Local Differential Privacy: Solution or Distraction?

Graham Cormode

g.cormode@warwick.ac.uk

Tejas Kulkarni (Warwick)

Divesh Srivastava (AT&T)
Local Differential Privacy

- Local Differential Privacy: ensure that every user’s output is DP
  - Aka (private) “Federated analytics”
Local Differential Privacy

Local Differential Privacy: ensure that every user’s output is DP
  - Aka (private) “Federated analytics”

LDP mostly built on variations of randomized response (RR)
  - With probability $p > \frac{1}{2}$, report the true (binary) answer
  - With probability $1-p$, lie
Local Differential Privacy

- Local Differential Privacy: ensure that every user’s output is DP
  - Aka (private) “Federated analytics”
- LDP mostly built on variations of randomized response (RR)
  - With probability $p > \frac{1}{2}$, report the true (binary) answer
  - With probability $1-p$, lie
- Now popular for gathering private frequency statistics at scale
  - RAPPOR in Chrome, combining RR with Bloom filters
  - In Apple iOS and MacOS, combining RR with sketches and transforms
  - This yields deployments of over 100 million users
Local Differential Privacy

- Local Differential Privacy: ensure that every user’s output is DP
  - Aka (private) “Federated analytics”
- LDP mostly built on variations of randomized response (RR)
  - With probability $p > \frac{1}{2}$, report the true (binary) answer
  - With probability $1-p$, lie
- Now popular for gathering private frequency statistics at scale
  - RAPPOR in Chrome, combining RR with Bloom filters
  - In Apple iOS and MacOS, combining RR with sketches and transforms
  - This yields deployments of over 100 million users
- Local Differential privacy widely deployed since 2015:
  Randomized response invented in 1965: five decade lead time!
Going beyond 1 bit of data

1 bit can tell you a lot, but can we do more?

- **Recent work**: materializing marginal distributions
  - Each user has \( d \) bits of data (encoding sensitive data)
  - We are interested in the distribution of combinations of attributes
Going beyond 1 bit of data

1 bit can tell you a lot, but can we do more?

- **Recent work**: materializing marginal distributions
  - Each user has $d$ bits of data (encoding sensitive data)
  - We are interested in the distribution of combinations of attributes

<table>
<thead>
<tr>
<th></th>
<th>Gender</th>
<th>Obese</th>
<th>High BP</th>
<th>Smoke</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zayn</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Going beyond 1 bit of data

1 bit can tell you a lot, but can we do more?

♦ Recent work: materializing marginal distributions
  – Each user has $d$ bits of data (encoding sensitive data)
  – We are interested in the distribution of combinations of attributes

<table>
<thead>
<tr>
<th></th>
<th>Gender</th>
<th>Obese</th>
<th>High BP</th>
<th>Smoke</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zayn</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gender/Obese</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.28</td>
<td>0.22</td>
</tr>
<tr>
<td>1</td>
<td>0.29</td>
<td>0.21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Disease/Smoke</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.55</td>
<td>0.15</td>
</tr>
<tr>
<td>1</td>
<td>0.10</td>
<td>0.20</td>
</tr>
</tbody>
</table>
Hadamard transform

Instead of materializing projections of data, we can transform it

- Via Hadamard transform (the discrete Fourier transform for the binary hypercube)
  
  
  - Simple and fast to apply

\[
\begin{bmatrix}
H^* & H^*
\end{bmatrix} = 
\begin{bmatrix}
-1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\
1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & 1
\end{bmatrix}
\]
Hadamard transform

Instead of materializing projections of data, we can transform it

- Via Hadamard transform (the discrete Fourier transform for the binary hypercube)
  - Simple and fast to apply
    \[
    \begin{bmatrix}
    H^* & H^* \\
    H^* & -H^*
    \end{bmatrix} = \begin{bmatrix}
    -1 & 1 & 1 & 1 \\
    1 & -1 & 1 & 1 \\
    1 & 1 & -1 & 1 \\
    1 & 1 & 1 & -1 \\
    -1 & 1 & 1 & 1 \\
    1 & -1 & 1 & 1 \\
    1 & 1 & -1 & 1 \\
    1 & 1 & 1 & -1 \\
    \end{bmatrix}.
    \]

- Property 1: only $\binom{d}{k}$ coefficients are needed to build any $k$-way marginal
  - Reduces the amount of information to release
Hadamard transform

Instead of materializing projections of data, we can transform it

- **Via Hadamard transform** (the discrete Fourier transform for the binary hypercube)
  - Simple and fast to apply
  
  \[
  \begin{bmatrix}
  H^* & H^*
  \end{bmatrix}
  =
  \begin{bmatrix}
  -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\
  1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\
  1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\
  1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 \\
  -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\
  1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\
  1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
  1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
  \end{bmatrix}
  \]

- **Property 1**: only \((d \text{ choose } k)\) coefficients are needed to build any k-way marginal
  - Reduces the amount of information to release

- **Property 2**: Hadamard transform is a linear transform
  - Can estimate global coefficients by sampling and averaging
Hadamard transform

Instead of materializing projections of data, we can transform it

- **Via Hadamard transform** (the discrete Fourier transform for the binary hypercube)
  - Simple and fast to apply
  - **Property 1**: only \((d \text{ choose } k)\) coefficients are needed to build any \(k\)-way marginal
  - Reduces the amount of information to release

- **Property 2**: Hadamard transform is a linear transform
  - Can estimate global coefficients by sampling and averaging
  - Yields error proportional to \(2^{k/2}d^{k/2}/\sqrt{N}\)
  - Better than simply materializing marginals (in theory)
Empirical behaviour [C, Kulkarni, Srivastava SIGMOD 18]

- Compare three methods: Hadamard based (Inp_HT), marginal materialization (Marg_PS), Expectation maximization (Inp_EM)
- Measure sum of absolute error in materializing 2-way marginals
- \( N = 0.5M \) individuals, vary privacy parameter \( \varepsilon \) from 0.4 to 1.4
Application – building a Bayesian model

- **Aim:** build the tree with highest mutual information (MI)
- **Plot** shows MI on the ground truth data for evaluation purposes
Range Queries

- Given data from an ordered domain, we study range queries:
  - “How many data points fall in the range \([l, r]\)”? 
Given data from an ordered domain, we study range queries:

- “How many data points fall in the range \([l, r]\)”? 

Hierarchical approaches improve over summing point queries:

a) Impose a regular tree over the input domain, and sample nodes
   - Need to do post-processing to obtain consistent answers
b) Apply a Haar wavelet transform to input, and sample coefficients
Range Queries

- Given data from an ordered domain, we study range queries:
  - “How many data points fall in the range \([l, r]\)”? 
- Hierarchical approaches improve over summing point queries:
  a) Impose a regular tree over the input domain, and sample nodes
     ■ Need to do post-processing to obtain consistent answers 
  b) Apply a Haar wavelet transform to input, and sample coefficients 
- Which method is best? **Answer:** both are competitive!
  - Similar variance (up to leading constant) for optimal settings 
  - Similar empirical performance, slight preferences for different \(\varepsilon\) 
  - In contrast to the centralized case, where trees are preferred
Quantile queries [C, Kulkarni, Srivastava VLDB19]

- Use range queries to find ranges that cover a given fraction
  - E.g. the median is the 0.5 quantile query
Quantile queries [C, Kulkarni, Srivastava VLDB19]

♦ Use range queries to find ranges that cover a given fraction
  – E.g. the median is the 0.5 quantile query
♦ Both Hierarchical Histograms (HH) and Haar wavelets obtain similar results: very accurate answers for $N$ large enough
LDP as a solution

延期

For LDP to really work with good accuracy we need to have:
- Massive number of participating users (ideally millions)
- Relaxed privacy parameters ($\varepsilon = 8–16$ in Apple deployment)
- “Flexible” attitude to composition results (daily “reset”)
- Relatively simple analytics target (simple statistics)
LDP as a solution

- For LDP to really work with good accuracy we need to have:
  - Massive number of participating users (ideally millions)
  - Relaxed privacy parameters ($\varepsilon = 8–16$ in Apple deployment)
  - "Flexible" attitude to composition results (daily "reset")
  - Relatively simple analytics target (simple statistics)

- LDP is really good for:
  - Large deployments by well-resourced tech companies
  - Academic research generating new papers in popular model
LDP as a solution

♦ For LDP to really work with good accuracy we need to have:
  – Massive number of participating users (ideally millions)
  – Relaxed privacy parameters ($\varepsilon = 8–16$ in Apple deployment)
  – “Flexible” attitude to composition results (daily “reset”)
  – Relatively simple analytics target (simple statistics)

♦ LDP is really good for:
  – Large deployments by well-resourced tech companies
  – Academic research generating new papers in popular model

♦ LDP does not seem so good for:
  – Everyone else?
LDP as a solution

- For LDP to really work with good accuracy we need to have:
  - Massive number of participating users (ideally millions)
  - Relaxed privacy parameters ($\varepsilon = 8–16$ in Apple deployment)
  - “Flexible” attitude to composition results (daily “reset”)
  - Relatively simple analytics target (simple statistics)

- LDP is really good for:
  - Large deployments by well-resourced tech companies
  - Academic research generating new papers in popular model

- LDP does not seem so good for:
  - Everyone else?

- RAPPOR has been replaced in current Chrome versions
So is LDP a distraction in federated learning?

LDP in isolation does not provide a rounded solution, but:

- LDP plus deidentification of reports gives stronger privacy
  - “Shuffling” the messages gives $O(\epsilon/\sqrt{n})$ (centralized) DP
  - Generic bounds for sufficiently restricted LDP protocols
  - Tight bounds for core problems (e.g. sums and counts)
  - Many recent results [Bitau et al 2017] [Erlingsson et al. 2019] [Balle et al 2019] [Cheu et al 2019] ...

10
So is LDP a distraction in federated learning?

LDP in isolation does not provide a rounded solution, but:

- LDP plus deidentification of reports gives stronger privacy
  - “Shuffling” the messages gives $O(\frac{\varepsilon}{\sqrt{n}})$ (centralized) DP
  - Generic bounds for sufficiently restricted LDP protocols
  - Tight bounds for core problems (e.g. sums and counts)
  - Many recent results [Bitau et al 2017] [Erlingsson et al. 2019] [Balle et al 2019] [Cheu et al 2019] ...

- LDP protocols are good candidates for implementing with SMC
  - Simple partitions of quantities, small data per participant
  - One algorithm could “compile” to multiple target models?
So is LDP a distraction in federated learning?

LDP in isolation does not provide a rounded solution, but:

- LDP plus deidentification of reports gives stronger privacy
  - “Shuffling” the messages gives $O(\epsilon/\sqrt{n})$ (centralized) DP
  - Generic bounds for sufficiently restricted LDP protocols
  - Tight bounds for core problems (e.g. sums and counts)
  - Many recent results [Bitau et al 2017] [Erlingsson et al. 2019] [Balle et al 2019] [Cheu et al 2019] ...

- LDP protocols are good candidates for implementing with SMC
  - Simple partitions of quantities, small data per participant
  - One algorithm could “compile” to multiple target models?

- LDP may be a stepping stone to more powerful PETS