Locally Differentially Private Frequency Estimation Exploiting Consistency

Tianhao Wang
Purdue University

Joint work with Milan Lopuhaä-Zwakenberg,
Zitao Li, Boris Skoric, Ninghui Li
Privacy in Practice

• Local differential privacy is deployed
  • In Google Chrome browser, to collect browsing statistics
  • In Apple iOS and MacOS, to collect typing statistics
  • In Microsoft Windows, to collect telemetry data over time
  • In Alibaba, we built a system to collect user transaction info

• Different algorithms are proposed.
• They work for different tasks and different settings.
• They are all based on Randomized Response.
Randomized Response

• Survey technique for private questions
  
• Survey people:
  • “Do you have disease X?”
  
• Each person:
  • Flip a secret coin
  • Answer truth if head (w.p. 0.5)
  • Answer randomly if tail (w.p. 0.5):
    • reply “yes”/“no” w.p. 0.5

Pr[disease → yes] = Pr[disease → yes ∧ head] + Pr[disease → yes ∧ tail] = 0.5×1 + 0.5×0.5 = 0.75

Similarly:
Pr[disease → no] = 0.25
Pr[no disease → yes] = 0.25
Pr[no disease → no] = 0.75

Randomized Response

• To estimate the distribution:

• If \( n_{\text{yes}} \) out of \( n \) people have the disease, we expect to see:

\[
E[\text{yes}] = 0.75 + 0.25(n - n_{\text{yes}})
\]

• Inverting the above equation:

\[
\frac{n_{\text{yes}}}{n} = \frac{\text{yes} - 0.25}{0.5}
\]

• It is the unbiased estimation of the number of patients

\[
E[ n_{\text{yes}} ] = \frac{n_{\text{yes}}}{0.5} = n_{\text{yes}}
\]

• Similar for the “no”
Local Differential Privacy (LDP)

- Estimation function is done independent for each value $v$.
- The result is not consistent.
  - Some may be negative.
  - Sum may not be $n$ (the original number of users).

- In this work, we explore 10 different methods that improves the accuracy of LDP by enforcing consistency.

$$y = A(v)$$

takes input value $v$ and outputs $y$.

For any $v$ and $v'$, and any valid output $y$,
$$\frac{\Pr[A(v)=y]}{\Pr[A(v')=y]} \leq e^\varepsilon$$

Takes reports from all users and outputs $y$.
Making Estimations Consistent

1) The estimated frequency of each value is non-negative.
2) The sum of the estimated frequencies is 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Non-neg</th>
<th>Sum to 1</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td>Use existing estimation</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Baseline</strong></td>
<td>Convert negative est. to 0</td>
<td>Yes</td>
<td>No</td>
<td>O(d)</td>
</tr>
<tr>
<td><strong>Baseline</strong></td>
<td>Convert negative query result to 0</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Baseline</strong></td>
<td>Convert est. below threshold T to 0</td>
<td>Yes</td>
<td>No</td>
<td>O(d)</td>
</tr>
<tr>
<td><strong>Norm</strong></td>
<td>Add δ to est.</td>
<td>No</td>
<td>Yes</td>
<td>O(d)</td>
</tr>
<tr>
<td><strong>Norm-Mul</strong></td>
<td>Convert negative est. to 0, then multiply ϒ to positive est.</td>
<td>Yes</td>
<td>Yes</td>
<td>O(d)</td>
</tr>
<tr>
<td><strong>Norm-Cut</strong></td>
<td>Convert negative and small positive est. below ϑ to 0</td>
<td>Yes</td>
<td>Almost</td>
<td>O(d)</td>
</tr>
<tr>
<td><strong>Norm-Sub</strong></td>
<td>Convert negative est. to 0 while adding δ to positive est.</td>
<td>Yes</td>
<td>Yes</td>
<td>O(d)</td>
</tr>
<tr>
<td><strong>MLE-Apx</strong></td>
<td>Convert negative est. to 0, then add δ to positive est.</td>
<td>Yes</td>
<td>Yes</td>
<td>O(d)</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>Fit Power-Law dist., then minimize expected squared error.</td>
<td>Yes</td>
<td>No</td>
<td>O(√n+d)</td>
</tr>
<tr>
<td><strong>PowerNS</strong></td>
<td>Apply Norm-Sub after Power</td>
<td>Yes</td>
<td>Yes</td>
<td>O(√n+d)</td>
</tr>
</tbody>
</table>
Post-Processing: Toy Example

Truth

- Base-Pos: Convert negative to 0

Constraint 1: estimation is non-negative

Constraint 2: Sum of estimations is known

Estimated

Sum: 106%

Norm-Sub: Additively normalize the result

It is the solution to Constraint Least Square (CLS) and Approximate Maximal Likelihood Estimation (MLE)
Analysis of the Estimation in LDP

• Estimation function
  - \( \hat{n}_{\text{yes}} = \frac{l_{\text{yes}} - 0.25n}{0.5} \), more generally \( \hat{n}_v = \frac{l_v - qn}{p-q} \)

  - Probability of \( A(v) \) supporting \( v \) (disease → yes)
  - Probability of \( A(v') \) supporting \( v \) where \( v' \neq v \) (no disease → yes)

  - Takeaway: The noise of the LDP estimation approximately follows Gaussian distribution.

• Noise comes from Binomials
  - \( \text{Bin}(nv, p) + \text{Bin}(n - nv, q) = \text{Bin} \left( n, \frac{v}{n}p + \frac{n-v}{n}q \right) \)

• When \( n \) is large, noise \( \approx N(p'n, \sqrt{np'(1-p')}}) \) for \( p' = \frac{n_v}{n}p + \frac{n-nv}{n}q \)

Empirical Understanding

• 1 million reports following Zipf’s distribution (s=1.5) with 1024 values.
• 5000 runs (each dot is the mean).

**Estimated Norm**
- Sub: Additively normalize the result
- Base
- Pos: Convert negative to 0

Systematic positive bias to infrequent values.
Systematic negative bias to frequent values.

Bias is a bad thing. Should we stop post-processing?
No, because it prevents impossible events.
But how is it affect the utility?
Empirical Understanding

• 1 million reports following Zipf’s distribution (s=1.5) with 1024 values.
• 5000 runs (each dots represent a run).

Estimated Norm
- Sub: Additively normalize the result
- Base-Pos: Convert negative to 0

Variance is smaller for infrequent values.

Takeaway Message
• Utility is composed of bias and variance
• Post processing introduces bias but reduces variance
• Different method achieves different bias-variance tradeoff
Comparison of Different Methods

- Multiplicatively normalize the result

- Norm-Sub > Base-Pos > Base > Norm-Mul

- Exploiting constraint may or may not be helpful

More Privacy
Comparison of Different Methods

- Normalization-based methods work better.
- MSE is symmetric with $\rho = 50$ if the estimates sum up to 1.

• Uniformly sample $\rho$% elements from the domain.
• MSE of estimating a subset of values (set-value).
Summary

- LDP noise follows Gaussian.
- Norm-Sub is the solution to MLE.
- Exploiting priors is helpful.
- Different method works for different tasks.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>Use existing estimation</td>
</tr>
<tr>
<td>Base-Pos</td>
<td>Convert negative est. to 0</td>
</tr>
<tr>
<td>Post-Pos</td>
<td>Convert negative query result to 0</td>
</tr>
<tr>
<td>Base-Cut</td>
<td>Convert est. below threshold $T$ to 0</td>
</tr>
<tr>
<td>Norm</td>
<td>Add $\delta$ to est.</td>
</tr>
<tr>
<td>Norm-Mul</td>
<td>Convert negative est. to 0, then multiply $\Upsilon$ to positive est.</td>
</tr>
<tr>
<td>Norm-Cut</td>
<td>Convert negative and small positive est. below $\vartheta$ to 0</td>
</tr>
<tr>
<td>Norm-Sub</td>
<td>Convert negative est. to 0 while adding $\delta$ to positive est.</td>
</tr>
<tr>
<td>MLE-Apx</td>
<td>Convert negative est. to 0, then add $\delta$ to positive est.</td>
</tr>
<tr>
<td>Power</td>
<td>Fit Power-Law dist., then minimize expected squared error.</td>
</tr>
<tr>
<td>PowerNS</td>
<td>Apply Norm-Sub after Power</td>
</tr>
</tbody>
</table>