香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

計算機科學及工程學系 THE DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Personalized Truncation for Personalized Privacy

Dajun Sun, Wei Dong, Yuan Qiu, Ke Yi

Hong Kong University of Science and Technology

Personalized Differential Privacy(PDP) Basics

- Personalized Differential Privacy (PDP)[Jorgensen et al. 2015]
  - For any pair of database instances $\mathbf{I}$, $\mathbf{I}'$, they are neighbors on u ($\mathbf{I}\sim_u\mathbf{I}'$) if they differ by user u's information
  - Each user u specifies his own privacy parameter $\Phi(u)$
  - PDP Definition: A mechanism M is Φ-PDP if for any $\mathbf{I}\sim_u\mathbf{I}'$ and any subset of outputs Y :

$$\Pr[M(\mathbf{I}) \in Y] \leq e^{\Phi(u)} \cdot \Pr[M(\mathbf{I}') \in Y]$$

Problem Definition

- We study the sum estimation problem under PDP
- Assume n users, each user u holds:
  - An integer value $\mathbf{I}(u) \in \{0,1, \dots, B\}$
  - His privacy parameter $\Phi(u)$
- Want to produce a privatized estimation for Sum($\mathbf{I}$)=$\sum_u \mathbf{I}(u)$
- Naïve approach: Add a Laplace noise with scale $\frac{B}{\varepsilon_{\min}}$
  - B is the global sensitivity
  - $\varepsilon_{\min} = \min_u \Phi(u)$ is the strongest privacy requirement

Challenges

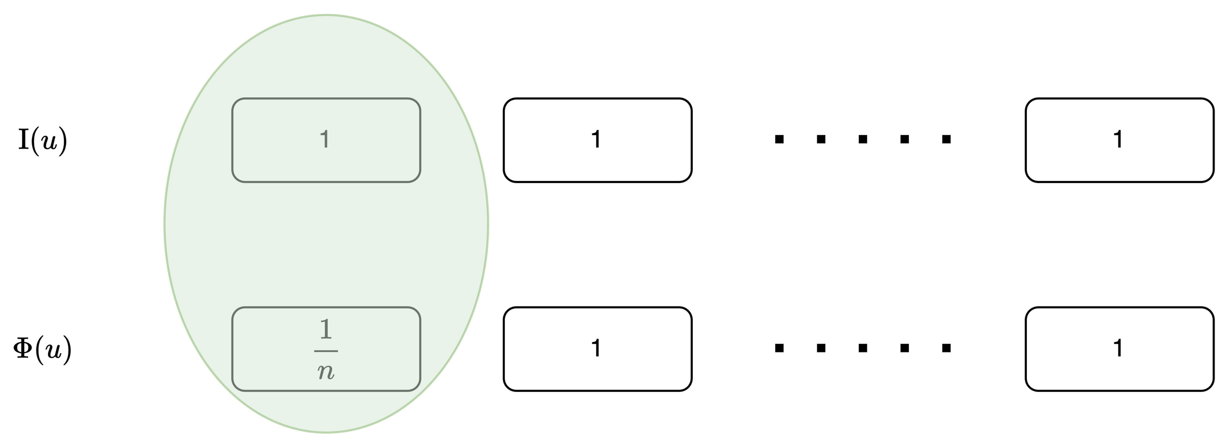
Large global sensitivity B — Previous solution → Truncation based mechanisms

Personalized privacy parameters — Previous solution → PDP Sampling (No error guarantee)

Consider these two challenges together → **Our Proposal: Personalized Truncation**

Warm-up: PDP Bit Counting

- $\mathbf{I}(u)$= 0 or 1 (B=1)
  - The only challenge comes from the PDP model
- Observation: For a user with small $\Phi(u)$, it may be a better choice to **delete** it from $\mathbf{I}$
  - Including it induces at least O(1/ $\Phi(u)$) error due to privacy
  - Deleting it only introduces a bias of at most 1, but required noise can be much smaller



- Naive method: Laplace mechanism with privacy parameter $\varepsilon_{\min} = \min_u \Phi(u)$
  - O(n) noise
- Delete the first user: O(1) noise and bias

Question: How to determine which user should be deleted?

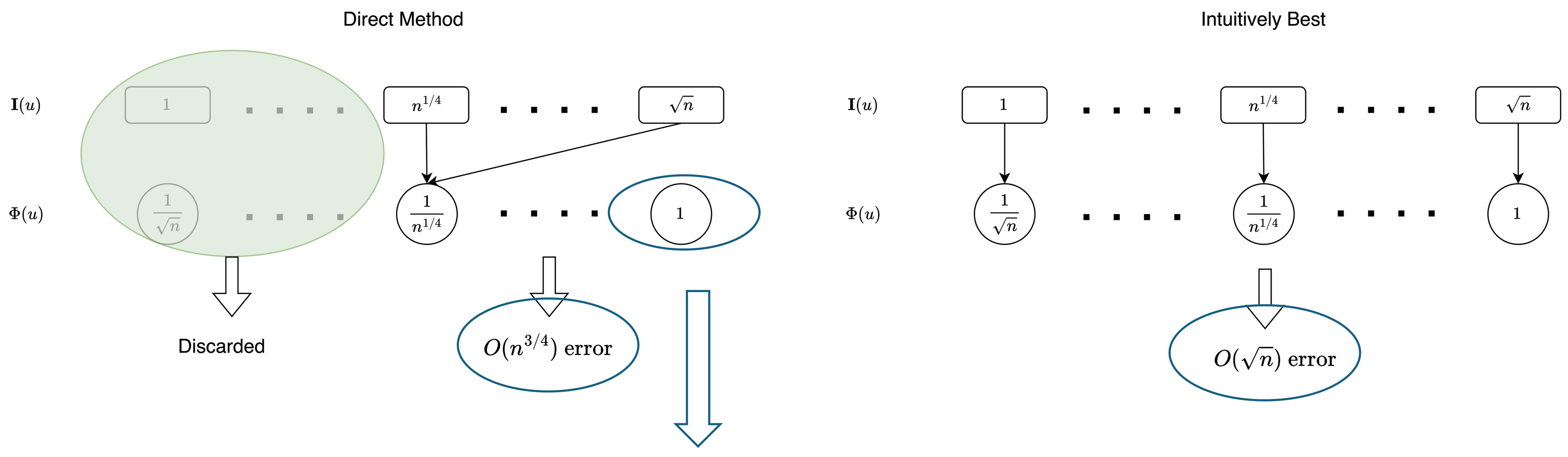
- Assume $\Phi(u_i)$ is sorted in non-decreasing order, if we delete the first k users:
  - Bias O(k), noise O(1/ $\Phi(u_{k+1})$)
- Intuitively, we want to check each $k = 1,2, \dots, n$ and select the best result
  - Searching for k requires n times composition
  - Want to skip some indexes
- Instead of searching an index k, we want to search a threshold ε that
  - Deleting all users with $\Phi(u) \leq \varepsilon$ leads to good result
  - Use doubling search, say, only consider $\varepsilon = \Phi(u_1), 2\Phi(u_1), \dots, 2^{\log\frac{\Phi(u_n)}{\Phi(u_1)}}\Phi(u_1)$
  - Searching for ε requires $\log\frac{\Phi(u_n)}{\Phi(u_1)}$ times composition
  - Achieves $\widetilde{O}(1)$* optimal error

General PDP Sum Problem

Adopt the idea of PDP count? Say,…

- Find a threshold ε and discard all users with $\Phi(u) \leq \varepsilon$
- Apply ε-(standard) DP sum algorithm on the remaining data

This is no longer optimal



Privacy budgets that are greater than ε are wasted

- To make use of each user's privacy, we assign each user a personalized threshold
  - Defined by a truncation vector $\boldsymbol{\tau}$
    - $\boldsymbol{\tau}(u)$ denotes the threshold for user u
    - $\mathbf{I}_{\boldsymbol{\tau}}(u)$=min($\mathbf{I}(u)$, $\boldsymbol{\tau}(u)$)
  - We show adding a noise with scale $s = \max_u \boldsymbol{\tau}(u)/\Phi(u)$ preserves Φ-PDP
- We design a mechanism to search a suitable noise scale s
- Achieves an error with only logarithm overhead compared with the error from the optimal truncation vector
  - $\widetilde{O}(1) *$ optimal error

Personalized Sum under User DP

- It is not always the case that input data can be expressed as $\mathbf{I}(u)$
- In relational database, the results after executing SQL queries are more complicated
  - Each user may own multiple values
  - Each value may be contributed by multiple users
- The user-level PDP model:
  - Each value is denoted as $\mathbf{I}(\boldsymbol{u}) \in \{0,1, \dots, \mathrm{B}\}$
    - $\boldsymbol{u}$ is a subset of users
  - Each user has a privacy parameter $\Phi(u)$

Customer

CK	NK
c1	US
c2	Japan

Supplier

SK	NK
s1	US
s2	UK

Lineitem

SK	OK	Price	Discount
s1	o1	p1	d1
s2	o2	p2	d2
s2	o3	p3	d3

Orders

OK	CK
o1	c1
o2	c1
o3	c2

Join →

$\mathbf{I}$

CK	SK	$\mathbf{I}(\{c,s\})$
c1	s1	p1(1-d1)
c1	s2	p2(1-d2)
c2	s2	p3(1-d3)

Challenge: How to do truncation on $\mathbf{I}$ ($\boldsymbol{u}$)

- Cannot directly do truncation on $\mathbf{I}(\boldsymbol{u})$
  - Consider $u_1, \dots, u_n$ and $\mathbf{I}(u_1, u_i)$=1, $\boldsymbol{\tau}$=1
  - The direct truncation method $\mathbf{I}_{\boldsymbol{\tau}}(\boldsymbol{u})$=min($\mathbf{I}(\boldsymbol{u})$,$u \in \boldsymbol{u}$, $\boldsymbol{\tau}(u)$) will not change any record
    - Sum($\mathbf{I}_{\boldsymbol{\tau}}$) has sensitivity n-1
- We design a linear program to obtain the truncated dataset $\mathbf{I}_{\boldsymbol{\tau}}$

$$\begin{aligned} \max \quad & \sum_{\boldsymbol{u}} \mathrm{I}_{\boldsymbol{\tau}}(\boldsymbol{u}) \\ \text{s.t.} \quad & \sum_{\boldsymbol{u} \ni u} \mathrm{I}_{\boldsymbol{\tau}}(\boldsymbol{u}) \leq \boldsymbol{\tau}(u), \quad u \in \mathcal{U}, \\ & 0 \leq \mathrm{I}_{\boldsymbol{\tau}}(\boldsymbol{u}) \leq \mathrm{I}(\boldsymbol{u}), \quad \boldsymbol{u} \subseteq 2^{\mathcal{U}}. \end{aligned}$$

- Achieves an error of $\widetilde{O}(1) * \min_{\boldsymbol{\tau}}$ Error($\mathbf{I}_{\boldsymbol{\tau}}$)

Experiments

Problem Type	Data	Query Result	Technique	Relative Error(%)	Time(s)
Count	Synthetic	100,000	Naive	12.0	0.000002
			Sampling	16.0	0.03
			PDP EM	0.4	0.5
			PDP Count	1.2	0.04
			PDP Count (with sampling)	**0.5**	**0.2**
Sum	Synthetic	99,933,209	Naive	100	0.08
			Sampling	16.0	1.0
			PDP Sum	**1.1**	**2.0**
	Bank	61,589,682	Naive	100	0.08
			Sampling	19.5	0.19
			PDP Sum	**8.6**	**0.96**

Table 1: Summary of results for count and sum under default setting where $|\mathcal{U}| = 10^6$, the performance of our best PDP mechanism is boldfaced.

Problem Type	Data	Query Result	Query Time	Technique	Relative Error(%)	Time(s)
q_{1-}	Deezer	846,915	1.22	Naive	100	293.4
				Sampling	30.3	220.9
				PDP Query	**4.1**	**131.7**
$q_{\triangle}$		794,210	4.66	Naive	100	6,201
				Sampling	48.2	6,585
				PDP Query	**16.2**	**174.5**
Q_5	TPC-H	240,000	2.55	Naive	100	30.2
				Sampling	30.3	30.5
				PDP Query	**5.5**	**27.3**
Q_7		218,000,000	3.28	Naive	100	1,108
				Sampling	30.4	1,043
				PDP Query	**5.7**	**1,325**

Table 2: Summary of results for PDP query answering under default setting, the performance of our PDP mechanism is boldfaced.