Data Poisoning Attacks to Local Differential Privacy Protocols

Xiaoyu Cao, Jinyuan Jia, Neil Gong
Data Statistics Collection

What is the most popular item this week?

Answer
Local Differential Privacy (LDP)

Untrusted

Noise
Data Poisoning Attacks

- Noise

Frequency Estimation
Heavy Hitter Identification
Frequency Estimation

• Each user holds 1 item $v \in [d] = \{1, 2, \ldots, d\}$
• Estimate frequency of each item
• Three key steps
  • **Encode**: item $v \rightarrow$ encoded value $x \in D$
  • **Perturb**: randomly perturbs $x \rightarrow y$ in $D$
  • **Aggregate**: estimate frequency from $y$
• $\varepsilon$-LDP
  • $\forall v_1, \forall v_2, \forall y$, $\Pr(PE(v_1) = y) \leq e^\varepsilon \Pr(PE(v_2) = y)$
  • PE: perturbed encoded
Pure LDP

• An LDP protocol is pure if
  • \( \exists 0 < q < p < 1, \forall v_1 \neq v_2 \)
  • \( \Pr(\text{PE}(v_1) \in \{ y | v_1 \in S(y) \}) = p \)
  • \( \Pr(\text{PE}(v_2) \in \{ y | v_1 \in S(y) \}) = q \)

• \( S(y) \) is the set of items that \( y \) supports
  • \( y \) supports \( v \): \( y \) “votes for” \( v \) in \textbf{Aggregate} step
  • Protocol-dependent

• \textbf{Aggregate} as follows
  • \( \tilde{f}_v = \frac{\frac{1}{n} \sum_{i=1}^{n} 1_{S(y_i)(v)}}{p - q}, \quad 1_{S(y_i)(v)}: \text{indicator function} \)
kRR

• Encode
  • \( v \rightarrow v \)

• Perturb
  • \( y = v \) w.p. \( p = \frac{e^\varepsilon}{d-1+e^\varepsilon} \),
  • \( y = a \) w.p. \( q = \frac{1}{d-1+e^\varepsilon}, \forall a \neq v \)

• Aggregate
  • \( S(y) = \{ y \} \)
OUE

• **Encode**
  - $v \rightarrow d$-bit one-hot binary vector

• **Perturb**
  - $y_v = 1$ w.p. $p = \frac{1}{2}$
  - $y_i = 1$ w.p. $q = \frac{1}{e^{\varepsilon+1}}, \forall i \neq v$

• **Aggregate**
  - $S(y) = \{v | y_v = 1\}$

---

e.g., $d = 8, v = 4$

0 0 0 1 0 0 0 0

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
\end{array}
\]
OLH

• Encode
  • $v \rightarrow (H, h)$. H: hash function. h: hash value.
  • $h \in [d']$, $d' = e^\varepsilon + 1 < d$

• Perturb: only hash value
  • $y = (H, h)$ w.p. $p' = \frac{e^\varepsilon}{e^\varepsilon + d' - 1}$,
  • $y = (H, a)$ w.p. $q' = \frac{1}{e^\varepsilon + d' - 1}$, $\forall a \neq h$

• Aggregate
  • $S(y) = \{v | H(v) = a\}$ for $y = (H, a)$
  • $p = p'$, $q = 1/d'$
Heavy Hitter Identification

• Find the most frequent $k$ items

• Prefix Extending Method (PEM)
  • Users are divided into groups
  • Iteratively find portions of frequent values
  • Each group uses OLH
Threat Model

• Goal: promote a set of target items $T$
  • Frequency estimation: increase estimated frequency
  • Heavy hitter identification: promote to be heavy hitters

• Background knowledge
  • LDP protocol

• Capability
  • Inject fake accounts
Metrics

• Frequency estimation: **Overall gain**
  • Frequency gain \( \Delta \tilde{f}_t = \tilde{f}_{t,a} - \tilde{f}_{t,b} \), \( \tilde{f}_{t,a} \): after attack, \( \tilde{f}_{t,b} \): before attack
  • Overall gain \( G = \sum_{t \in T} \mathbb{E}(\Delta \tilde{f}_t) \)
  • \( G \) depends on the set of attacker-crafted perturbed values \( Y \)
  • Attacker manipulates **Encode/Perturb** to craft \( Y \) that maximizes \( G \)

• Heavy hitter identification: **Success rate**
  • The fraction of target items identified as heavy hitters
Attack Frequency Estimation

• Random perturbed-value attack (RPA)
  • Each fake user randomly selects $y \in D$

• Random item attack (RIA)
  • Each fake user randomly selects a target item
  • Follow the LDP protocol to find $y$

• Maximal gain attack (MGA)
  • Find $Y$ by solving $\max_Y G(Y)$
Maximal Gain Attack (MGA)

- Assume \( n \) genuine users and \( m \) fake users
- We can calculate \( G \) as follows

\[
G = \frac{\sum_{i=n+1}^{n+m} \sum_{t \in T} 1_{S(y_i)}(t)}{(n + m)(p + q)} - \frac{m(f_T(p - q) + rq)}{(n + m)(p - q)}
\]

- \( f_T \): sum of true frequencies over the target items among genuine users
- \( r = |T| \): number of target items

Independent of \( Y \), denoted by \( c \)
Maximal Gain Attack (MGA)

• Optimal solution for $Y$

$$Y^* = \arg \max_Y \sum_{i=n+1}^{n+m} \sum_{t \in T} 1_S(y_i)(t)$$

• For each fake user, we craft its perturbed value via solving

$$y^* = \arg \max_{y \in D} \sum_{t \in T} 1_S(y)(t)$$

Maximize the number of target items that $y^*$ supports
MGA-kRR

• $\sum_{t \in T} 1_{S(y)}(t) \leq 1$

• $\sum_{t \in T} 1_{S(y)}(t) = 1$ when $y \in T$

• MGA-kRR selects any target item as $y$

• $G = \frac{m}{(n+m)(p+q)} - c$
MGA-OUE

\[ \sum_{t \in T} 1_{S(y)}(t) \leq r \]
\[ \sum_{t \in T} 1_{S(y)}(t) = r \text{ when } \forall t \in T, \ y_t = 1 \]

\[ G = \frac{rm}{(n+m)(p+q)} - c \]

- MGA-OUE sets \( t \)-th bit in vector \( y \) as 1 for all \( t \in T \)
  - Randomly sets other bits such that number of 1’s seems normal

\[ T = \{2,4\} \]

| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
---|---|---|---|---|---|---|---|---|---|

MGA-OLH

\[ \sum_{t \in T} 1_{S(y)}(t) \leq r \]

\[ \sum_{t \in T} 1_{S(y)}(t) = r \text{ when } \forall t \in T, \ H(t) = a \]

MGA-OLH searches for \( H \) and \( a \) such that \( \forall t \in T, \ H(t) = a \)

\[ G = \frac{rm}{(n+m)(p+q)} - c \]
Overall Gains of Attacks

\[ \beta = \frac{m}{n+m} : \text{fraction of fake users among all users} \]
\[ d: \text{number of items} \]
\[ r = |T|: \text{number of target items} \]
\[ f_T: \text{sum of true frequencies of the target items} \]
\[ \varepsilon: \text{privacy budget} \]

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>kRR</th>
<th>OUE</th>
<th>OLH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random perturbed-value attack (RPA)</td>
<td>(\beta\left(\frac{r}{d} - f_T\right))</td>
<td>(\beta(r - f_T))</td>
<td>(-\beta f_T)</td>
</tr>
<tr>
<td>Random item attack (RIA)</td>
<td>(\beta(1 - f_T))</td>
<td>(\beta(1 - f_T))</td>
<td>(\beta(1 - f_T))</td>
</tr>
<tr>
<td>Maximal gain attack (MGA)</td>
<td>(\beta(1 - f_T) + \frac{\beta(d-r)}{e^\varepsilon - 1})</td>
<td>(\beta(2r - f_T) + \frac{2\beta r}{e^\varepsilon - 1})</td>
<td>(\beta(2r - f_T) + \frac{2\beta r}{e^\varepsilon - 1})</td>
</tr>
<tr>
<td>Standard deviation of estimation</td>
<td>(\frac{r\sqrt{d^2 - 2d + e^\varepsilon}}{(e^\varepsilon - 1)\sqrt{n}})</td>
<td>(\frac{2re^{\varepsilon/2}}{(e^\varepsilon - 1)\sqrt{n}})</td>
<td>(\frac{2re^{\varepsilon/2}}{(e^\varepsilon - 1)\sqrt{n}})</td>
</tr>
</tbody>
</table>
## Takeaways

<table>
<thead>
<tr>
<th>Attack</th>
<th>kRR</th>
<th>OUE</th>
<th>OLH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random perturbed-value attack (RPA)</td>
<td>$\beta\left(\frac{r}{d} - f_T\right)$</td>
<td>$\beta(r - f_T)$</td>
<td>$-\beta f_T$</td>
</tr>
<tr>
<td>Random item attack (RIA)</td>
<td>$\beta(1 - f_T)$</td>
<td>$\beta(1 - f_T)$</td>
<td>$\beta(1 - f_T)$</td>
</tr>
<tr>
<td>Maximal gain attack (MGA)</td>
<td>$\beta(1 - f_T) + \frac{\beta(d-r)}{e^{\beta} - 1}$</td>
<td>$\beta(2r - f_T) + \frac{2\beta r}{e^{\beta} - 1}$</td>
<td>$\beta(2r - f_T) + \frac{2\beta r}{e^{\beta} - 1}$</td>
</tr>
<tr>
<td>Standard deviation of estimation</td>
<td>$\frac{r\sqrt{d-2} + e^{\beta}}{e^\beta - 1 \sqrt{n}}$</td>
<td>$\frac{2re^{\beta/2}}{(e^\beta - 1)\sqrt{n}}$</td>
<td>$\frac{2re^{\beta/2}}{(e^\beta - 1)\sqrt{n}}$</td>
</tr>
</tbody>
</table>

**MGA is the most powerful attack**

The overall gain of MGA is much larger than the standard deviation of estimation for the protocols.
## Takeaways

<table>
<thead>
<tr>
<th></th>
<th>kRR</th>
<th>OUE</th>
<th>OLH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random perturbed-value attack (RPA)</td>
<td>$\beta \left( \frac{r}{d} - f_T \right)$</td>
<td>$\beta \left( r - f_T \right)$</td>
<td>$-\beta f_T$</td>
</tr>
<tr>
<td>Random item attack (RIA)</td>
<td>$\beta \left( 1 - f_T \right)$</td>
<td>$\beta \left( 1 - f_T \right)$</td>
<td>$\beta \left( 1 - f_T \right)$</td>
</tr>
<tr>
<td>Maximal gain attack (MGA)</td>
<td>$\beta \left( 1 - f_T \right) + \frac{\beta \left( d - r \right)}{e^{e-1}}$</td>
<td>$\beta \left( 2r - f_T \right) + \frac{2\beta r}{e^{e-1}}$</td>
<td>$\beta \left( 2r - f_T \right) + \frac{2\beta r}{e^{e-1}}$</td>
</tr>
<tr>
<td>Standard deviation of estimation</td>
<td>$\frac{r \sqrt{d-2} + e^{e}}{\left(e^{e-1}\right)^{1/2}}$</td>
<td>$\frac{2r e^{e/2}}{\left(e^{e-1}\right)^{1/2}}$</td>
<td>$\frac{2r e^{e/2}}{\left(e^{e-1}\right)^{1/2}}$</td>
</tr>
</tbody>
</table>

MGA achieves same overall gain for OUE and OLH

When the item domain $d$ is large, OUE and OLH are more secure
## Takeaways

There is a security-privacy tradeoff for the pure LDP protocols.

Smaller $\varepsilon \rightarrow$ stronger privacy & weaker security.
Attack Heavy Hitter Identification

- Heavy hitter identification uses frequency estimation oracles
- We can use frequency estimation attacks
- For PEM, we perform MGA-OLH in each group
Evaluation Results

Fire dataset, $r = 10$, $\beta = 0.05$, $\varepsilon = 1$

MGA can promote 9 out of 10 random-selected target items to be top-10, and all 10 target items to be top-15 heavy hitters.
Countermeasures

• Normalization

• Detecting fake users

• Conditional probability based detection
Detecting Fake Users

| User 1: | 0 | 1 | 0 | 1 | 1 | 1 |
| User 2: | 1 | 1 | 0 | 1 | 1 | 0 |
| User 3: | 0 | 0 | 1 | 0 | 0 | 1 |
| User 4: | 0 | 1 | 1 | 1 | 1 | 0 |

MGA maximizes the gain by having $y$ support all target items.

Common pattern in $y$ of fake users.

Detect via frequent itemset mining.
Detecting and Removing Fake Users

IPUMS dataset, $\varepsilon = 1$, OUE
Adaptive Attacks (MGA-A)

- Attacker can evade the detection via randomly selecting $r'$ target items to support in each $y$
Conclusions

• We propose data poisoning attacks to LDP that can effectively promote target items

• We show the security-privacy trade-off in LDP protocols

• Advanced defenses are needed to defend against our attacks
Thanks