GUPT: Privacy Preserving Data Analysis Made Easy

Prashanth Mohan, Abhradeep Thakurta, Elaine Shi, Dawn Song and David Culler
What is the privacy risk?

Salary Database

Data mining/Machine learning

Mean Clustering Classification
What is the privacy risk?

I want information about Alice’s salary

Objective:
Do not leak “too-much” information about individual salaries
How can we define privacy?

From the output, any “neighboring pair” DB1 and DB2 should be indistinguishable.
Differential privacy [DMNS06]

\[
\Pr\left[ A(D) \in S \right] \leq \exp(\epsilon) \times \Pr\left[ A(D') \in S \right]
\]

- **Privacy budget**
- **Randomized algorithm**
- **Neighbors:** two datasets differing in exactly one entry
- **Any measurable set**

\[
A(D) = f(D) + \text{Lap}\left( \frac{\Delta f}{\epsilon} \right)^d
\]

\[
\Delta f = \max_{D,D'} \left\| f(D) - f(D') \right\|_1
\]

\[
\text{Lap}(\lambda) \sim \frac{1}{2\lambda} e^{-\frac{|x|}{\lambda}}
\]
Current approaches [McSherry09, RSKSW10, HPN11]
Current approaches [McSherry09, RSKSW10, HPN11]
Challenge: Making DP usable

• Executing unmodified code/ binaries

• Privacy budget allocation
  – The privacy parameter $\epsilon$ (also called privacy budget) is a limited quantity
  – $\epsilon$ needs to be allocated effectively for a given task

• Improve accuracy of results

• Handle side channel attacks
Contributions

**GUPT**: platform for differentially private execution of unmodified user code

1. **Improve output accuracy**: resampling, optimal block size estimation

2. **Usability**: describing privacy budget in terms of accuracy, privacy budget allocation

3. **Protection against side-channel attacks**: state attack, privacy-budget attack, timing attack
Talk outline

• System design

• Improving result accuracy

• Privacy budget maintenance

• Evaluation
1. User defined function (UDF) be executed ($f$)
2. Accuracy ($\rho$)/privacy budget ($\epsilon$)

Data Owner

1. Dataset
2. Overall privacy budget ($\epsilon_{Total}$)

GUPT

Data Analyst

Differentially private answer
1. User defined function (UDF) be executed ($f$)
2. Accuracy ($\rho$)/privacy budget ($\epsilon$)

Differentially private answer
Main idea: Sample and Aggregate [NRS07, Smith11]

Data Set

D_1
D_2
D_3
D_4

\ldots

D_k

Isolated Execution Chambers
Main idea: Sample and Aggregate [NRS07, Smith11]

Implication for GUPT: No need to compute the sensitivity $\Delta f$ for the user code $f$
Sample and Aggregate: Algorithm

1. Partition the dataset into blocks $D_1, ..., D_k$ of equal size
2. Clamp the output on each block $f(D \downarrow i)$ between predefined bounds $min$ and $max$
3. Output $1/k \sum f(D \downarrow i) + Lap(d|max - min|/k\epsilon)^d$

Privacy:
- Algorithm satisfies $\epsilon$-differential privacy

Accuracy:
- In this talk we show experimental results
- For theoretical bounds see either our paper or [Smith11]
Where is error introduced?

Data Set

Differentially Private Output

D_1, D_2, D_3, D_4, ..., D_k

Average

Estimation Error

Noise

D_1, D_2, D_3, D_4, ..., D_k

Differentially Private Output
Noise in Sample and Aggregate

Number of blocks: \( k = \frac{n}{\beta} \)

\[
A(D) = \frac{1}{k} \sum f(D_i) + \text{Lap}\left( \frac{d \cdot |\text{max} - \text{min}|}{k\varepsilon} \right)^d
\]
Reduce variance by resampling

- Each entry in $D$ appear in exactly $\ell$ blocks
- Consider $k' = \ell n / \beta$ number of blocks
- Each block contains exactly one copy of each data entry
Reduce variance by resampling

- Each entry in $D$ appear in exactly $\ell$ blocks
- Consider $k' = \ell n / \beta$ number of blocks
- Each block contains exactly one copy of each data entry
Reduce variance by resampling

- Changing one data entry affects only $\ell$ blocks
Reduce variance by resampling

Number of blocks

\[ k' = \frac{\ln \beta}{\beta} \]

\[ A(D) = \frac{1}{k'} \sum f(D_i) + \text{Lap} \left( \frac{dl | \max - \min |}{k' \epsilon} \right)^d \]
Reduce variance by resampling

\[ \text{Data Set} \]

\[ D_1 \quad D_2 \quad D_3 \quad D_4 \quad \ldots \quad D_{k'} \]

\[ n \]

\[ \beta \]

**Advantage:**

- Reduce variance in the output without increasing the noise
Recap

• Introduced GUPT with its essential building blocks

• Discussed the main algorithmic idea (Sample and Aggregate [NRS07, Smith11])

• Proposed an idea to reduce the variance in the output via resampling
New model: Aging of sensitivity

- A new model where the privacy concern of data degrades over time

- Implications for GUPT:
  - Estimating optimal block size
  - Relate privacy budget to accuracy requirement
New model: Aging of sensitivity

Dataset: $D_{\text{TOTAL}}$

Dataset: $D_{\text{OLD}}$

Dataset: $D$

- $D_{\text{OLD}}$ has little or no privacy concern as compared to $D$
- $D_{\text{OLD}}$ is used for setting optimal parameters for GUPT
Estimation of DP parameters

Data Analyst

Code $f$
Expected accuracy/privacy budget

Preprocess

Block size, privacy budget policy

GUPT

Private output

Aged dataset $D_{OLD}$

Real dataset
What is the right block size?

- There is a trade-off between estimation error and noise
- Select a block size $\beta$ that minimizes the total error

Total Error:

$$|\beta/\text{D}^{\text{OLD}}| \sum f(D_i) - f(D^{\text{OLD}})| + \text{Noise}(\text{min}, \text{max}, \epsilon, \beta)$$

Estimation Error

Noise
Identifying optimal block size
Privacy budget management

End-Users understand accuracy goal better than privacy budget $\epsilon$

Provide (approx.) 90% accurate answer for 90% of my queries

$\epsilon$-differentially private output

Privacy budget $\epsilon$ selected automatically based on accuracy requirement
Accuracy vs privacy budget
Experimental result: Output accuracy

![Graph showing normalized intra-cluster variance vs. privacy budget (ε). The graph compares Baseline ICV, GUPT-loose, and GUPT-tight.]
Limitations

• Works only for outputs with fixed dimensions
  – Average
  – Median, percentile
  – K-means clustering
  – Logistic regression
  – MLE

• Expects an implicit ordering of outputs

• Needs reasonably large datasets

• Works well for applications which estimates property of the data generating distribution
Future work

• **Most exciting:** Explore the use of GUPT for time series data

• Reduce error dependence on the dimensionality of the output

• Estimate the optimal block size and privacy budget privately (instead of using Aging of sensitivity model)
Public release of GUPT available at https://github.com/prashmohan/GUPT
Two kind of data services

**Type 1:** Silo-based services

<table>
<thead>
<tr>
<th>Bob's financial documents</th>
<th>Tax application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommendations</td>
</tr>
<tr>
<td>Traffic advice</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
</tr>
<tr>
<td>Charlie</td>
</tr>
<tr>
<td>David</td>
</tr>
<tr>
<td>....</td>
</tr>
</tbody>
</table>
Two kind of data services

Type 2: Data intelligence

Recommendations
Traffic advice

This talk
Related Work: PINQ [McSherry09]

- Flexible programming layer abstraction
- Privacy operations mostly transparent to programmers
Related Work: Airavat [RSKSW10]

Dataset

Trusted Reducer

Untrusted Mapper

Data Data Data Data Data
Increased lifetime of privacy budget

Normalized privacy budget lifetime

- **GUPT constant $\epsilon=1$**
- **GUPT variable $\epsilon$**
- **GUPT constant $\epsilon=0.3$**
Side-channel attacks [HPN11]

- **Timing Attack:** Use *computation time* as a side-channel information to identify a data record

- **State Attack:** Use *global state variable* for microqueries to identify a data record

- **Privacy Budget Attack:** Use the *privacy budget* $\epsilon$ to leak information
## Side-channel attacks

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing attack</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>State attack</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Privacy Budget attack</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

**Note:** Fuzz has 2.5-6x computation overhead due to protection against side-channel attacks
Protection against timing attack

- Make each block take exactly the same computation time $T$.
- If any block takes more than $T$, then output a default value.

Data Set

$D_1$ $D_2$ $D_3$ $D_4$ $\cdots$ $D_k$

Average

Differentially Private Output
Budget management across iterations is hard

// runs one step of the iterative k-means algorithm.
public static void kMeansStep(PINQueryable<double[]> input, double[][], centers, double epsilon)
{
    // partition data set by the supplied centers; somewhat icky in pure LINQ...
    // (and it assumes centers[0] exists )
    var parts = input.Partition(centers, x => NearestCenter(x, centers));
    // update each of the centers
    foreach (var center in centers)
    {
        var part = parts[center];
        foreach (var index in Enumerable.Range(0, center.Length))
        {
            center[index] = part.NoisyAverage(epsilon, x => x[index]);
        }
    }
}

Private k-Means clustering code in PINQ [McSherry09]

- Total privacy budget $\varepsilon = \sum \downarrow i = 1, \cdots, m \varepsilon \downarrow i$
- $m$ is the number of iterations
**k-means clustering:** Comparison between PINQ and GUPT

Take away: The system should internally **manage** the privacy budget