Improving Usability of Differential Privacy at Scale

PEPR '20, October 2020

Miguel Guevara
Product Manager

Milinda Perera
Software Engineer
Usability Problem of Differential Privacy
A simple dataset ...

Movie ratings:

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Date</th>
<th>Rating</th>
<th>Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>81478</td>
<td>1999-12-15</td>
<td>3</td>
<td>Mulan</td>
</tr>
<tr>
<td>92729</td>
<td>2000-10-24</td>
<td>5</td>
<td>The Piano</td>
</tr>
<tr>
<td>245371</td>
<td>2001-01-12</td>
<td>1</td>
<td>Office Space</td>
</tr>
<tr>
<td>383404</td>
<td>2005-02-02</td>
<td>4</td>
<td>The Matrix</td>
</tr>
</tbody>
</table>
A simple aggregation ...

Movie counts by date and rating:

```
SELECT
  date, rating,
  COUNT(movie_id) AS movie_count
FROM movie.ratings
GROUP BY 1, 2;
```

<table>
<thead>
<tr>
<th>Date</th>
<th>Rating</th>
<th>Movie Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999-12-15</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2000-10-24</td>
<td>5</td>
<td>60</td>
</tr>
<tr>
<td>2001-01-12</td>
<td>1</td>
<td>793</td>
</tr>
<tr>
<td>2005-02-02</td>
<td>4</td>
<td>8043</td>
</tr>
</tbody>
</table>
Same aggregation with anonymization ...

Using a SQL engine* to query with Differential Privacy:

```
SELECT WITH ANONYMIZATION OPTIONS (epsilon = 1.0986, delta = 0.00001)
    date, rating,
    ANON_COUNT(movie_id CLAMPED BETWEEN 0 AND 70) AS movie_count
FROM movie.ratings
GROUP BY 1, 2;
```

* Differentially Private SQL with Bounded User Contribution
  Wilson et al., PoPETs, 2020
Using a SQL engine to query with Differential Privacy:

```
SELECT WITH ANONYMIZATION OPTIONS(epsilon = 1.0986, delta = 0.00001)
    date, rating,
    ANON_COUNT(movie_id CLAMPED BETWEEN 0 AND 70) AS movie_count
FROM movie.ratings
GROUP BY 1, 2;
```
The usability problem

I just want to Anonymize a dataset
Quantifying Privacy and Utility
Quantifying Privacy and Utility

Why? Bridge the usability gap of Differential Privacy

How?

- Define privacy vs utility metrics
- Provide infrastructure to safely and efficiently compute them at scale
- Allow self-service model
Demo!
System Architecture
System Architecture

1. Source Data → SQL Engine
2. Source Data → “Sketches”
3. “Sketches” → SQL Engine

Create Analysis

View Analysis

Script

UI Server

Browser

Analysis
Highlights

- Median query latency within **seconds**!
- End-to-end analysis for most datasets takes only **minutes**
- **Intuitive** utility metrics for teams

"This is super useful for tuning parameters, we were missing something just like that :)

- A happy product team
Future Work
Future Work

- Open source this work
- Local Differential Privacy
- More Functions (AnonMean, AnonMedian, AnonQuantiles)
Thank you!