Oblivious Sampling Algorithms for Private Data Analysis

Sajin Sasy and Olga Ohrimenko

Contributions
- Introduce Private-Sampling-Based Query Framework
- Take advantage of Differential Privacy and Privacy Amplification from sampling
- Design secure sampling for hiding sample identity: oblivious sampling
- Design efficient dataset sampling algorithms
- Experimental evaluations of accuracy of machine learning models trained with minibatches produced by sampling instead of shuffling

Goal & Motivation
- Enable data scientists to query data while providing strong privacy guarantees on user data.
- Trusted Execution Environments (TEE) restricts data access and protects data while its computed upon.
- TEEs can leak data access patterns which lead to privacy loss.

Privacy via Sampling and Differential Privacy
Consider a randomized mechanism \( \mathcal{M} \) that is \((\epsilon, \delta)\)-DP, and a mechanism \( \mathcal{M'} \) that uses samples of size m from a dataset with n elements using any of the following methods:
1) Sampling without Replacement (SWO)
2) Poisson Sampling
3) Shuffle-based Sampling
Informally, Privacy Amplification of Differential Privacy states that \( \epsilon \times \mathcal{M'} \) with SWO or Poisson sampling provides an order of \( O(\sqrt{n/m}) \) smaller epsilon than the popularly used Shuffle-based sampling.

Algorithm for Oblivious Sampling Without Replacement (SWO)

**Input:** A dataset (D) of n records

**Goal:** Obtain samples of size m. (m=2)

**Constraint:** Hide sample identity. Memory access patterns of the algorithm execution should not reveal any information about the elements in any sample.

**Output:** k = n/m samples of size m from D. (k=3)

**Algorithm Correctness**
We prove that the algorithm returns samples of size m drawn truly randomly from the n elements, up to an injective and random key mapping.

Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Shuffle</th>
<th>SWO</th>
<th>Poisson</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MNIST</strong></td>
<td>97.50 (98.33)</td>
<td>97.43 (98.31)</td>
<td>97.47 (98.31)</td>
</tr>
<tr>
<td><strong>DP MNIST</strong></td>
<td>94.06 (94.10)</td>
<td>94.03 (94.05)</td>
<td>94.10 (94.01)</td>
</tr>
<tr>
<td><strong>CIFAR-10</strong></td>
<td>79.6 (83.2)</td>
<td>79.0 (82.9)</td>
<td></td>
</tr>
<tr>
<td><strong>DP CIFAR-10</strong></td>
<td>73.4 (72.3)</td>
<td>72.5 (71.0)</td>
<td></td>
</tr>
<tr>
<td><strong>( \epsilon )</strong></td>
<td>9.39</td>
<td>2.13</td>
<td>0.82</td>
</tr>
</tbody>
</table>

Full Paper
- Oblivious Poisson sampling algorithm
- Security analysis
- More detailed experiments

For more information: ssasy@uwaterloo.ca    oohrim@microsoft.com