Differential Privacy in the Streaming Model

Jalaj Upadhyay

Pennsylvania State University

January 07, 2016
*Analyst* wishes to get some task done on the *Database*.
Differential Privacy: The Framework

Privacy guard provides privacy of individuals in the Database
Differential Privacy: The Framework

The privacy guard performs the task on the Database
The idea is that absence or presence of an individual entry should not change the output "by much"
The idea is that absence or presence of an individual entry should not change the output "by much"

**Definition.** A randomized algorithm, \( \mathcal{M} \), gives \((\varepsilon, \delta)\)-differential privacy if, for all "neighboring data," \( \mathcal{D} \) and \( \widetilde{\mathcal{D}} \), and for all \( S \subseteq \text{Range}(\mathcal{M}) \),

\[
\Pr [\mathcal{M}(\mathcal{D}) \in S] \leq \exp(\varepsilon) \Pr [\mathcal{M}(\widetilde{\mathcal{D}}) \in S] + \delta
\]
The idea is that absence or presence of an individual entry should not change the output “by much"  

**Definition.** A randomized algorithm, $\mathcal{M}$, gives $(\varepsilon, \delta)$-differential privacy if, for all “neighboring data," $D$ and $\tilde{D}$, and for all $S \subseteq \text{Range}(\mathcal{M})$,

$$
\Pr[\mathcal{M}(D) \in S] \leq \exp(\varepsilon) \Pr[\mathcal{M}(\tilde{D}) \in S] + \delta
$$

We restrict how the privacy guard can access the database
Differentially Private Streaming Model of Computation

Privacy Guard

Private Matrix

\[
\begin{array}{ccc}
8 & 1 & 6 \\
3 & 5 & 7 \\
4 & 9 & 2 \\
5 & 3 & 7 \\
6 & 2 & 1 \\
2 & 6 & 7 \\
2 & 1 & 9 \\
1 & 1 & 6 \\
1 & 3 & 1 \\
\end{array}
\]
Differentially Private Streaming Model of Computation

Privacy Guard

1. Operates on the stream
2. Update the data structure

<table>
<thead>
<tr>
<th>8</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Private Matrix

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>7</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{pmatrix}
4 \\
6 \\
1
\end{pmatrix}
\]
Differentially Private Streaming Model of Computation

Privacy Guard

• Operates on the stream

Private Matrix

\[
\begin{pmatrix}
4 & 3 \\
6 & 2 \\
1 & 8
\end{pmatrix}
\]

⇒

\[
\begin{pmatrix}
1 & 1 \\
5 & 1 \\
9 & 3 \\
3 & 2 \\
6 & 1 \\
1 & 3
\end{pmatrix}
\]
Privacy Guard

- Operates on the stream
- Update the data structure

\[
\begin{pmatrix}
4 & 3 & 4 \\
6 & 2 & 8 \\
1 & 8 & 9
\end{pmatrix}
\]
An analyst comes along
An analyst comes along

request to do a task
An analyst comes along

request to do a task

- uses
  \[
  \begin{pmatrix}
  4 & 3 & 4 \\
  6 & 2 & 8 \\
  1 & 8 & 9
  \end{pmatrix}
  \]
An analyst comes along

request to do a task

performs the task

• uses

\[
\begin{pmatrix}
4 & 3 & 4 \\
6 & 2 & 8 \\
1 & 8 & 9
\end{pmatrix}
\]
cannot figure out
individual
information
Differentially Private Streaming Model of Computation

cannot figure out individual information

Privacy goal achieved
Following are the extra parameters

1. **number of passes** over the matrix
2. **space requirement of the data structures**
3. **time required to update the data structures**
Following are the extra parameters

1. number of passes over the matrix
2. `space requirement` of the data structures
3. time required to update the data structures
Following are the extra parameters:

1. number of passes over the matrix
2. space requirement of the data structures
3. time required to update the data structures
The Main Idea

Non-private Setting
The Main Idea

Non-private Setting

Data-structure is a sketch generated using random matrix
Non-private Setting

Data-structure is a sketch generated using random matrix

Efficient one-pass streaming algorithms
The Main Idea

Private Setting
The Main Idea

Private Setting

Special distribution of random matrices
The Main Idea

Private Setting

Special distribution of random matrices

Sketch generated using a random matrix picked from this distribution
The Main Idea

Private Setting

Special distribution of random matrices

Sketch generated using a random matrix picked from this distribution

Differentially private one-pass streaming algorithms
The Main Idea

Private Setting

Special distribution of random matrices

+ Sketch generated using a random matrix picked from this distribution

⇓

Differentially private one-pass streaming algorithms
First Approach

Streaming Private Sketch Generator (PSG$_1$)

- Pick a random Gaussian matrix $\Phi$
- Multiply $\Phi$ to the streamed column

Theorem. If the singular values of the streamed matrix to PSG$_1$ algorithm are at least $\sigma_1 = \frac{4\sqrt{r}\log(2/\delta) \log(r/\delta)}{\varepsilon}$, then PSG$_1$ preserves ($\varepsilon,\delta$)-differential privacy.

Similar result was shown by [BBDS12] for non-streaming algorithms.
First Approach

Streaming Private Sketch Generator (PSG₁)

Pick a random Gaussian matrix $\Phi$
Multiply $\Phi$ to the streamed column

**Theorem.** If the singular values of the streamed matrix to PSG₁ algorithm are at least

$$\sigma_1 := \left(4\sqrt{r \log(2/\delta) \log(r/\delta)}\right)/\varepsilon,$$

then PSG₁ preserves $(\varepsilon, \delta)$-differential privacy.
First Approach

Streaming Private Sketch Generator (PSG₁)

Pick a random Gaussian matrix $\Phi$
Multiply $\Phi$ to the streamed column

**Theorem.** If the singular values of the streamed matrix to PSG₁ algorithm are at least

$$\sigma₁ := \left( \frac{4\sqrt{r \log(2/\delta) \log(r/\delta)}}{\varepsilon} \right),$$

then PSG₁ preserves $(\varepsilon, \delta)$-differential privacy

Similar result was shown by [BBDS12] for non-streaming algorithms
First Approach

Streaming Private Sketch Generator \((\text{PSG}_2)\)

Pick a random Gaussian matrix \(\Phi\)
Multiply \(\Phi^T \Phi\) to the streamed column

**Theorem.** If the singular values of the streamed matrix to the \(\text{PSG}_2\) algorithm are at least
\[
s_2 := \frac{4r \log(r/\delta)}{\varepsilon},
\]
then \(\text{PSG}_2\) preserves \((\varepsilon, \delta)\)-differential privacy.
A Meta Algorithm

- Get a stream in the form of column vector
A Meta Algorithm

- Get a stream in the form of column vector
- Perturb the vector to lift the singular values
A Meta Algorithm

- Get a stream in the form of column vector
- Perturb the vector to lift the singular values
- Feed it to $\text{PSG}_1$ or $\text{PSG}_2$
A Meta Algorithm

- Get a stream in the form of column vector
- Perturb the vector to lift the singular values
- Feed it to $\text{PSG}_1$ or $\text{PSG}_2$
- Perform any post-processing
Another Candidate for $\Phi$: Update-time Efficiency

1. Pick $\{g_1, \cdots, g_n\} \sim N(0, 1)^n$

2. Divide it into $r$ equal blocks of vectors $\Phi_1, \cdots, \Phi_r$.

$$
P := \begin{pmatrix}
\Phi_1 & 0^{n/r} & \cdots & 0^{n/r} \\
0^{n/r} & \Phi_2 & \cdots & 0^{n/r} \\
\vdots & \vdots & \ddots & \vdots \\
0^{n/r} & \cdots & 0^{n/r} & \Phi_r
\end{pmatrix}
$$

Compute $\Phi = \sqrt{\frac{1}{r}}PPW$, where $W$ is a randomized Hadamard matrix and $\Pi$ is a permutation matrix.
Thank you for your attention