Problem formulation

Given $x \in \mathbb{R}^d$ with sparsity $\|x\|_0 \leq k$ and $\|x\|_\infty \leq u$, apply a mechanism $\mathcal{A}$ s.t. $\mathcal{A}(x)$ is $\epsilon$-differentially private with respect to pairs of vectors $x, x' \in \mathbb{R}^d$ with $\|x - x'\|_1 \leq 1$

Goals:
- $\mathcal{A}(x)$ stored in small-space data structure, space depending on $k, d, u$, and $n = \|x\|_1$
- Ability to quickly query for a value $x_i$
- Per-query error similar to Lap$(1/\epsilon)$

Main result

We introduce the Approximate Laplace Projection (ALP) private sparse vector representation, with these properties:

<table>
<thead>
<tr>
<th>Space in bits</th>
<th>Access time</th>
<th>Per-query error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(k \log (d+u))$</td>
<td>$O(\log d)$</td>
<td>$O(1) \cdot \text{Lap}(1/\epsilon)$</td>
</tr>
</tbody>
</table>

Expected value of Lap$(1/\epsilon)$ is $O(1/\epsilon)$

Techniques

Initially scale to values $y_i = \epsilon x_i$

**Difficult case:** Small values, $|y_i| = O(\log d)$

**Idea for the case $k = 1$:**
- Randomly round $y_i$ to an integer $y'_i$
- Use unary $O(\log d)$-bit representation of $y'_i$
- Flip each bit with probability $1/3$
- Maximum-likelihood estimator $\hat{y}'_i$ for $y'_i$
- Estimate for $x_i$ is $\hat{y}'_i/\epsilon$

**Extending to $k > 1$:**

Use hashing to randomly choose where to place each bit in unary representation of $y'_i$

Approximate DP

Previously investigated by, e.g., [KKMN09], [BNS16], [BV18], [LKSS18], [CGSS20]. In this setting, it was known how to improve the per-query error bound to $O(\log(1/\delta)/\epsilon)$.

Our mechanism has the following properties:

<table>
<thead>
<tr>
<th>Space in bits</th>
<th>Access time</th>
<th>Per-query error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{O}(k \log (d+u))$</td>
<td>$O(\log (1/\delta))$</td>
<td>$O(1) \cdot \text{Lap}(1/\epsilon)$</td>
</tr>
</tbody>
</table>

Open problem

Is it possible to improve access time while not increasing space and error?
If so, is it possible to achieve these properties?

<table>
<thead>
<tr>
<th>Space in bits</th>
<th>Access time</th>
<th>Per-query error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(k \log (d+u))$</td>
<td>$O(1)$</td>
<td>$O(1) \cdot \text{Lap}(1/\epsilon)$</td>
</tr>
</tbody>
</table>

Known trade-offs*

<table>
<thead>
<tr>
<th>Reference</th>
<th>Space in bits</th>
<th>Access time</th>
<th>Per-query error (expected, worst)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMNS06</td>
<td>$O(d \log u)$</td>
<td>$O(1)$</td>
<td>$O(1/\epsilon)$</td>
</tr>
<tr>
<td>KKMN09**</td>
<td>$O(k \log (d+u))$</td>
<td>$O(1)$</td>
<td>$O(\log (1/\delta)/\epsilon)$</td>
</tr>
<tr>
<td>CPST12</td>
<td>$O(k \log (d+u))$</td>
<td>$O(1)$</td>
<td>$O(\log (d)/\epsilon)$</td>
</tr>
<tr>
<td>BV18</td>
<td>$\tilde{O}(n/\epsilon \log d)$</td>
<td>$\tilde{O}(n/\epsilon)$</td>
<td>$O(1/\epsilon)$</td>
</tr>
</tbody>
</table>

* Results are explicit or follow directly from the references
** Approximate differential privacy

References


Research supported by the VILLUM foundation grant number 16582.