Data Poisoning Attacks to Local Differential Privacy Protocols

Xiaoyu Cao, Jinyuan Jia, Neil Gong
Data Statistics Collection

Trusted Data

What is the most popular item this week?

Answer
Local Differential Privacy (LDP)

Untrusted

Noise

People:
- Black woman
- Red-haired woman
- Man with black hair
- Man with red hair

Diagram shows people contributing data, which is added to noise before being sent to an untrusted cloud for processing.
Data Poisoning Attacks

Frequency Estimation
Heavy Hitter Identification

Noise
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
  • $\forall v_1, \forall v_2, \forall y$, $\Pr(\text{PE}(v_1) = y) \leq e^\varepsilon \Pr(\text{PE}(v_2) = y)$
  • PE: perturbed encoded
Pure LDP

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

- $S(y)$ is the set of items that $y$ supports
  - $y$ supports $v$: $y$ “votes for” $v$ in **Aggregate** step
  - Protocol-dependent

- **Aggregate** as follows
  - $\tilde{f}_v = \frac{\frac{1}{n} \sum_{i=1}^{n} I_{S(y_i)}(v) - q}{p - q}$, $I_{S(y_i)}(v)$: 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**
  - \( S(y) = \{v | y_v = 1\} \)

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

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

E.g., \( d = 8, v = 4 \)
OLH

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

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

• **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

Prices through the course of our analysis range from $0.01 to $0.20 per Twitter account, with a median cost of $0.04 for all merchants. Despite the large overall span, Yahoo accounts, like Hotmail, are widely available, with prices ranging from $0.006 – 0.015 per account.
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
Maximal Gain Attack (MGA)

• Optimal solution for \( \mathbf{Y} \)

\[
\mathbf{Y}^* = \arg \max_{\mathbf{Y}} \sum_{i=n+1}^{n+m} \sum_{t \in T} 1_{S(y_i)}(t)
\]

• 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

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

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

• $\sum_{t \in T} \mathbb{1}_{S(y)}(t) \leq r$

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

| Attack Type                              | kRR                     | OUE                     | OLH                     |
|------------------------------------------|-------------------------|-------------------------|-------------------------|
| Random perturbed-value attack (RPA)      | $\beta \left( \frac{r}{d} - f_T \right)$ | $\beta (r - f_T)$      | $-\beta f_T$           |
| Random item attack (RIA)                 | $\beta (1 - f_T)$      | $\beta (1 - f_T)$      | $\beta (1 - f_T)$      |
| Maximal gain attack (MGA)                | $\beta (1 - f_T) + \frac{\beta (d-r)}{e^\varepsilon - 1}$ | $\beta (2r - f_T) + \frac{2\beta r}{e^\varepsilon - 1}$ | $\beta (2r - f_T) + \frac{2\beta r}{e^\varepsilon - 1}$ |
| Standard deviation of estimation         | $\frac{r \sqrt{d - 2 + e^\varepsilon}}{(e^\varepsilon - 1) \sqrt{n}}$ | $\frac{2r e^{\varepsilon/2}}{(e^\varepsilon - 1) \sqrt{n}}$ | $\frac{2r e^{\varepsilon/2}}{(e^\varepsilon - 1) \sqrt{n}}$ |

\[
\beta = \frac{m}{n+m} : \text{ fraction of fake users among all users}
\]

\[
d: \text{ number of items}
\]

\[
r = |T|: \text{ number of target items}
\]

\[
f_T: \text{ sum of true frequencies of the target items}
\]

\[
\varepsilon: \text{ privacy budget}
\]
### Takeaways

|                          | kRR                                      | OUE                                      | OLH                                     |
|--------------------------|------------------------------------------|------------------------------------------|-----------------------------------------|
| Random perturbed-value attack (RPA) | $\beta\left(\frac{r}{d} - f_T\right)$ | $\beta(r - f_T)$                          | $-\beta f_T$                           |
| Random item attack (RIA)  | $\beta(1 - f_T)$                        | $\beta(1 - f_T)$                        | $\beta(1 - f_T)$                       |
| Maximal gain attack (MGA) | $\beta(1 - f_T) + \frac{\beta(d-r)}{e^k-1}$ | $\beta(2r - f_T) + \frac{2\beta r}{e^k-1}$ | $\beta(2r - f_T) + \frac{2\beta r}{e^k-1}$ |
| Standard deviation of estimation | $\frac{r\sqrt{d-2+e^k}}{(e^k-1)\sqrt{n}}$ | $\frac{2re^{k/2}}{(e^k-1)\sqrt{n}}$ | $\frac{2re^{k/2}}{(e^k-1)\sqrt{n}}$ |

MGA is the most powerful attack

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

|                              | kRR                              | OUE                | OLH                |
|------------------------------|----------------------------------|--------------------|--------------------|
| Random perturbed-value attack (RPA) | $\beta \left( \frac{x}{d} - f_T \right)$ | $\beta (r - f_T)$ | $-\beta f_T$       |
| Random item attack (RIA)      | $\beta (1 - f_T)$                | $\beta (1 - f_T)$ | $\beta (1 - f_T)$  |
| Maximal gain attack (MGA)     | $\beta (1 - f_T) + \frac{\beta (d-r)}{e^\epsilon - 1}$ | $\beta (2r - f_T) + \frac{2\beta r}{e^\epsilon - 1}$ | $\beta (2r - f_T) + \frac{2\beta r}{e^\epsilon - 1}$ |
| Standard deviation of estimation | $\frac{r \sqrt{d - 2 + e^\epsilon}}{(e^\epsilon - 1) \sqrt{n}}$ | $\frac{2re^{\epsilon/2}}{(e^\epsilon - 1) \sqrt{n}}$ | $\frac{2re^{\epsilon/2}}{(e^\epsilon - 1) \sqrt{n}}$ |

MGA achieves same overall gain for OUE and OLH

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

|                          | kRR                                | OUE                                | OLH                                |
|--------------------------|------------------------------------|------------------------------------|------------------------------------|
| Random perturbed-value attack (RPA) | $\beta \left(\frac{r}{d} - f_T\right)$ | $\beta (r - f_T)$                  | $-\beta f_T$                       |
| Random item attack (RIA)  | $\beta (1 - f_T)$                  | $\beta (1 - f_T)$                  | $\beta (1 - f_T)$                  |
| Maximal gain attack (MGA) | $\beta (1 - f_T) + \frac{\beta}{d^2 - (\epsilon - 1)}$ | $\beta (2r - f_T) + \frac{2\beta r}{d^2 - (\epsilon - 1)}$ | $\beta (2r - f_T) + \frac{2\beta r}{d^2 - (\epsilon - 1)}$ |
| Standard deviation of estimation | $\frac{r\sqrt{d - 2} + \epsilon^2}{(\epsilon - 1)\sqrt{n}}$ | $\frac{2r\epsilon^{1/2}}{(\epsilon^2 - 1)\sqrt{n}}$ | $\frac{2r\epsilon^{1/2}}{(\epsilon^2 - 1)\sqrt{n}}$ |

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

Smaller $\epsilon \rightarrow$ 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

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