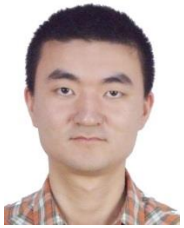
Differential Privacy in the Local Setting

Ninghui Li (Purdue University)

Joint work with Tianhao Wang, Somesh Jha, Jeremiah Blocki



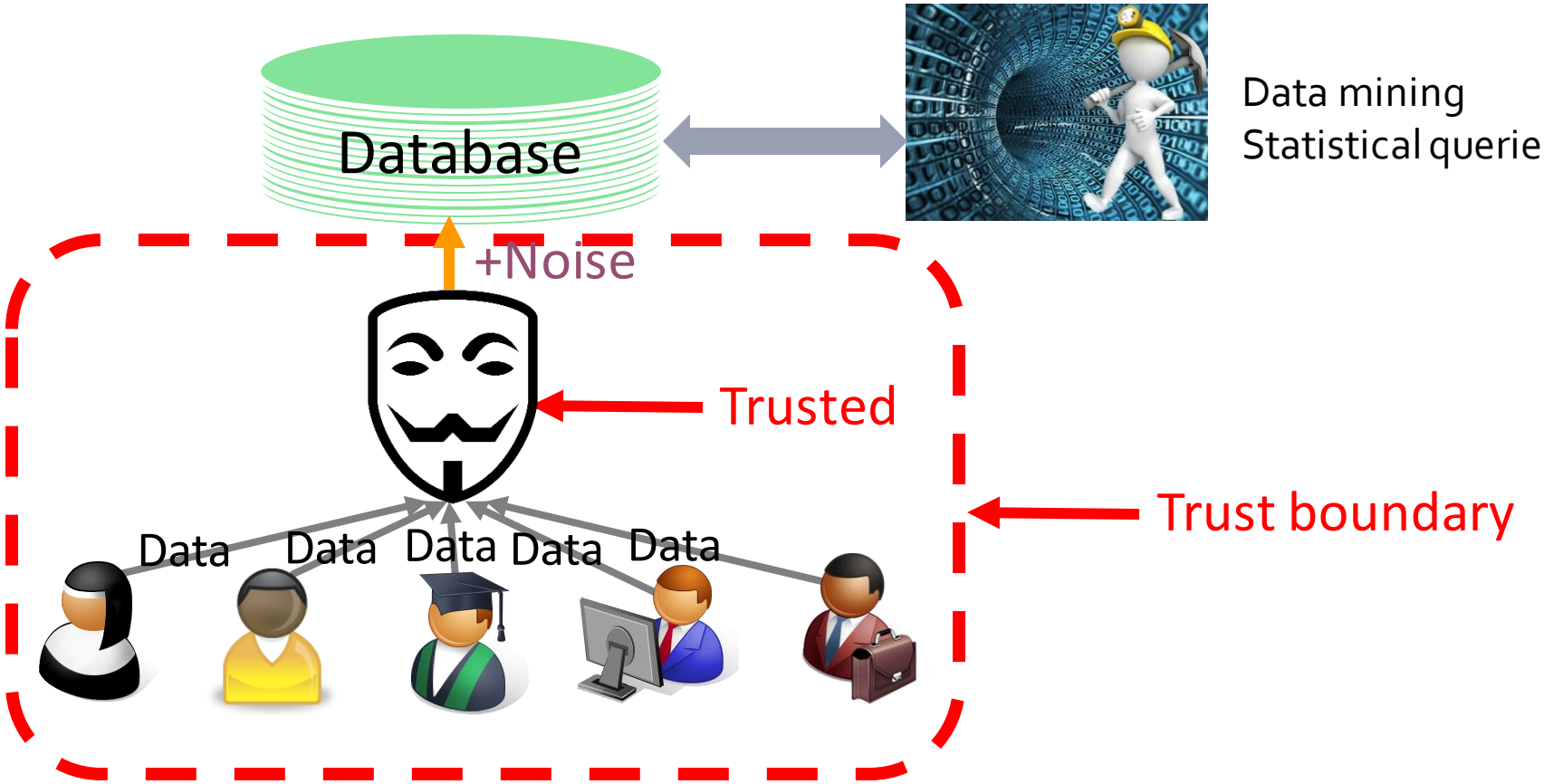
Differential Privacy

Data mining
Statistical queries

Classical/
centralized
setting

Differential Privacy Interpretation:
The decision to include/exclude an individual's record has limited (ε) influence on the outcome.
Smaller ε ➔ Stronger Privacy

Differential Privacy



Local Differential Privacy

As Apple starts analyzing web browsing & health data, how comfortable are you with differential privacy?

Ben Lovejoy - Jul. 7th 2017 6:59 am PT @benlovejoy

Outline

- Motivation (just covered)
- Histogram Estimation
- Frequent Itemset Mining
- Other problems

Outline

- Motivation
- Histogram Estimation
- Frequent Itemset Mining
- Other problems

Abstract LDP Protocol for Frequency Estimation

- $x \coloneqq E(v)$

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.

P satisfies ε-LDP

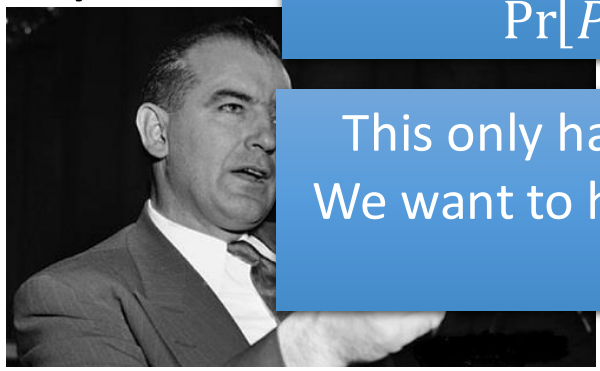
y

- $c \coloneqq Est(\{y\})$

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

Frequency estimation

The Warner Model (1965)



We say a prot
for any v and i
$\frac{\Pr[P(}{\Pr[P(}$

This only han
We want to ha

- Survey technique for private questions
- Survey people:
  - "Are you communist party?"
- Each person:
  - Flip a secret coin
  - Answer truth if head (w/p 0.5)
  - Answer randomly if tail
  - E.g., a communist will answer "yes" w/p 75%, and "no" w/p 25%
- To get unbiased estimation of the distribution:
  - If n_v out of n people are communist, we expect to see

    $E[I_v] = 0.75n_v + 0.25(n - n_v)$ "yes" answers
  - $c(n_v) = \frac{I_v - 0.25n}{0.5}$ is the unbiased estimation of number of communists
  - Since $E[c(n_v)] = \frac{E[I_v] - 0.25n}{0.5} = n_v$

Provide deniability:
Seeing answer, not certain about the secret.

Frequency Estimation Protocols

- Randomised response: a survey technique for eliminating evasive answer bias
  - S.L. Warner, Journal of Ame. Stat. Ass. 1965
  - Direct Encoding (Generalized Random Response)
- RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response.
  - Ú. Erlingsson, V. Pihur, A. Korolova, CCS 2014
  - Unary Encoding, Encode into a bit-vector
- Local, Private, Efficient Protocols for Succinct Histograms
  - R. Bassily, A. Smith. STOC 2015.
  - Binary Local Hash: Encode by hashing and then perturb
- Locally Differentially Private Protocols for Frequency Estimation
  - T. Wang, J. Blocki, N. Li, S. Jha: USENIX Security 2017

Direct Encoding (Random Response)

Intuitively, the hig

However, when d i

- User:
  - Encode $x = v$ (suppose v from $D = \{1,2, \dots, d\}$)
  - Toss a coin with bias p
  - If it is head, report the true value $y = x$
  - Otherwise, report any other value with probability $q = \frac{1-p}{d-1}$ (uniformly at random)
    - $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+d-1}, q = \frac{1}{e^{\varepsilon}+d-1} \Rightarrow \frac{\Pr[P(\boldsymbol{v})=\boldsymbol{v}]}{\Pr[P(\boldsymbol{v}\prime)=\boldsymbol{v}]} = \frac{p}{q} = e^{\varepsilon}$
- Aggregator:
  - Suppose n_v users possess value v, I_v is the number of reports on v.
  - $E[I_v] = n_v \cdot p + (n - n_v) \cdot q$
  - Unbiased Estimation: $c(v) = \frac{I_v - n \cdot q}{p-q}$

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
  - $p_{1\to1} = p_{0\to0} = p = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}$ $\qquad p_{1\to0} = p_{0\to1} = q = \frac{1}{e^{\varepsilon/2}+1}$
  - $\Rightarrow \frac{\Pr[P(E(v))=\boldsymbol{x}]}{\Pr[P(E(v'))=\boldsymbol{x}]} \leq \frac{p_{1\to1}}{p_{0\to1}} \times \frac{p_{0\to0}}{p_{1\to0}} = e^{\varepsilon}$
    - 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.
  - When d is large, UE is better than DE.
- To estimate frequency of each value, do it for each bit.

Binary Local Hash

- The protocol description in [Bassily-Smith '15] is complicated
- This is an equivalent description
- Each user uses a random hash function from D to $\{0,1\}$
- The user then perturbs the bit with probabilities
  - $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1} = \frac{e^{\varepsilon}}{e^{\varepsilon}+1}, q = \frac{1}{e^{\varepsilon}+g-1} = \frac{1}{e^{\varepsilon}+1}$

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

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

Our Work

- We measure utility of a mechanism by its variance
  - E.g., in Random Response,
    - $Var[c(v)] = Var\left[\frac{I_v - n \cdot q}{p-q}\right] = \frac{Var[I_v]}{(p-q)^2} \approx \frac{n \cdot q \cdot (1-q)}{(p-q)^2}$
- We p… cast existi…
  - Ea… v
    - …ts each value with a corresponding 1
    - E.g., In BLH, Support(y) = $\{v \mid H(v) = y\}$
  - A pure protocol is specified by p' and q'
    - Each input is perturbed into a value "supporting it" with p', and into a value not supporting it with q'

$min_{q'} Var[c(v)]$

or $min_{q'} \frac{n \cdot q' \cdot (1-q')}{(p'-q)'^2}$

where p', q' satisfy ε-LDP

Optimized Unary Encoding (UE)

- In the original UE, 1 and 0 are treated symmetrically
  - $p_{1\to1} = p_{0\to0} = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}, \quad p_{1\to0} = p_{0\to1} = \frac{1}{e^{\varepsilon/2}+1}$
- **Observation:** In the input, there are a lot more 0's than 1's when d is large.
- **Key Insight:** We can perturb 0 and 1 differently and should reduce $p_{0\to1}$ as much as possible
  - $p_{1\to1} = \frac{1}{2}, \quad p_{1\to0} = \frac{1}{2}$
  - $p_{0\to0} = \frac{e^{\varepsilon}}{e^{\varepsilon}+1}, \quad p_{0\to1} = \frac{1}{e^{\varepsilon}+1}$
    - $\frac{p_{1\to1}}{p_{0\to1}} \times \frac{p_{0\to0}}{p_{1\to0}} \leq e^{\epsilon}$

Optimized Local Hash (OLH)

- In original BLH, secret is **compressed** into a bit, **perturbed** and transmitted.
- Both steps cause information loss:
  - Compressing: loses much
  - Perturbation: information loss depends on ϵ
- **Key Insight**: We want to make a balance between the two steps:
  - By compressing into more groups, the first step carries more information
- Variance is optimized when $g = e^{\varepsilon} + 1$
- See our paper for details.

Comparison of Different 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.

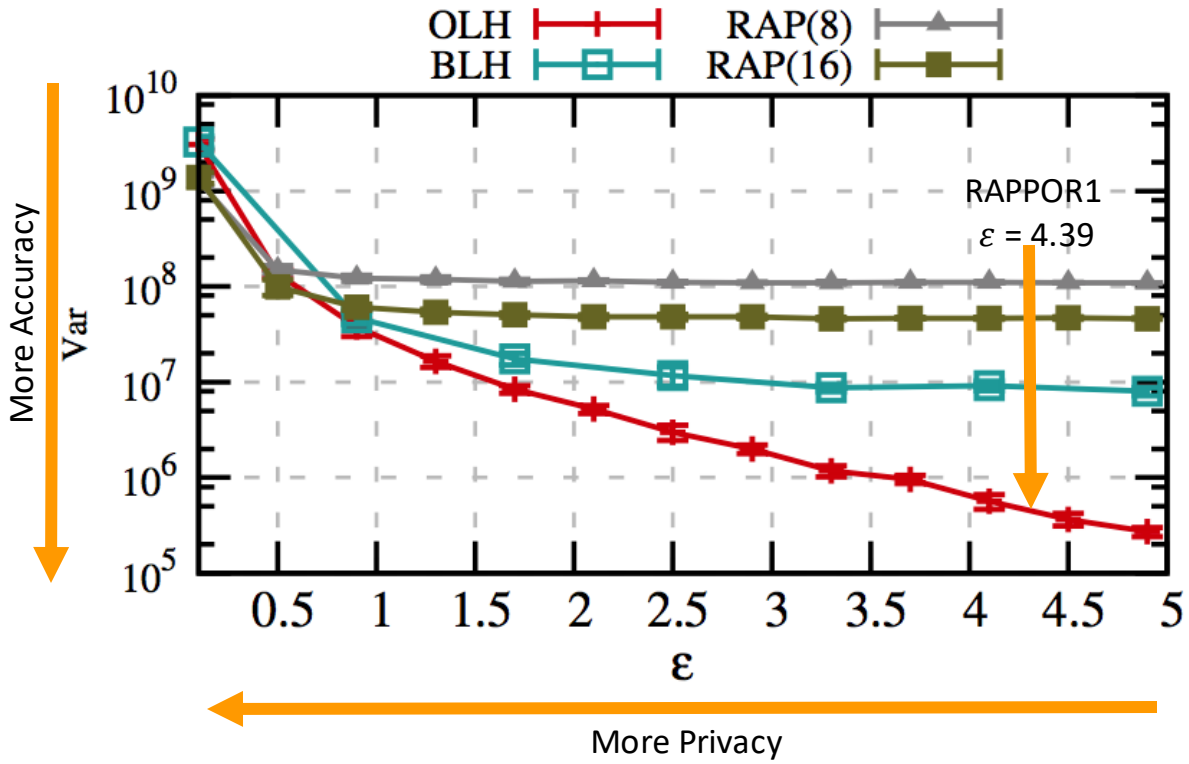
Direct Encoding has greater variance with larger d

OUE and OLH have the same variance
But OLH has smaller communication cost

Experiments Highlights

- Dataset: Kosarak dataset
  - (also on Rockyou dataset and a Synthetic dataset)
- Competitors: RAPPOR, BLH, OLH
  - Randomized Response is not compared because the domain is large
- Key Results:
  - OLH performs better, with variance orders of magnitudes smaller, especially when ε is large
  - This also confirms our analytical conclusion

Accuracy on Frequent Values



Outline

- Motivation (already covered)
- Histogram Estimation (just covered)
- Frequent Itemset Mining
- Other problems

Outline

- Motivation
- Histogram Estimation
- Frequent Itemset Mining
- Other problems

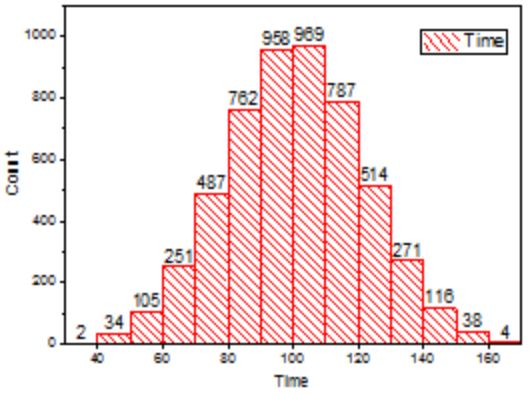
The Frequency Oracle

*For ease of presentation, let's forget the encoding function and assume it is contained in the perturbation

- $y \coloneqq P(v)$ takes input value v from domain D 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



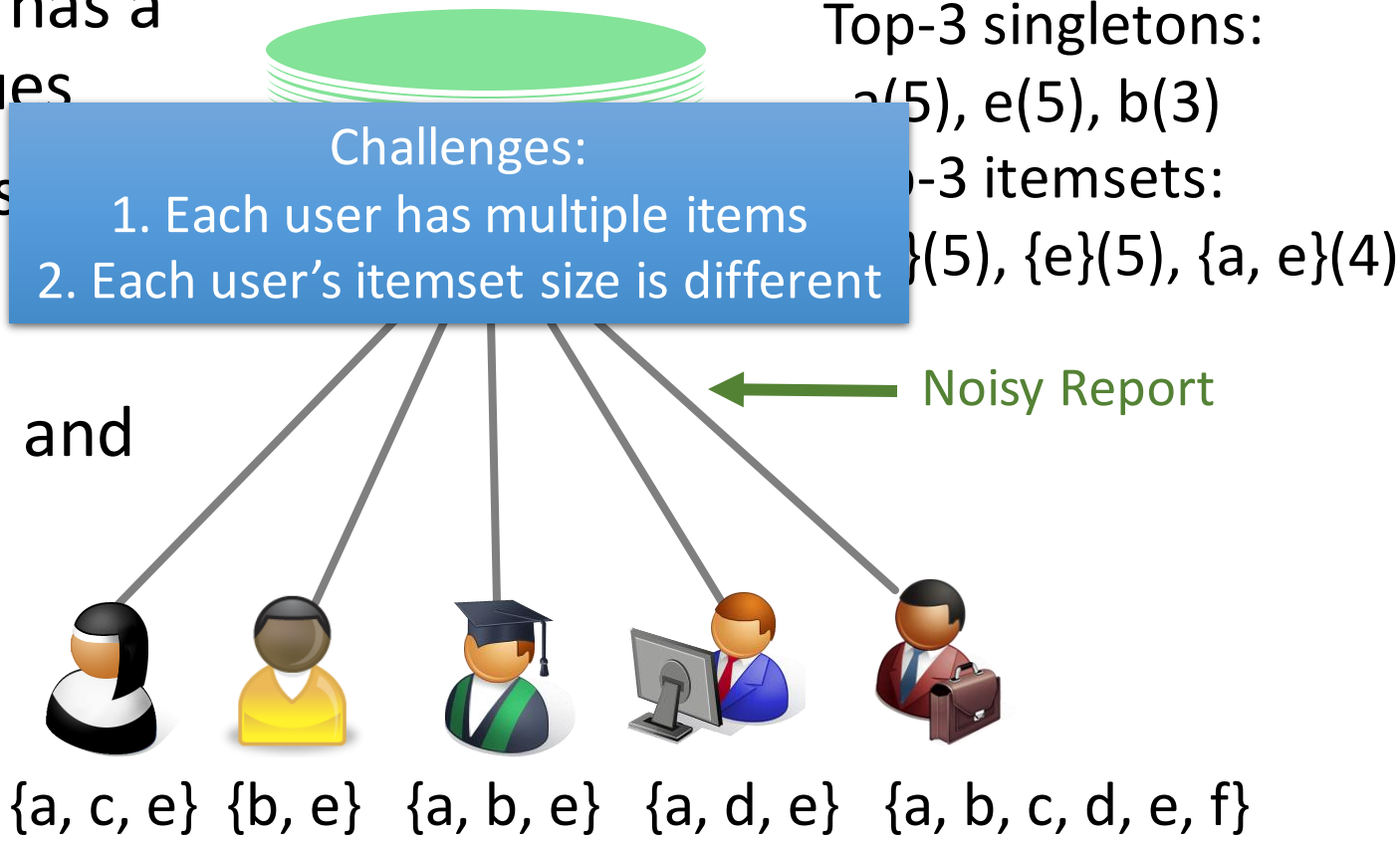
FO is ε -LDP iff for any v and v' from D, and any valid output y,

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

Frequent Itemset Mining

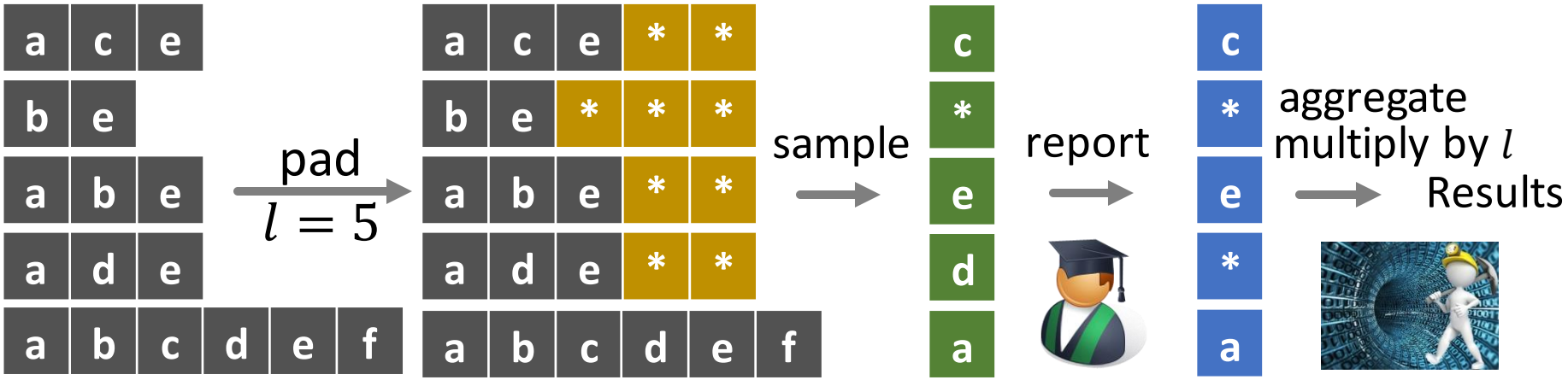
- Each user has a set of values
- The goal is to find the frequent *singletons* and *itemsets*

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



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 a Frequency Oracle (e.g., Random Response)

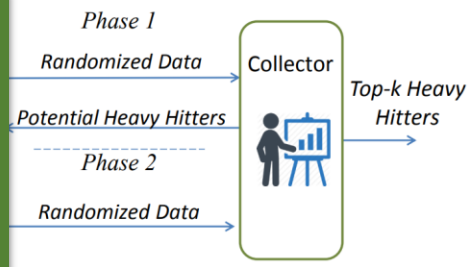


LDPMiner ["Heavy hitter estimation over set-valued data with local differential privacy"Qin et al', CCS'16]

- Ph
  - distribution
  - Randomly selec
  - Report via Freq
- Phase 2 (estimat
  - Intersects $\boldsymbol{v}$ wit k identified ones
  - Pad to k
    - Ensures no missed item
  - Randomly select one
  - Report via Frequency Oracle

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

Our contribution
PSFO framework
Find frequent items
Find frequent itemsets



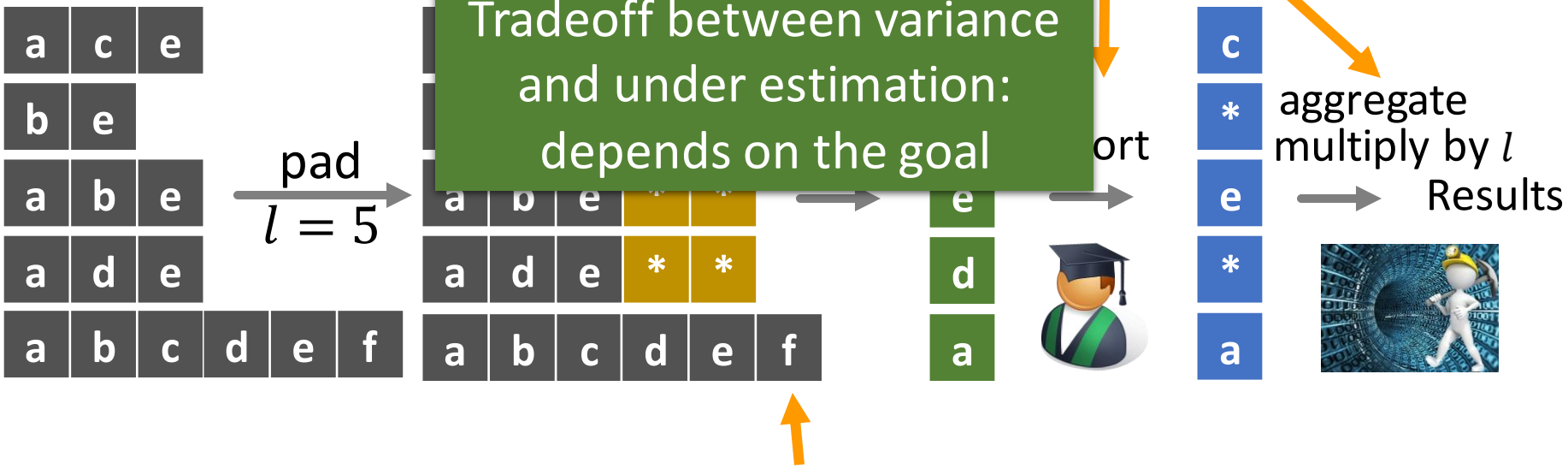
Could find frequent items only.
Left finding frequent itemsets as an open problem.

Sources of error

FO Variance:

To satisfy LDP, permutation introduces noise

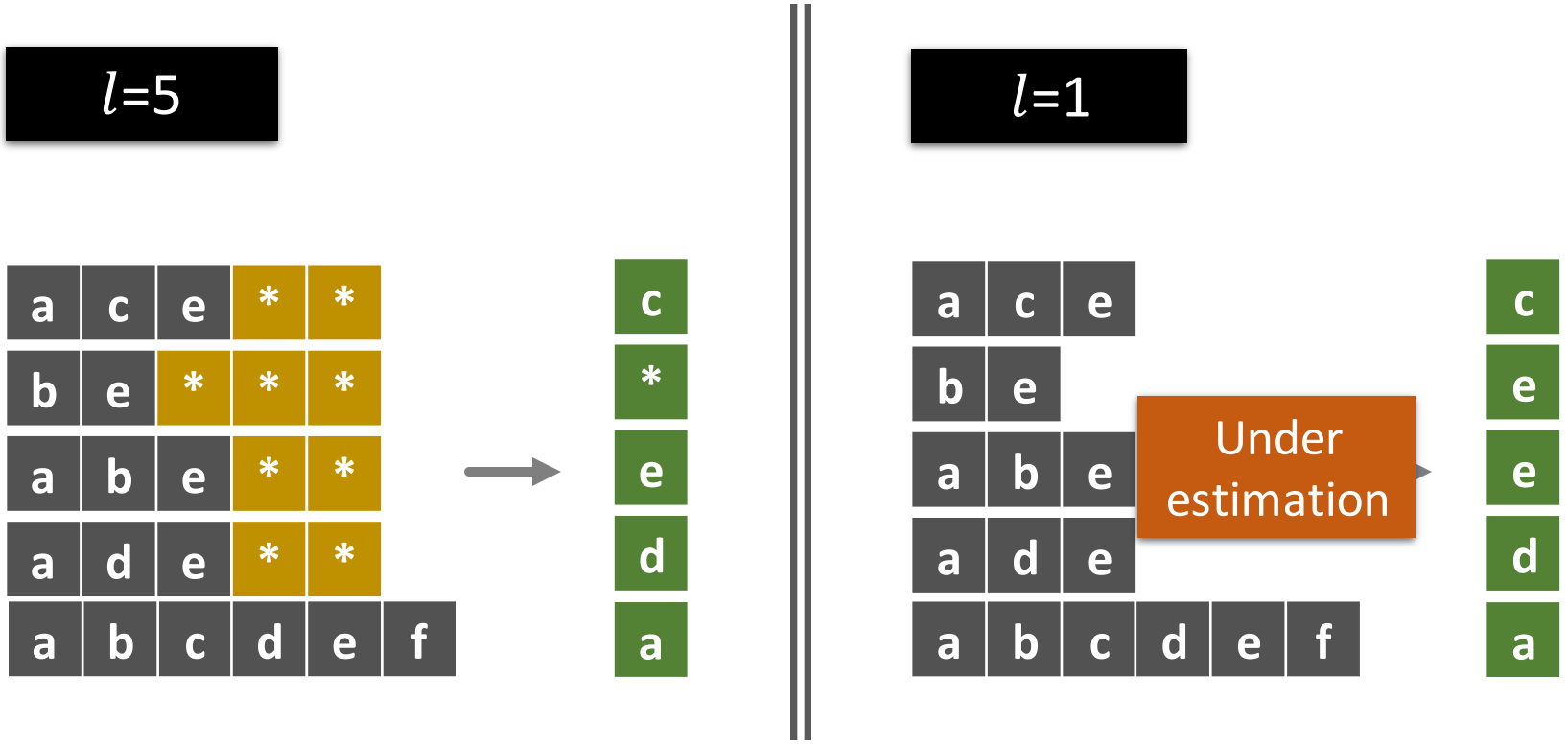
It increases with l, quadratically



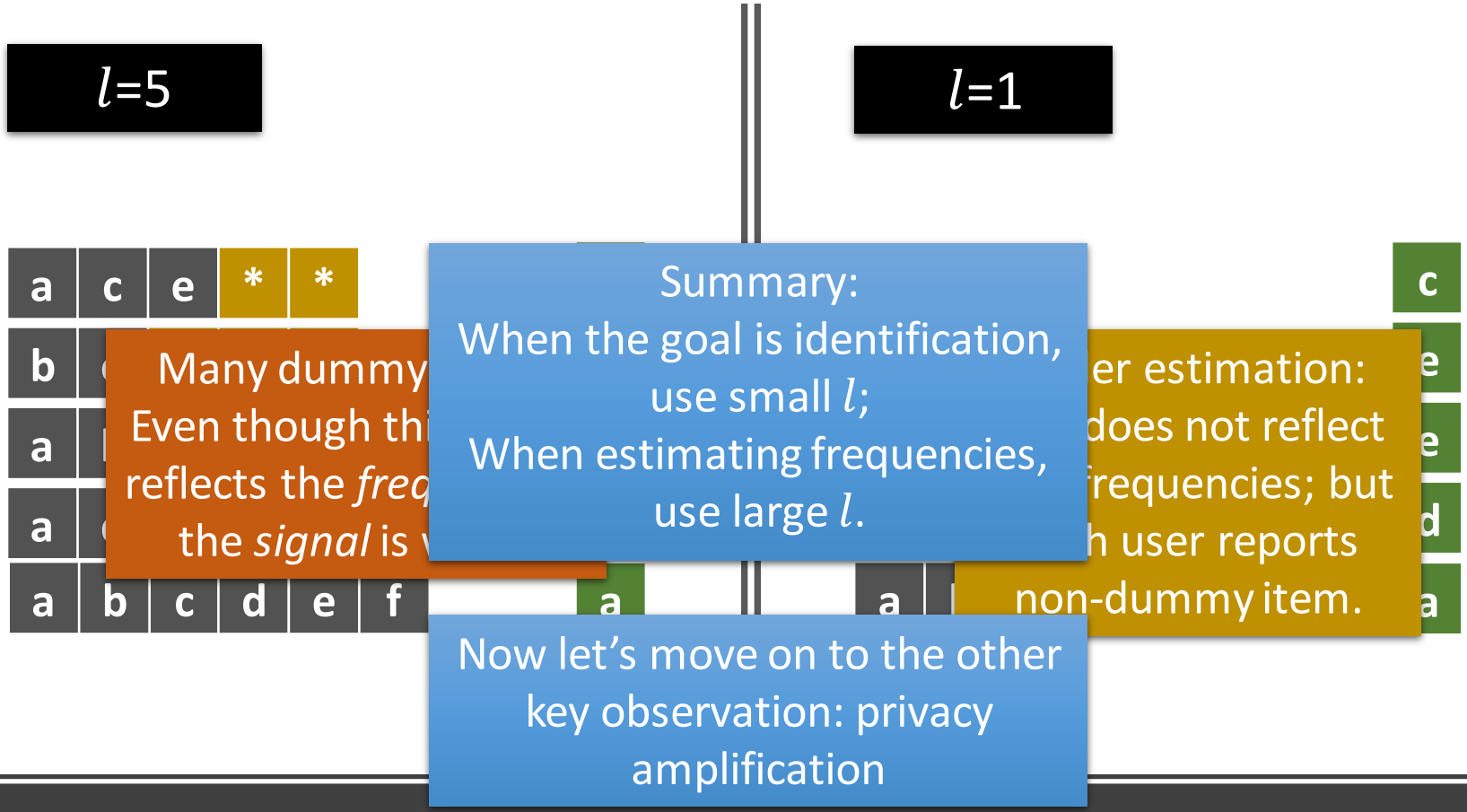
Under Estimation:

Items are selected with 1/6 but multiplied with 5

It decreases when l increases



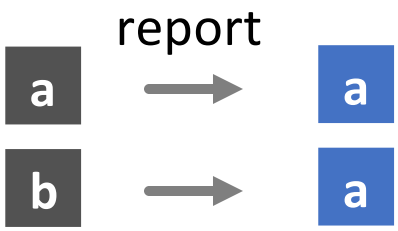
Goal 1: Estimation

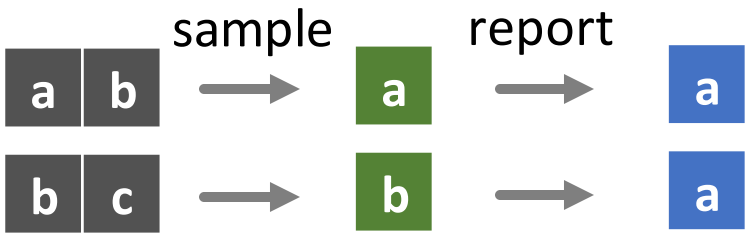


Goal 2: Identification

Privacy Amplification

- LDP bounds the perturbation
- With sampling, things are different



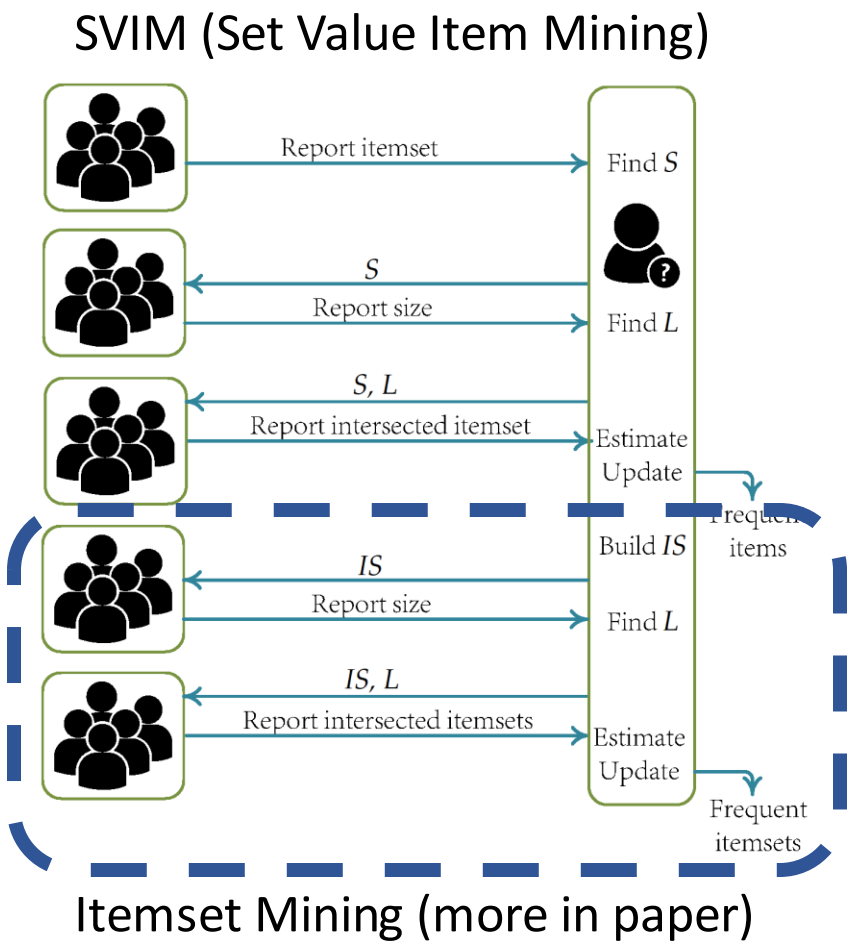


$$\frac{\Pr[P(a) = a]}{\Pr[P(b) = a]} \leq e^{\varepsilon}$$

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

Our Protocol

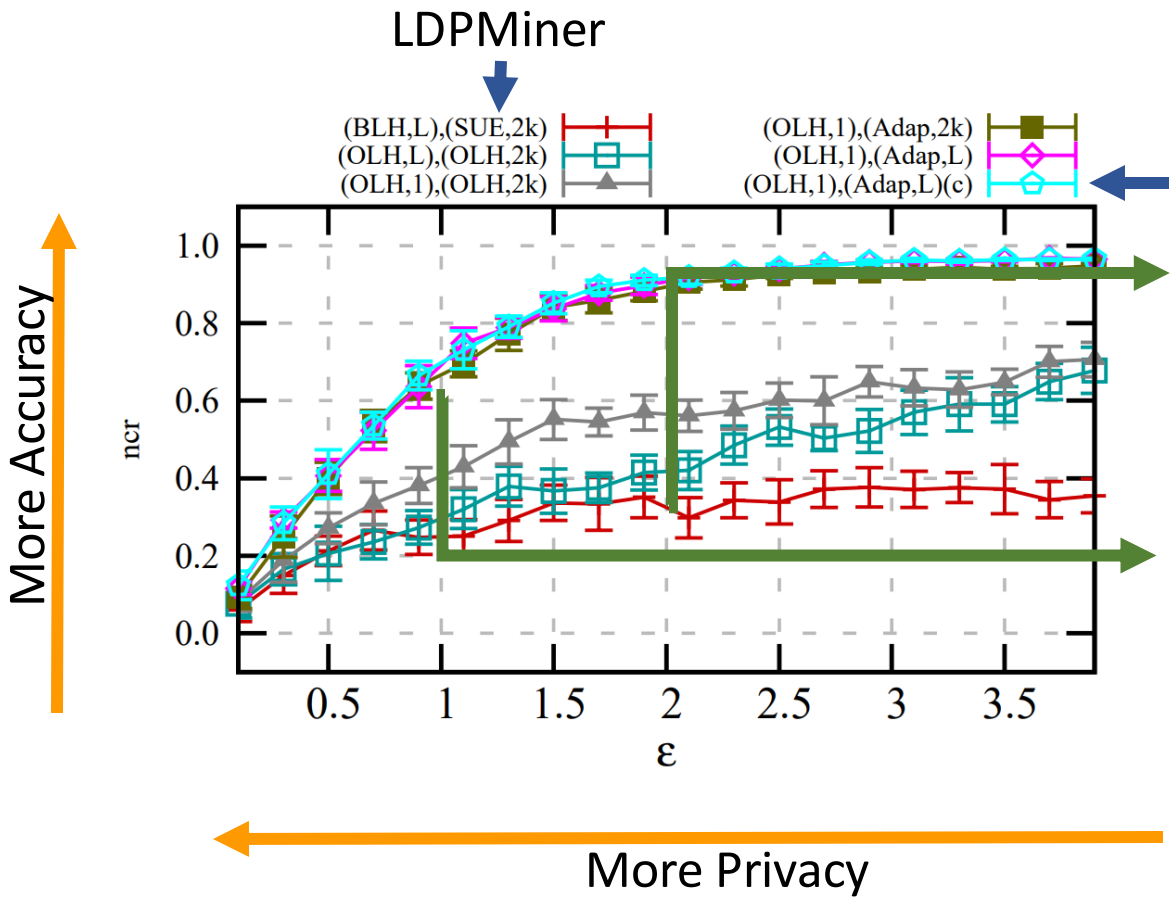
- Phase 1 (identify candidates)
  - Randomly select one ($l = 1$)
  - Report via Frequency Oracle
- Phase 2 (estimate frequency)
  - Intersects v with the k identified ones
  - Pad to l
    - l is the 90 percentile
  - Randomly select one
  - Report via Frequency Oracle
    - With $\varepsilon' = \ln(l(e^{\varepsilon} - 1) + 1)$



Experiments Highlights

- Dataset: POS dataset
  - (also on Kosarak, Online, and Synthetic dataset)
- Competitors: LDPMiner
  - Several improved variants of LDPMiner
- Key Results:
  - Significantly outperforms existing methods
  - Reasonable results in practice

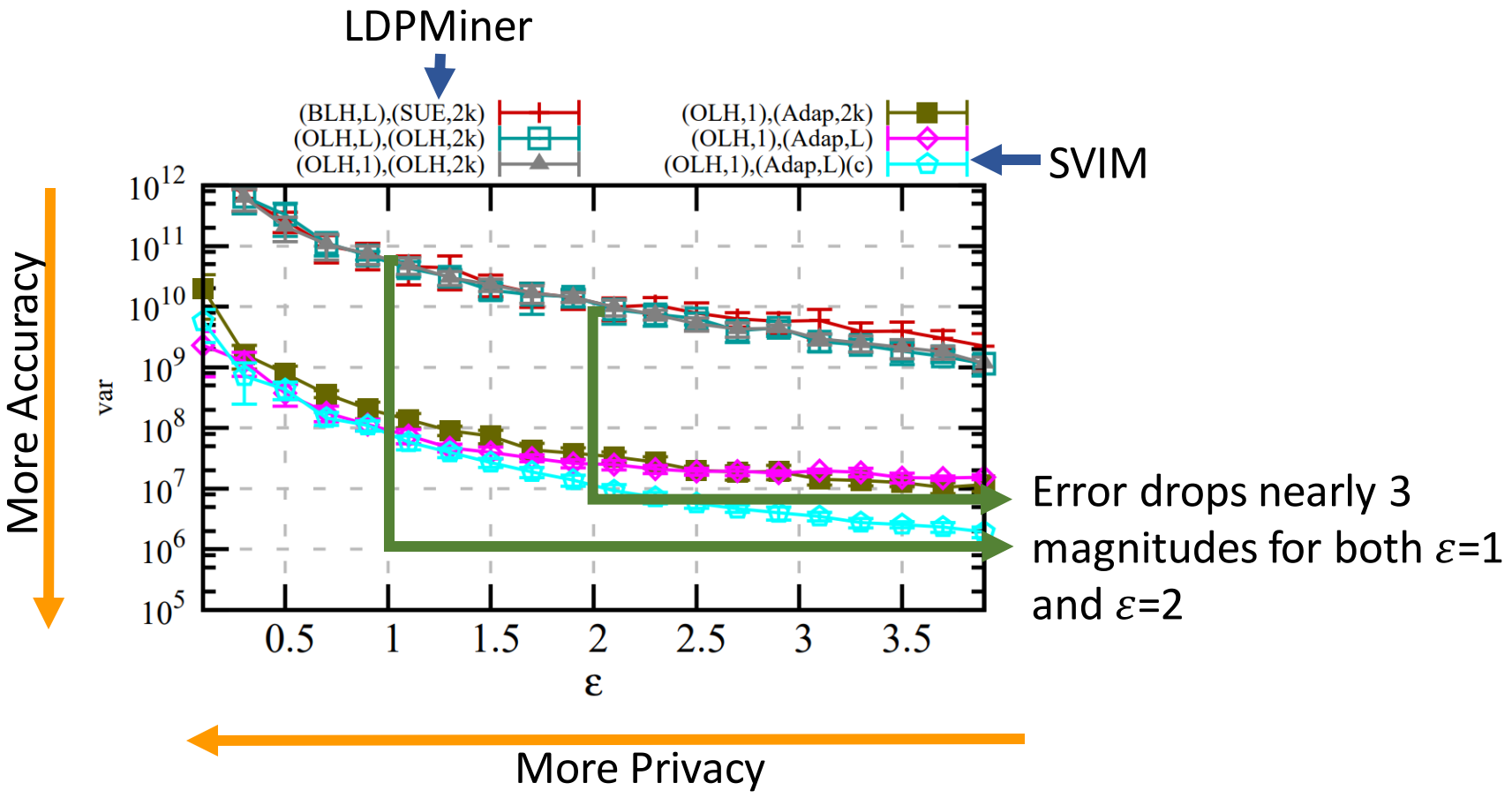
Identifying Frequent Singletons



When ε=2,
SVIM: 0.9, LDPMiner: 0.3

When ε=1,
SVIM: 0.7, LDPMiner: 0.25

Estimating Frequent Singletons



Outline

- Motivation (covered)
- Histogram Estimation (covered)
- Frequent Itemset Mining (covered)
- Other problems

Outline

- Motivation
- Histogram Estimation
- Frequent Itemset Mining
- Other problems

The heavy hitter problem

- Goal: Find the most frequent k values from D
- Assumption: each user has a single value x and it is represented in bits
- Idea (under review):
  - 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
  - Combine the result.

Other interesting problems

- Estimating mean:
  - Goal: Find the mean of continuous values
  - Assumption: Each user has a single value x within the range of $[-1, +1]$
  - Intuition: Report +1 with higher probability if x closer to +1
  - [https://arxiv.org/abs/1606.05053,https://arxiv.org/pdf/1712.01524]
- Stochastic gradient descent
  - Goal: Find the optimal machine learning model
  - Assumption: Each user has a vector $\boldsymbol{x}$
  - Intuition: Bolt-on sgd with noisy update
  - [https://arxiv.org/abs/1606.05053]
- Bound the privacy leakage
  - Goal: Make multiple, periodic collection possible
  - Assumption: Each user has a value $x(t)$ that change with time
  - Intuition: Decide whether to participate based on the current result
  - [https://arxiv.org/abs/1802.07128]
- Publish graph statistics [CCS'17], location data [TIFS'18], marginal [SIGMOD'18], etc

Conclusion

- Motivation
  - Users in LDP do not need to trust server.
  - By permute the secret in a principled way, users maintain deniability, while server can learn statistical information
- Histogram Estimation
  - We surveyed and optimized existing mechanism
  - We proposed Optimized Local Hash
- Frequent Itemset Mining
  - We can mine frequent singleton/itemset
- Other problems
  - Heavy hitter problem
  - Mean estimation
  - Stochastic gradient descent