The Geometry of Differential Privacy: the Approximate and Sparse Cases

Aleksandar Nikolov\textsuperscript{1}  Kunal Talwar\textsuperscript{2}  Li Zhang\textsuperscript{2}

Rutgers U.

Microsoft Research, SVC
Outline

1. Intro

2. Dense Case \((n = \Omega(d))\)

3. Sparse Case \((n = o(d))\)
Example

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Zip Code</th>
<th>Smoker</th>
<th>Lung Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>089341</td>
<td>M</td>
<td>07306</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>908734</td>
<td>F</td>
<td>10001</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>560671</td>
<td>M</td>
<td>08541</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

The data is both sensitive (medical information) and personally identifiable (with the right kind of side information).

**Universe**: All possible settings of the attributes

**Histogram**: Number of users for each setting of the attributes.

**Queries**:

- How many male smokers have lung cancer?
- How many more female smokers are there than male smokers?
Setting

- A universe $U$ of user types; $|U| = N$
- A database $D \in U^n$ of $n$ users, each having some type in $U$
- The database in histogram representation:
  - $x \in \mathbb{R}^U$: $x_i$ is the number of users in the database having type $i \in U$
  - $\|x\|_1 = \sum_{i \in U} |x_i| = n$
  - $D \Delta D' \leq 1 \iff \|x - x'\|_1 \leq 1$
Linear Queries

A useful and rich primitive: linear queries on the histogram \( x \).

- **Linear Query**: \( \langle a, x \rangle \)

- **Query Matrix**: \( d \) linear queries: \( Ax \) where \( A \in \mathbb{R}^{d \times N} \)
  - when \( A \) is a 0-1 matrix, we call the \( d \) queries *counting queries*
Privacy

*Privacy Goal:* compute *aggregate* statistics (here: linear queries) without revealing the type of any user, even to an adversary who knows the types of all other users.
Privacy Goal: compute aggregate statistics (here: linear queries) without revealing the type of any user, even to an adversary who knows the types of all other users.

Definition

An algorithm $M$ with input domain $\mathbb{R}^N$ and output range $Y$ is $(\varepsilon, \delta)$-differentially private if for every $n$, every $x, x'$ with $\|x - x'\|_1 \leq 1$, and every measurable $S \subseteq Y$, $M$ satisfies

$$\Pr[M(x) \in S] \leq e^\varepsilon \Pr[M(x') \in S] + \delta.$$
Privacy

**Privacy Goal:** compute *aggregate* statistics (here: linear queries) without revealing the type of any user, even to an adversary who knows the types of all other users.

**Definition**

An algorithm $\mathcal{M}$ with input domain $\mathbb{R}^N$ and output range $Y$ is $(\varepsilon, \delta)$-differentially private if for every $n$, every $x, x'$ with $\|x - x'\|_1 \leq 1$, and every measurable $S \subseteq Y$, $\mathcal{M}$ satisfies

$$\Pr[\mathcal{M}(x) \in S] \leq e^\varepsilon \Pr[\mathcal{M}(x') \in S] + \delta.$$ 

**Intuition:** Algorithm does almost the same, no matter if a particular user participated or not. *Incentive to participate in a study.*
Accuracy

Accuracy of algorithm \( \mathcal{M} \) – *mean squared error*:

\[
\text{Err}(\mathcal{M}, A, n) = \max_{x: \|x\|_1 \leq n} \mathbb{E}_{d} \frac{1}{d} \|\mathcal{M}(A, x, n) - Ax\|_2^2
\]

\[
\text{Err}(\mathcal{M}, A) = \max_n \text{Err}(\mathcal{M}, A, n)
\]
Accuracy

Accuracy of algorithm $\mathcal{M}$ – *mean squared error*:

$$\text{Err}(\mathcal{M}, A, n) = \max_{x: \|x\|_1 \leq n} \mathbb{E}_{d} \frac{1}{d} \| \mathcal{M}(A, x, n) - Ax \|_2^2$$

$$\text{Err}(\mathcal{M}, A) = \max_{n} \text{Err}(\mathcal{M}, A, n)$$

Optimal error on $A$ and on databases of size up to $n$ is:

$$\text{Opt}_{\epsilon, \delta}(A, n) = \min_{\mathcal{M}} \text{Err}(\mathcal{M}, A, n),$$

where the minimum is over all $(\epsilon, \delta)$-differentially private algorithms $\mathcal{M}$. 
Accuracy

Accuracy of algorithm $\mathcal{M}$ – *mean squared error*:

$$\text{Err}(\mathcal{M}, A, n) = \max_{x : \|x\|_1 \leq n} \mathbb{E} \frac{1}{d} \| \mathcal{M}(A, x, n) - Ax \|_2^2$$

$$\text{Err}(\mathcal{M}, A) = \max_n \text{Err}(\mathcal{M}, A, n)$$

Optimal error on $A$ and on databases of size up to $n$ is:

$$\text{Opt}_{\varepsilon, \delta}(A, n) = \min_{\mathcal{M}} \text{Err}(\mathcal{M}, A, n),$$

where the minimum is over all $(\varepsilon, \delta)$-differentially private algorithms $\mathcal{M}$.

The optimum when database size is unrestricted:

$$\text{Opt}_{\varepsilon, \delta}(A) = \max_n \text{Opt}_{\varepsilon, \delta}(A, n)$$
Universal bounds on error

For $A \in [0, 1]^{d \times N}$:

- $\text{Opt}_{\varepsilon, \delta}(A) = O(d)$

  - [DKM$^+06$]: Add $N(0, \sqrt{d}c(\varepsilon, \delta))$ noise to each query answer
  - [DN03]: Tight for random $A$
Universal bounds on error

For $A \in [0, 1]^{d \times N}$:

- $\text{Opt}_{\varepsilon, \delta}(A) = O(d)$
  - [DKM+06]: Add $N(0, \sqrt{d}\epsilon) \delta$ noise to each query answer
  - [DN03]: Tight for random $A$

- $\text{Opt}_{\varepsilon, \delta}(A, n) = O(n \sqrt{\log N})$
  - [HR10, GRU12]: Multiplicative weights, median mechanism
  - [DN03]: Tight up to the $\sqrt{\log N}$ for random $A$
Universal bounds on error

For $A \in [0, 1]^{d \times N}$:

- $\text{Opt}_{\varepsilon, \delta}(A) = O(d)$
  - [DKM$^+$06]: Add $N(0, \sqrt{d}c(\varepsilon, \delta))$ noise to each query answer
  - [DN03]: Tight for random $A$

- $\text{Opt}_{\varepsilon, \delta}(A, n) = O(n\sqrt{\log N})$
  - [HR10, GRU12]: Multiplicative weights, median mechanism
  - [DN03]: Tight up to the $\sqrt{\log N}$ for random $A$

- $\text{Opt}_{\varepsilon, 0}(A, n) = O(n^{4/3} \text{polylog}(N))$
  - [BLR08]: Learning theoretic techniques
  - This work: $\text{Opt}_{\varepsilon, 0}(A, n) = O(n \text{polylog}(N, d))$
Special $A$?

Some matrices $A$ require a lot less error:

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{pmatrix}$$
Special $A$?

Some matrices $A$ require a lot less error:

$$A = \begin{pmatrix}
1 & 0 & \cdots & 0 & 0 \\
1 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
1 & 1 & \cdots & 1 & 0 \\
1 & 1 & \cdots & 1 & 1 \\
\end{pmatrix}$$

- $\text{Opt}_{\epsilon,\delta}(A) = O(\text{polylog}(d))$
  - Algorithm: answer a different set of queries, based on a binary tree data structure
  - Notice: $A$ is TUM
Results

- An algorithm $\mathcal{M}$ is $\alpha$-optimal in the dense case if it is $(\varepsilon, \delta)$-d.p. and
  \[ \text{Err}(\mathcal{M}, A) \leq \alpha \text{Opt}_{\varepsilon,\delta}(A) \]

- An algorithm $\mathcal{M}$ is $\alpha$-optimal in the sparse case if it is $(\varepsilon, \delta)$-d.p. and
  \[ \text{Err}(\mathcal{M}, n) \leq \alpha \text{Opt}_{\varepsilon,\delta}(A, n) \]

<table>
<thead>
<tr>
<th>$\alpha$ =</th>
<th>Unbounded $n$ (Dense)</th>
<th>Bounded $n$ (Sparse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\varepsilon, 0)$-d.p.</td>
<td>$\text{polylog}(d)$ (^1)</td>
<td>?</td>
</tr>
<tr>
<td>$(\varepsilon, \delta)$-d.p.</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Table: Values for $\alpha$

---

\(^1\) [HT10, BDKT12]
Results

- An algorithm $\mathcal{M}$ is $\alpha$-optimal in the dense case if it is $(\epsilon, \delta)$-d.p. and
  \[ \text{Err}(\mathcal{M}, A) \leq \alpha \text{Opt}_{\epsilon, \delta}(A) \]

- An algorithm $\mathcal{M}$ is $\alpha$-optimal in the sparse case if it is $(\epsilon, \delta)$-d.p. and
  \[ \text{Err}(\mathcal{M}, n) \leq \alpha \text{Opt}_{\epsilon, \delta}(A, n) \]

<table>
<thead>
<tr>
<th>$\alpha = \quad$</th>
<th>Unbounded $n$ (Dense)</th>
<th>Bounded $n$ (Sparse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\epsilon, 0)$-d.p.</td>
<td>polylog($d$) $^1$</td>
<td>polylog($d, N$)</td>
</tr>
<tr>
<td>$(\epsilon, \delta)$-d.p.</td>
<td>polylog($d$)</td>
<td>polylog($d, N$)</td>
</tr>
</tbody>
</table>

Table: Values for $\alpha$

$^1$ [HT10, BDKT12]
Growth of Error with $n$

- $\text{Err}(\mathcal{M}_{\text{dense}}, A)$
- $\text{Err}(\mathcal{M}_{\text{sparse}}, A, n)$
- $\text{Opt}(A, n)$
What the algorithms look like?

- **Dense case** \((n = \Omega(d))\)
  - Add correlated Gaussian noise \(w\) and output \(\tilde{y} = Ax + w\)

- **Sparse case** \((n = o(d))\)
  - Compute noisy answers \(\tilde{y}\) using the dense case algorithm
  - Find the closest set of answers \(\hat{y}\) that can be generated by a database \(x\) of size \(\|x\|_1 \leq n\)
Outline

1. Intro

2. Dense Case \((n = \Omega(d))\)

3. Sparse Case \((n = o(d))\)
The Lead Actor: $K$

Let $K = AB_1$ where $B_1$ is the $\ell_1$ ball:

- $nK$ is all query answers that can be generated by a size $n$-database.
- $K = \text{conv}\{\pm a_1, \ldots, \pm a_N\}$
Preliminaries: Gaussian Mechanism

Basic algorithm $\mathcal{M}_{\text{GN}}$:

- Say $K \subseteq rB_2^d$ ($\ell_2$-sensitivity is $r$)
- Output $Ax + w$, where $w \sim N(0, r \cdot c(\varepsilon, \delta))^d$

Properties:

- satisfies $(\varepsilon, \delta)$-differential privacy
- $\text{Err}(\mathcal{M}_{\text{GN}}, A) = O(r^2)$
Preliminaries: Noise Lower Bounds

- [HT10]: $\text{Opt}_{\varepsilon,0}(A) \geq d^2 \text{vol}(K)^{2/d}$

Nikolov, Talwar, Zhang (Rutgers, SVC) Geometry of Privacy p. III
Preliminaries: Noise Lower Bounds

- [HT10]: $\text{Opt}_{\varepsilon,0}(A) \geq d^2 \text{vol}(K)^{2/d}$

- [MN12]: Say $S$ is a simplex of $d$ vertices of $K$ and the origin
  $\Rightarrow \text{Opt}_{\varepsilon,\delta}(A) \geq d^2 \text{vol}(S)^{2/d}$
  - lower bound uses combinatorial discrepancy
Preliminaries: Noise Lower Bounds

- [HT10]: $\text{Opt}_{\varepsilon,0}(A) \geq d^2 \text{vol}(K)^{2/d}$

- [MN12]: Say $S$ is a simplex of $d$ vertices of $K$ and the origin
  $\Rightarrow \text{Opt}_{\varepsilon,\delta}(A) \geq d^2 \text{vol}(S)^{2/d}$
  - lower bound uses combinatorial discrepancy

- $\text{Opt}_{\varepsilon,\delta}(\Pi A) \leq \text{Opt}_{\varepsilon,\delta}(A)$ for any projection $\Pi$ $\Rightarrow$ can use lower bound on any $\Pi A$ to lower bound $\text{Opt}_{\varepsilon,\delta}(A)$. 
**Preliminaries: The Löwner Ellipsoid**

- Every $K$ has a unique minimum volume ellipsoid (MEE) containing it. [Joh48].

- [BT87, Ver01]: If the MEE of $K$ is a ball $rB^d_2$, there are $\Omega(d)$ contact points of $rB^d_2$ and $K$ which are *pairwise nearly orthogonal*. 

\[ K \]
Preliminaries: The Löwner Ellipsoid

- Every $K$ has a unique minimum volume ellipsoid (MEE) containing it. [Joh48].

- [BT87, Ver01]: If the MEE of $K$ is a ball $rB_2^d$, there are $\Omega(d)$ contact points of $rB_2^d$ and $K$ which are pairwise nearly orthogonal.
Preliminaries: The Löwner Ellipsoid

Every $K$ has a unique minimum volume ellipsoid (MEE) containing it. [Joh48].

[BT87, Ver01]: If the MEE of $K$ is a ball $rB_2^d$, there are $\Omega(d)$ contact points of $rB_2^d$ and $K$ which are pairwise nearly orthogonal.
Optimality: pt 1

[BT87, Ver01]: If the MEE of $K$ is a ball $rB_2^d$, there are $\Omega(d)$ contact points of $rB_2^d$ and $K$ which are *pairwise nearly orthogonal*.

- **When the MEE of $K$ is a ball:**
  - Take the simplex $S$ spanned by the nearly orthogonal contact points
  - $d^2 \text{vol}(S)^{2/d} = \Omega(r^2)$
  - *The Gaussian Mechanism* $Ax + N(0, r \cdot c(\varepsilon, \delta))$ is optimal!
Optimality: pt 2

But when the MEE is a “long” ellipse?:

- Find a subspace $\mathcal{V}$ (of dimension $\Omega(d)$) such that $\Pi_{\mathcal{V}} E$ is like a sphere
- Run Gaussian Mechanism on $\Pi_{\mathcal{V}} K$ and recurse on $\mathcal{V}^\perp$
- Can still get a large simplex even inside $\mathcal{V}$ using the full power of [Ver01].
Outline

1. Intro
2. Dense Case \((n = \Omega(d))\)
3. Sparse Case \((n = o(d))\)
Sparse case noise lower bound

- If $S$ is a simplex of $k \leq n$ vertices of $K$ and the origin
  $\Rightarrow \text{Opt}_{\epsilon, \delta}(A, n) \geq \frac{1}{d} k^3 \frac{\text{vol}(S)^2}{k}$

- **Notice:** when the MEE of $K$ is a ball, we found a simplex $S$ which is almost regular
  $\Rightarrow$ any face of $S$ gives a lower bound of $\Omega\left(\frac{n}{d} r^2\right)$

- But what algorithm matches the bound?
Simple Algorithm for Sparse Case

Gaussian Noise + Least Squares Estimation $\mathcal{M}_{\text{GN} + \text{LSE}}$:

1. **Add noise**: Compute $\tilde{y} = Ax + w$ for $w \sim N(0, r \cdot c(\varepsilon, \delta))^d$

2. **Project**: Output $\arg \min \{ \|\hat{y} - \tilde{y}\|_2 : \hat{y} \in nK \}$.
Sparse Case \((n = o(d))\)

Optimality: MEE is a ball

\[
\frac{1}{d} \|\hat{y} - y\|_2^2 \leq \frac{4}{d} \|w\|_2^2.
\]
Optimality: MEE is a ball

\[ \hat{y} = Ax + w \]

\[ \frac{1}{d} \| \hat{y} - y \|_2^2 \leq \frac{4}{d} \| w \|_2^2. \]

\[ \frac{1}{d} \| \hat{y} - y \|_2^2 \leq \frac{2}{d} |\langle w, \hat{y} - y \rangle|. \]
Optimality: MEE is a ball

\[ \frac{1}{d} \| \hat{y} - y \|_2^2 \leq \frac{4}{d} \| w \|_2^2. \]

\[ \frac{1}{d} \| \hat{y} - y \|_2^2 \leq \frac{2}{d} | \langle w, \hat{y} - y \rangle |. \]

\[ \mathbb{E} \frac{2}{d} | \langle w, \hat{y} - y \rangle | \leq \mathbb{E} \frac{4n}{d} \| A^T w \|_\infty \]

\[ = \mathbb{E} \frac{4}{d} | \Pi_w(nK) | \]

\[ \leq \frac{4n}{d} r^2 \sqrt{\log N} \]
**Optimality: MEE is a ball**

\[ \hat{y} = Ax + w \]

- \( \frac{1}{d} \| \hat{y} - y \|_2^2 \leq \frac{4}{d} \| w \|_2^2 \).
- \( \frac{1}{d} \| \hat{y} - y \|_2^2 \leq \frac{2}{d} |\langle w, \hat{y} - y \rangle| \).

\[
E \frac{2}{d} |\langle w, \hat{y} - y \rangle| \leq E \frac{4n}{d} \| A^T w \|_{\infty} \\
= E \frac{4}{d} |\Pi_w(nK)| \\
\leq \frac{4n}{d} r^2 \sqrt{\log N}
\]

*The Gaussian Mechanism + LSE is nearly optimal!*

\[ y = Ax \]

\[ \hat{y} = A\hat{x} \]

\[ \tilde{y} = Ax + w \]
Optimality: General

Same ideas as before:

- Find a subspace $\mathcal{V}$ such that $\Pi_\mathcal{V} E$ is like a sphere
- Run Gaussian Mechanism + LSE on $\Pi_\mathcal{V} K$ and recurse on $\mathcal{V}^\perp$
- Full power of [Ver01] gives a lower bound.
Miscellanea

- $(\varepsilon, 0)$-differential privacy: use generalized $K$-norm noise of [HT10, BDKT12] to “approximate” Gaussian noise.

- In the dense case can extend to worst-case error per query using boosting.

- Our lower bounds are in terms of hereditary discrepancy and our upper bounds are efficiently computable and nearly matching: first polylogarithmic approximation to hereditary discrepancy.
Summary and open questions

- A simple $(\varepsilon, \delta)$-d.p. algorithm for answering linear queries optimally for any workload $A$ and database size $n$.
- Improved on the error bound of [BLR08]
- Polylogarithmic approximation for hereditary discrepancy.

Questions:
- Can an algorithm that processes queries online be competitive?
- Other cases where simple least squares regression provably helps?
- Other data parameters that help reduce error?
Thank you!
Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar.

Avrim Blum, Katrina Ligett, and Aaron Roth.

J. Bourgain and L. Tzafriri.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor.
Sparse Case \((n = o(d))\)

Irit Dinur and Kobbi Nissim.
Revealing information while preserving privacy.

Anupam Gupta, Aaron Roth, and Jonathan Ullman.
Iterative constructions and private data release.

M. Hardt and G. Rothblum.
A multiplicative weights mechanism for privacy-preserving data analysis.

Moritz Hardt and Kunal Talwar.
On the geometry of differential privacy.

F. John.
Extremum problems with inequalities as subsidiary conditions.
Sparse Case ($n = o(d)$)

In *Studies and Essays presented to R. Courant on his 60th Birthday*, pages 187–204, 1948.
