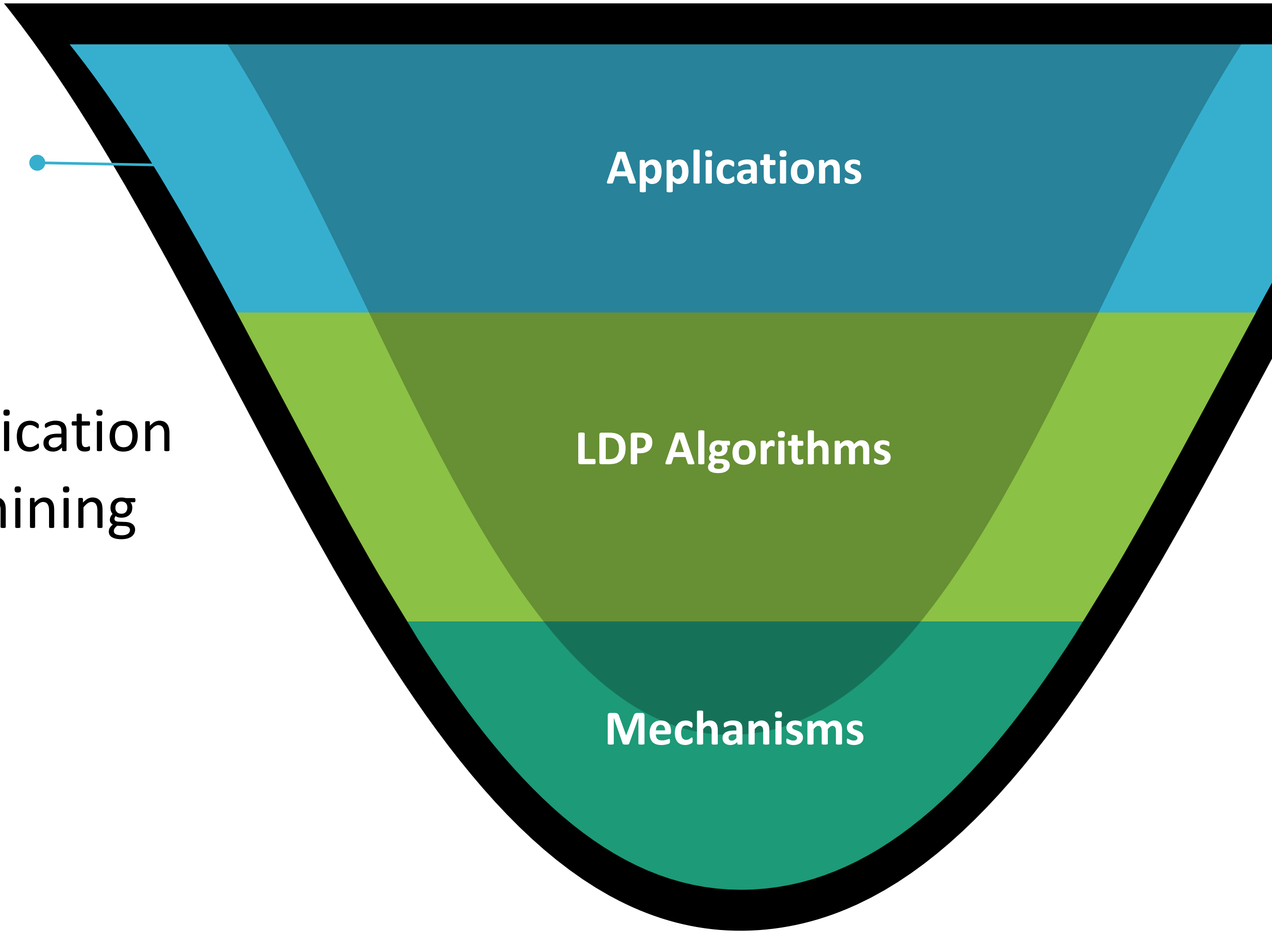
How to interpret the results

- Amount of noise is constant for each category
- If the true count is small, it may be overwhelmed by the noise, especially when domain size is big
- Estimates that are close to the quantity of noise will be replaced with 0



LDP Applications

Applications built from LDP algorithms



Focus on

- Heavy hitter identification
- Frequent itemset mining

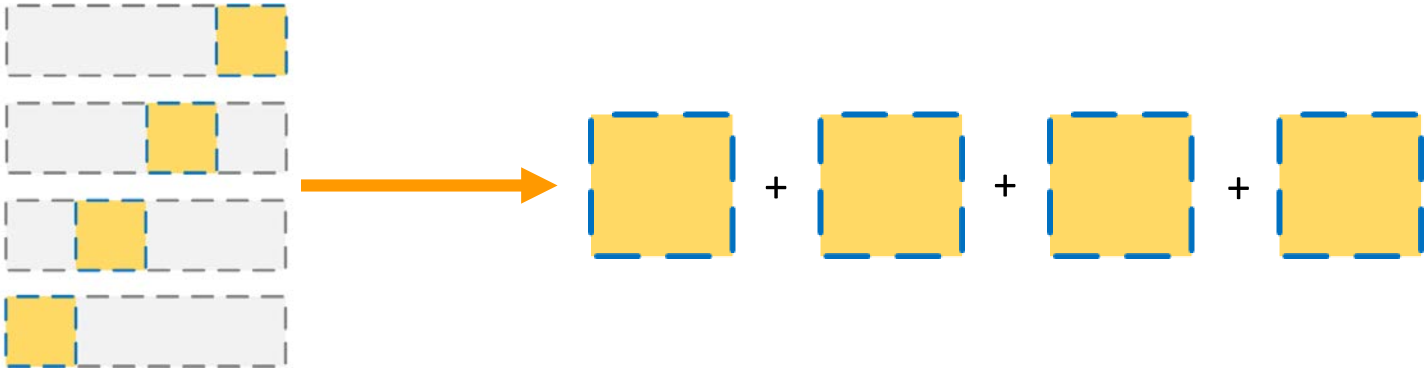
Heavy Hitter Estimation

The heavy hitter problem

- Goal: Find the k most frequent values from a large D
- Scenario (Application): Find the most popular
  - url
  - hashtag
  - new phrase
- Assumption:
  - each user has a single value x and it is represented in bits
  - D is large (when D is small, frequency oracle suffices)

A First Solution

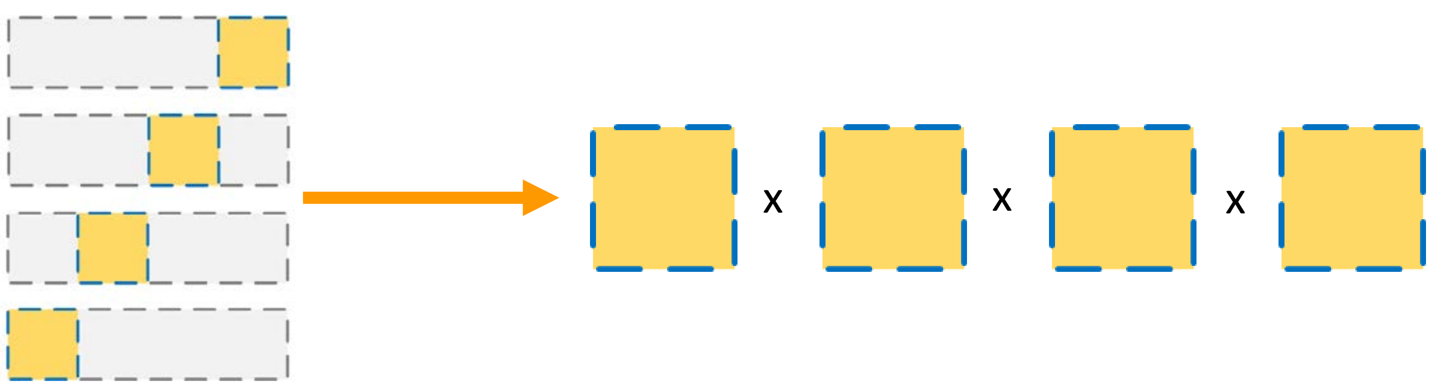
- Simpler Goal: Find one most frequent value from D
- Idea:
  - Users are partitioned into four groups
  - Each user reports one portion of its string (segment)
  - Server queries FO to one find frequent pattern in each segment
  - Concatenate the four frequent patterns



A First Solution

- Goal: Find k most frequent values from D
- Idea:
  - Server u… each segmen…
  - Calculat… sets of frequent patterns

Drawback:
Composing the four segment candidate sets gives a very large set of results.



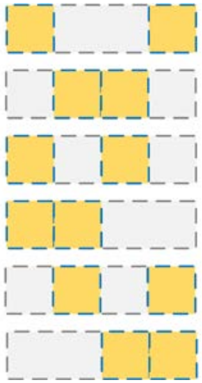
Proposals

- Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries
  - G. Fanti, V. Pihur, and U. Erlingsson, PoPETS 2016.
  - Segment Pair Method
- Local, Private, Efficient Protocols for Succinct Histograms
  - R. Bassily, A. Smith. STOC 2015.
  - Multiple Channel Method
- Prefix Extending Methods (state-of-the-art)

Segment Pair Method

- Each user reports a pair of two randomly chosen segments.
- A-priori principle:
  - A pair of segments is frequent iff both segments are frequent
  - A string is frequent iff any pair of segments is frequent
- Step 1: For e ... ent
- Step 2: For e ... pairs
- Step 3: Builc
  - each node
  - each edge represents a frequent segment pair

Drawback:
There are many possible pairs,
accuracy for each group is limited
$(Var[c(v)/n] = \frac{4e^{\varepsilon}}{n \cdot (e^{\varepsilon}-1)^2})$

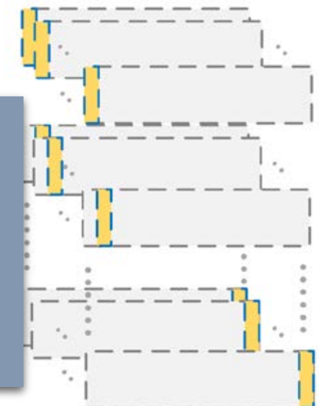


- Step 4: Find cliques in the graph (heavy hitter candidates)
- Step 5: Estimate frequencies of the heavy hitters

Multiple Channel Method

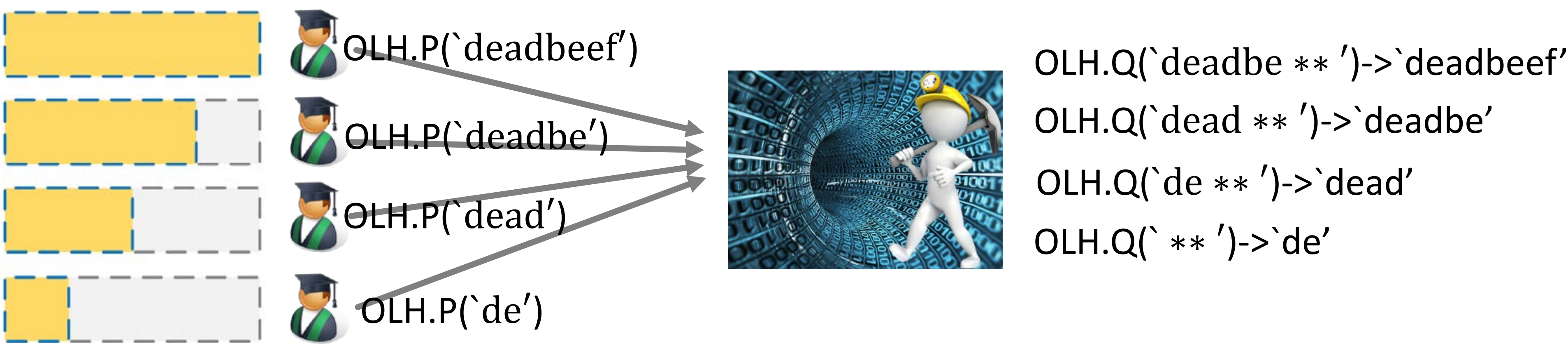
- Suppose there is only one heavy hitter, we can afford the Cartesian product, which contains only one element.
- Use multiple channels and isolate heavy hitters.
- Each u
  - In channel $H(v)$, report $v[i]$;
  - In other channels, report a uniformly random bit.
- Aggregator identifies the dominant bits in each channel
- Estimate frequencies of the heavy hitters

Drawback:
To avoid collision, many ($n^{1.5}$) channels are used.
Number of users in each channel is limited.
Computational cost is high.



Prefix Extending Method

- Start from a prefix, and gradually extend this prefix.
- Identify the frequent patterns for a small prefix first, and then extend to a larger prefix.
- Result for the last group can be used for frequency estimation



$v =$ `deadbeef'

Prefix Extending Style Proposals

- Practical locally private heavy hitters
  - R. Bassily, K. Nissim, U. Stemmer, and A. Thakurta, NIPS'17
  - TreeHist
- Locally Differentially Private Heavy Hitter Identification
  - T. Wang, N. Li, S. Jha: arXiv 2017.
  - PEM
- Privtrie: Effective frequent term discovery under local differential privacy
  - N. Wang, X. Xiao, Y. Yang, T. D. Hoang, H. Shin, J. Shin, and Y. Ge, ICDE'18
  - PrivTrie (For a different setting)

Comparison

Assume the size of domain D is 2^m; each value is encoded into m bits

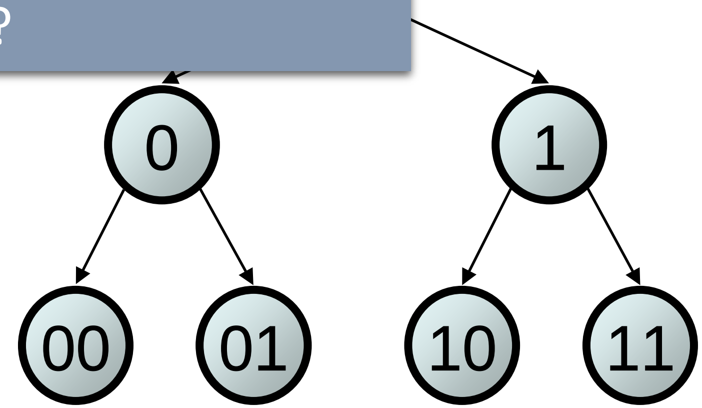
- TreeHist
  - Partition the users into m groups, each reporting one additional bit

Research Question:
How to determine number of additional bits each phase examines?

- PEM
  - Pro

  report as many bits as possible
- PrivTrie (interactive)
  - Propose to allocate less users

  on the top, more in the lower levels
  - One bit at a time



More Bits or Fewer Bits?

- Intuition:
  - More bits -> Less groups -> More users in one -> More accurate and less rounds
  - Less bits -> Less candidates -> Less likely an infrequent pattern becomes frequent
- Analyze the expected utility score.
- An optimization problem!

Optimize expected utility

- Goal: Maximize expected number of heavy hitters that can be identified
- Inpu
- Out y add
- Ass
  - A reasonable distribution (the more close the better)
  - Probabilistic approximations

Findings:

- Ideally (infeasible), all users report full string, and probe the FO for all possible string gives optimal result.
- The constraint will be the computational power.
- Each group should take as many bits as possible.

Frequent Itemset Mining

Frequent Itemset Mining

- Can be used for association rule mining etc

- Each user has a set of values

Strawman Method:
- Encode the itemset as a value in a bigger domain (of size 2^d).

Disadvantage:
- Cannot scale.
- If an item is contained in many infrequent itemsets, it will not be captured

Challenges:
1. Each user has multiple items
2. Each user's itemset size is different

{a, c, e} {b, e} {a, b, e} {a, d, e} {a, b, c, d, e, f}

- The goal is to find the frequent *singletons* and *itemsets*
- Top-3 singletons: e(5), a(4), b(3)

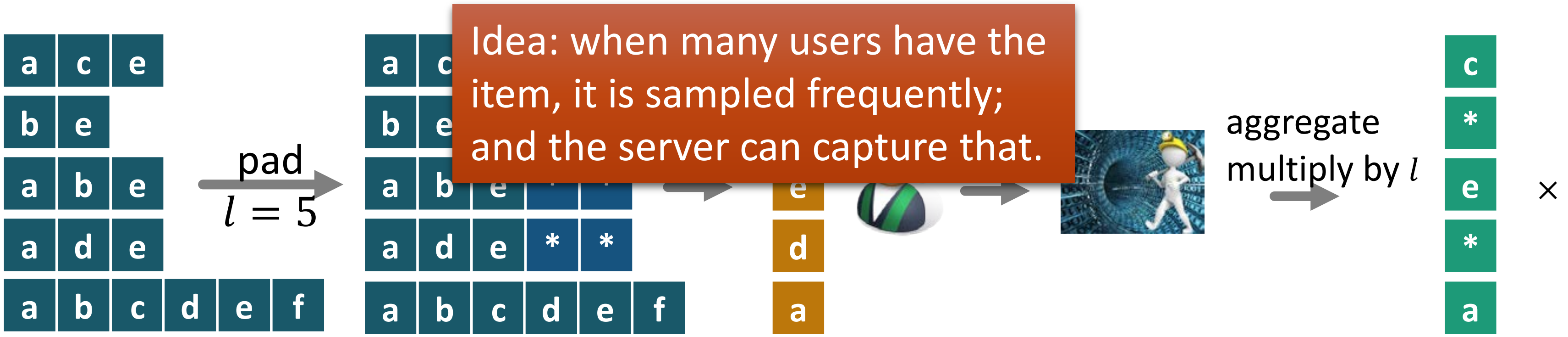
  Top-3 itemsets: {e}(5), {a}(4), {a, e}(4)

Proposals

- Heavy hitter estimation over set-valued data with local differential privacy. In CCS, 2016.
  - Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren. CCS 2016.
  - LDPMiner
- Locally Differentially Private Frequent Itemset Mining
  - T. Wang, N. Li, S. Jha: IEEE SP 2018.
  - SVIM/SVSM

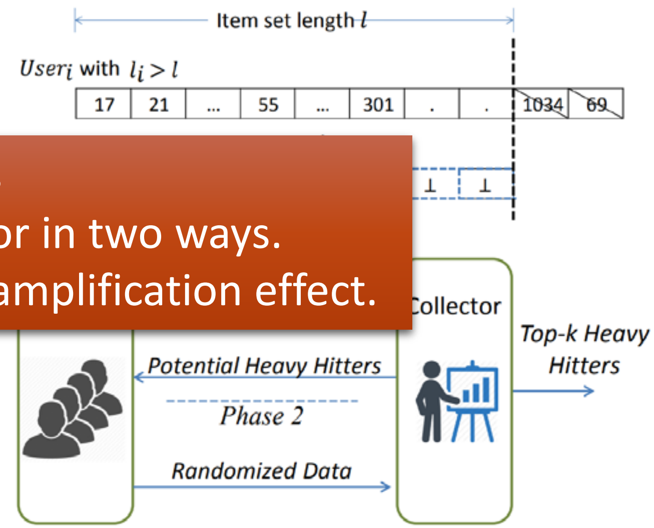
Pad and Sample Frequency Oracle

- Each user's itemset size is different
  - Pad it to a fixed length l
- Each user now has l items (or more)
  - Sample one at uniform random
  - Report via LDP (e.g., using Random Response)



LDPMiner

- Phase 1 (identify candidates)
  - Pad to l
    - l is the 90 percentile of the size distribution
  - Rand
  - Repo
  - Potential $2k$ frequent items returned
- Phase 2 (estimate frequency)
  - Intersects $\boldsymbol{v}$ with the $2k$ items
  - Pad to $2k$
    - Ensures no missed item
  - Randomly select one
  - Report

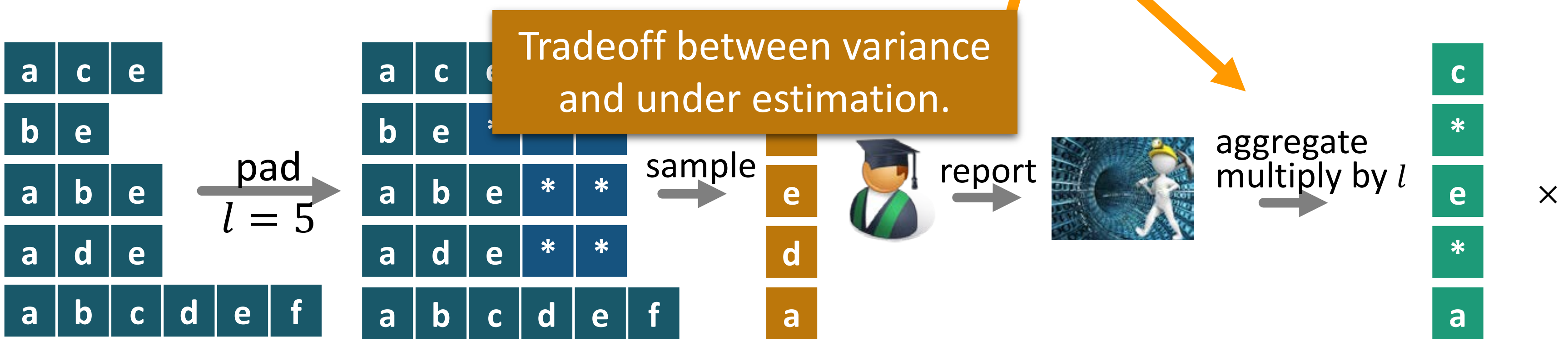


Observations
1. The value of l affects error in two ways.
2. Sampling may have a privacy amplification effect.

Could find frequent items only. Left finding frequent itemsets as an open problem.

Sources of error

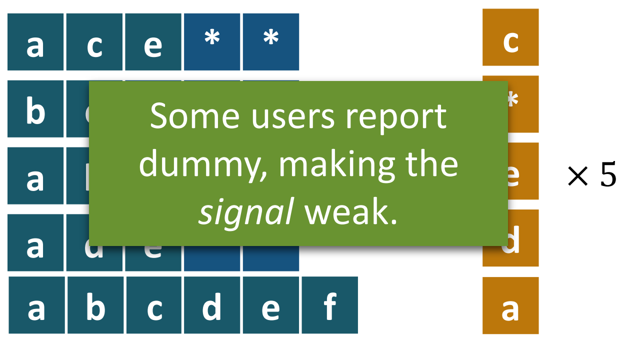
FO Variance:
To satisfy LDP, perturbation introduces noise
It increases with l, quadratically



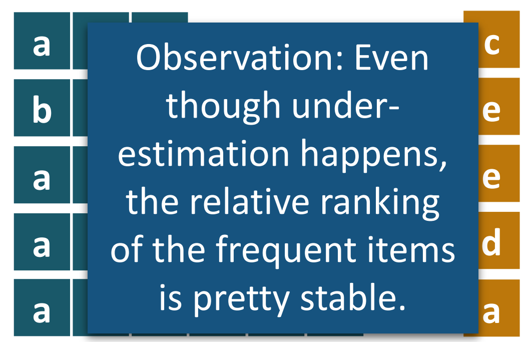
Under Estimation:
Items are selected with 1/6 but multiplied with 5
It decreases when l increases

Goal: Identification

l=5



l=1

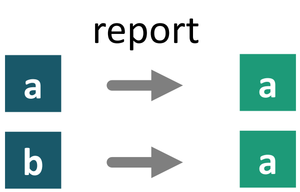


51

Privacy Amplification

- LDP bounds the perturbation.
  - E.g., in Random Response.
- With sampling, things are different.

Adaptively choose better frequency oracle based on the variance.



To satisfy ε –LDP, we can use $\varepsilon' = \ln(l(e^{\varepsilon} - 1) + 1)$

$$\frac{\Pr[P(a) = a]}{\Pr[P(b) = a]} \leq e^{\varepsilon}$$

Two randomization steps:

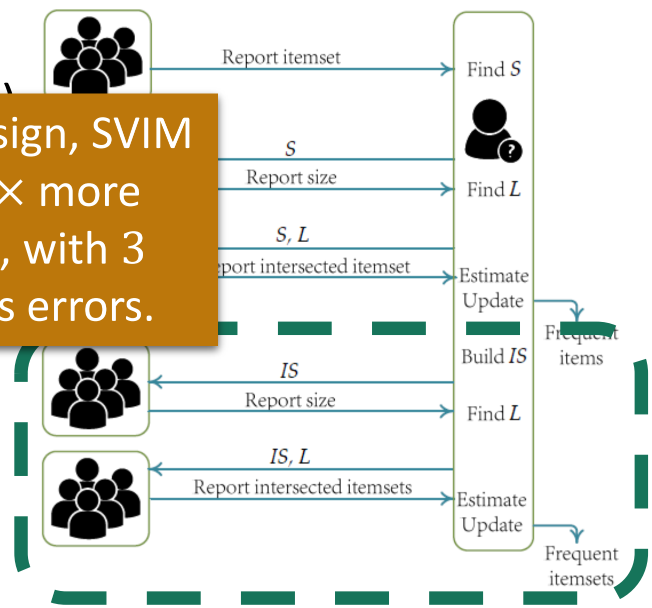
$$\begin{aligned} & \Pr[Report(a, b) = a] \\ = & \Pr[Sample(a, b) = a] \times \Pr[P(a) = a] \\ & + \Pr[Sample(a, b) = b] \times \Pr[P(b) = a] \end{aligned}$$

SVIM: Set-Value Item Mining

- Phase 1 (identify candidates)
  - Randomly select one ($l = 1$)
  - Report
  - Potential $2k$ frequent items returned
- Phase 2 (estimate len
  - Intersects $\boldsymbol{v}$ with th
    ones
  - Report the size
  - The 90-percentile l is returned
- Phase 3 (estimate frequency)
  - Pad to l
  - Randomly select one
  - Report via Adaptively chosen FO

With the new design, SVIM can identify $3\times$ more frequent items, with 3 magnitudes less errors.

SVIM (Set Value Item Mining)



Itemset Mining (more in paper)

Reporting Numerical Attributes

Numerical Mean Oracle

- $x \coloneqq E(v)$
takes input value v from domain D and outputs an encoded value x
- $y \coloneqq P(x)$
takes an encoded value x and outputs y.

$\xrightarrow{y}$

- $c \coloneqq Mean(\{y\})$
takes reports $\{y\}$ from all users and estimates mean $\frac{1}{n}\sum v$

Assume $D = [-1, +1]$

Mean estimation

Numerical Mean Oracle Proposals

- Collecting and analyzing data from smart device users with local differential privacy
  - T. T. Nguyen, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin. arXiv’16
- Collecting telemetry data privately
  - B. Ding, J. Kulkarni, and S. Yekhanin. NIPS’17

Using Existing Methods

- Apply Laplace/Gaussian noise
  - Noise is too much
- Use any Frequency Oracle
  - With the domain range partitioned into many bins
  - Transforms numerical problem to categorical problem
  - Pro: Have a better understanding of the distribution
  - Con: No optimal partition
    - Example: all values are 0.01; when there are two bins: $[-1,0), [0,+1]$, estimation will be far from truth

The Method

Discretize the problem, but using an unbiased, non-deterministic way.

- Encode the value v into a bit $x \in \{-1, +1\}$
  - $\Pr[E(v) = +1] = \frac{1}{2} + \frac{1}{2}v$, $\Pr[E(v) = -1] = \frac{1}{2} - \frac{1}{2}v$
  - This step ensures that encoding is unbiased.
- Perturb the bit, with a frequency oracle
  - Satisfy LDP
  - Provides better results

Complicated Numerical Settings

- $D \neq [-1, +1]$
  - If $D = [a, b]$
    - First convert to $[-1, +1]$; then convert the result back.
- $D = [-1, +1]^d$
  - Numerical vector setting
  - d is number of dimensions
  - Split privacy budget into each dimension
  - Report only one dimension (partition users)

Summary so far

- Random Response
- Frequency Oracles
- How to use FO
- Mean Oracle
- How to use MO

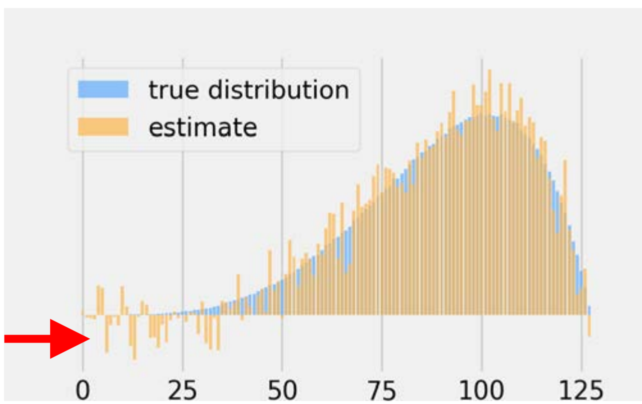
	categorical	**numerical**
Scalar	FO	Prob. Assign+RR
Vector	Split Users	Split Users

Consistency of Distribution Estimate

Consistency: $\sum_{v \in D} \hat{x}_v = 1$ and $\hat{x}_v \geq 0, \forall v \in D$

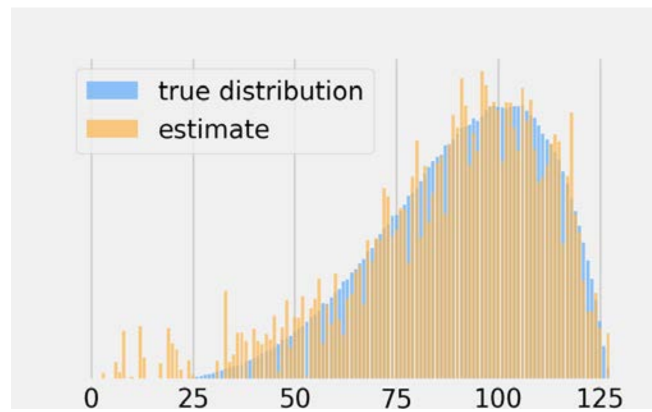
Enforce consistency: project the estimate frequencies onto simplex (L1 unit ball)

Post-processing algorithms [7]



Result of OUE

Enforce consistency



Result of post-processing
(project onto simplex)

[7]T. Wang, Z. Li, N. Li, M. Lopuhaä-Zwakenberg, and B. Skoric. Locally differentially private frequency estimation with consistency. In NDSS, 2020.

Improvement on HH

How to enforce consistency on Hierarchy Histogram(HH)?

Previous work[6] only focus on $\sum_{v \in D} \hat{x}_v = 1$ constraint, but no $\hat{x}_v \geq 0$

Our solution: **HH-ADMM**, idea from centralized DP[8].

Transform it to a constrained optimization problem

$$\text{Minimize } \tfrac{1}{2}(\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}})$$

$$\text{subject to } \mathrm{A}\hat{\boldsymbol{x}} = \boldsymbol{0}, \hat{\boldsymbol{x}} \geq \boldsymbol{0}, \hat{\boldsymbol{x}}_{\boldsymbol{0}} = 1$$

where $\hat{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}$ are all nodes in hierarchy histogram, elements in A

$$a_{ij} \begin{cases} 1, & \text{if } i = j \\ -1, & \text{node } j \text{ is a chil of node } i \\ 0, & \text{othersize} \end{cases}$$

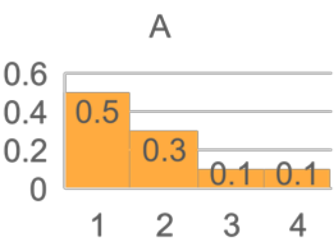
[6] T. Kulkarni, G. Cormode, and D. Srivastava. Answering range queries under local differential privacy. PVLDB, 2019

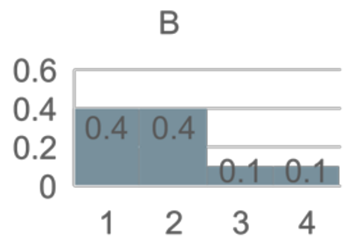
[8] J. Lee, Y. Wang, and D. Kifer. Maximum likelihood postprocessing for differential privacy under consistency constraints. SIGKDD 2015.

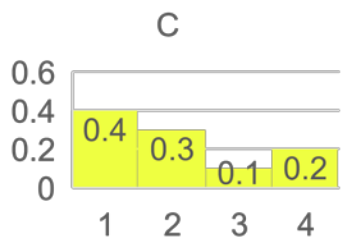
Ordered Nature of Numerical Domain

1. Values in numerical domain has distance between each other.

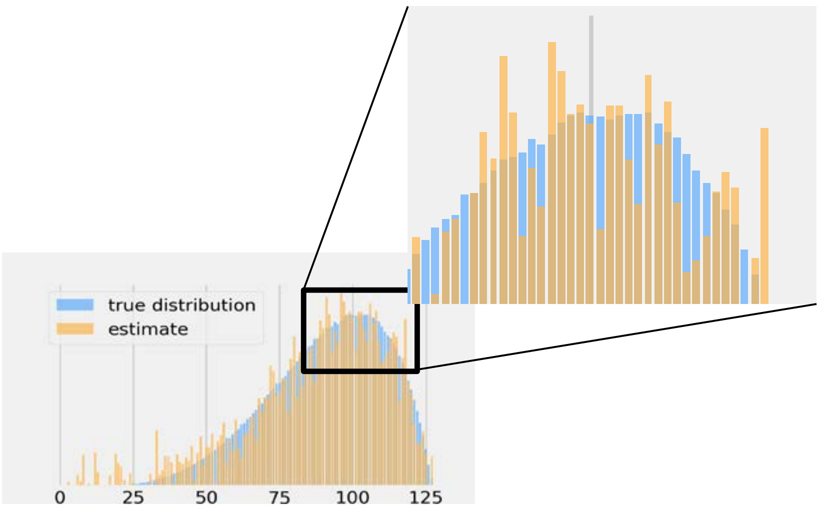
- Same L2 distance can results in very different distributions(A v.s. B and A v.s. C).
- Better metric to measure distribution distance: Wasserstein distance or KS distance.







2. Adjacent numerical values' frequencies do not vary dramatically.



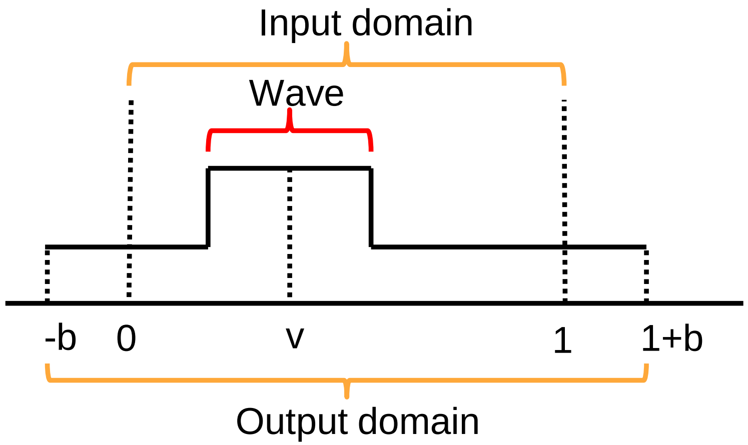
General Wave Mechanism (GW)

Intuition: in numerical domain, a report $\tilde{v}$ that is different from but close to the true value v also carries useful information about the distribution.

WLOG, assume that input domain $\mathrm{D} = [0,1]$ and output domain $\widetilde{D} = [-b, 1+b]$. Let $\mathrm{M}_v(\tilde{v}) = \Pr[\Psi(v) = \tilde{v}]$ be the probability density function of input v.

Definition (General Wave Mechanism (GW)).
There is a wave function $W: R \to [q, e^{\epsilon} q]$ with constant $q > 0$ and $\epsilon > 0$, such that the output probability density function $\mathrm{M}_v(\tilde{v}) = W(\tilde{v} - v)$:

1. $W(z) = q, \text{for } |z| > b$
2. $\int_{-b}^{b} W(z)dz = 1 - q$



Theorem 1: GW satisfies ϵ-LDP.

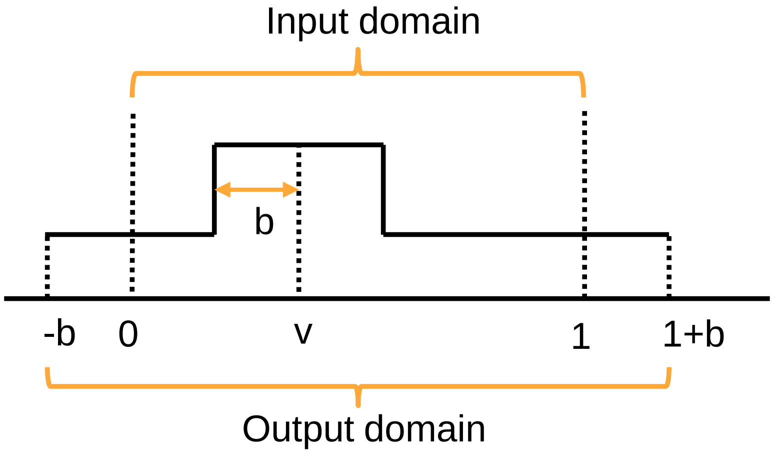
Square Wave Mechanism (SW)

How to decide the shape of wave in GW?

A special case of GW mechanism is SW Mechanism.

Definition (Square Wave Mechanism (SW)).

$$\mathrm{M}_v(\tilde{v}) = \begin{cases} p = \dfrac{e^{\epsilon}}{2be^{\epsilon}+1}, \text{if } |v-\tilde{v}| \leq b \\ q = \dfrac{1}{2be^{\epsilon}+1}, \text{otherwise} \end{cases}$$



Square Wave Mechanism

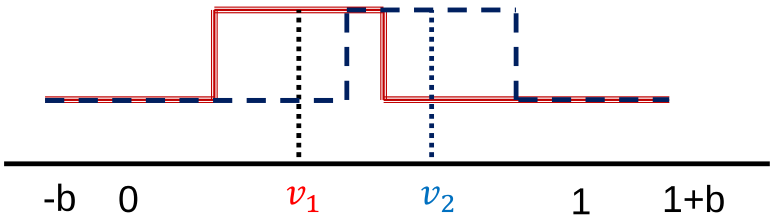
Why square wave instead of other wave shape?

Intuition: Given different values $v \neq v'$, if M_v and $\mathrm{M}_{v'}$ are identical, then there is no way to distinguish those values; the further apart M_v and $\mathrm{M}_{v'}$ are, the easier to tell them apart.

Theorem 2. For any fixed b and ϵ, the SW is the GW that maximizes the Wasserstein distance between any two output distributions of two different inputs.

Lemma 1. Given $v_1, v_2 \in \mathcal{D}$ as inputs to GW, where $v_2 > v_1$and let $\Delta = v_2 - v_1 > 0$, the Wasserstein distance between the output distributions of general wave mechanism is $\Delta(1 - (2b+1)q)$.

Lemma 2. For any fixed b and ϵ, the minimum q for GW is $\mathrm{q} = \frac{1}{2be^{\epsilon}+1}$, which is achieved if any only if the mechanism is SW.



Square Wave Mechanism

How to choose parameter b ?

- Heuristic choice: $b = \frac{\epsilon e^{\epsilon} - e^{\epsilon} + 1}{2e^{\epsilon}(e^{\epsilon} - 1 - \epsilon)}$,to maximize the upper bound of mutual information.
- When $\epsilon \to 0, b \to \frac{1}{2}; \epsilon \to \infty, b \to 0$.

Post-processing: EM

The reports $\tilde{v}$ are in $\widetilde{D} = [-b, 1+b]$.

How to map them back to $\mathrm{D} = [0,1]$?

1. Generate histogram with $\tilde{d}$ bins on $\widetilde{D}$ for the reported values.
2. Use EM algorithm to estimate the histogram with d bins on D.

Algorithm 1 Post-processing EM algorithm

Input: $\mathbf{M}, \tilde{\mathbf{v}}$

Output: $\hat{\mathbf{x}}$

while not converge **do**

E-step: $\forall i \in \{1, \ldots, d\}$,

$$P_i = \hat{\mathbf{x}}_i \sum_{j \in [\tilde{d}]} n_j \frac{\Pr\left[\tilde{v} \in \tilde{B}_j | v \in B_i, \hat{\mathbf{x}}\right]}{\Pr\left[\tilde{v} \in \tilde{B}_j | \hat{\mathbf{x}}\right]}$$
$$= \hat{\mathbf{x}}_i \sum_{j \in [\tilde{d}]} n_j \frac{\mathbf{M}_{j,i}}{\sum_{k=1}^{d} \mathbf{M}_{j,k} \hat{\mathbf{x}}_k}$$

M-step: $\forall i \in \{1, \ldots, d\}$,

$$\hat{\mathbf{x}}_i = \frac{P_i}{\sum_{k'=1}^{d} P_{k'}}$$

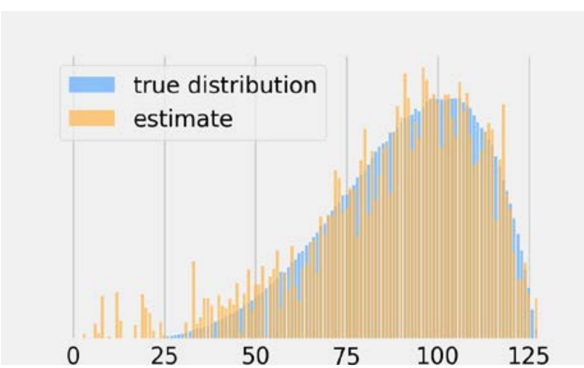
end while

Return $\hat{\mathbf{x}}$

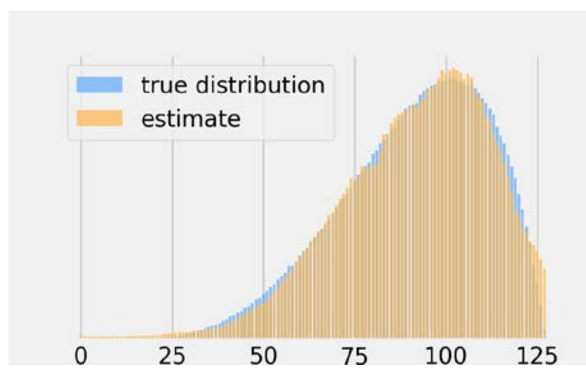
Post-processing: EM with smoothing (EMS)

How to use the prior knowledge that adjacent numerical values' frequencies do not vary dramatically?

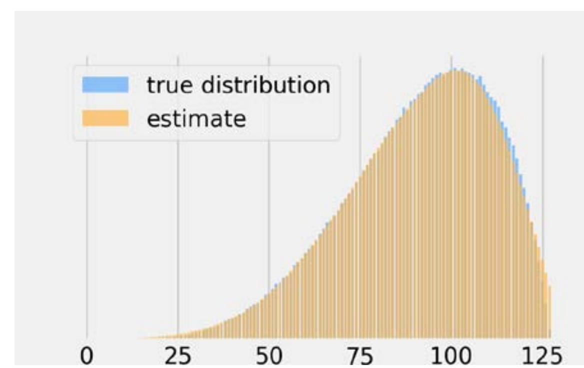
Smoothing after every M-step: $\hat{x_i} = \frac{1}{2}\hat{x_i} + \frac{1}{4}(\hat{x}_{i-1} + \hat{x}_{i+1})$



Result of Norm sub



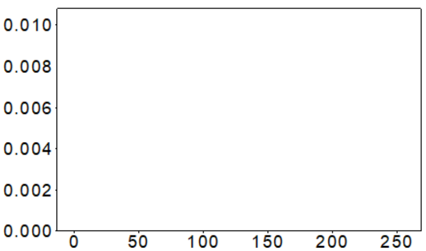
Result of SW+EM



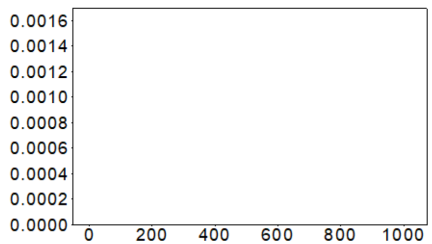
Result of SW+EMS

Experiments

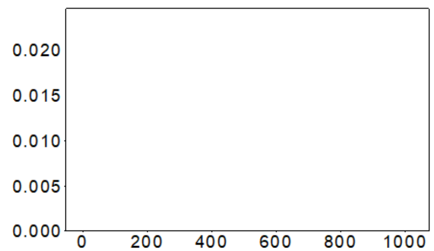
Four datasets:



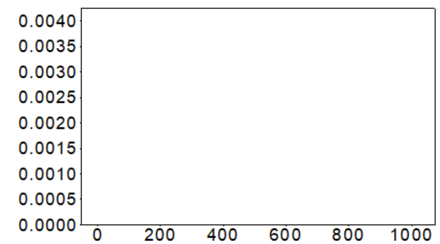
Beta(5,2)



Taxi pickup time



Income



Retirement

Metrics:

1. Wasserstein distance and Kolmogorov-Smirnov (KS) distance
2. Range queries
3. mean/variance/quantiles

Experiments

Wasserstein distance (a.k.a earth mover distance) : Given a frequency vector $\boldsymbol{x}$, the cumulative function $P(\boldsymbol{x}, v) = \sum_{i=1}^{v} x_v$, one dimension Wasserstein distance :

$$W_1(\boldsymbol{x}, \widehat{\boldsymbol{x}}) = \sum_{v \in D} |P(\boldsymbol{x}, v) - P(\widehat{\boldsymbol{x}}, v)|$$

Beta(5,2) | Taxi pickup time | Income | Retirement