Differential Privacy Surprises: Utility is not (always) monotonic on epsilon

Based on CCS ’23 paper by:
Mario Alvim\textsuperscript{1}, Natasha Fernandes\textsuperscript{2}, Annabelle McIver\textsuperscript{2},
Carroll Morgan\textsuperscript{3} and Gabriel Nunes\textsuperscript{1}

\textsuperscript{1} UFMG, Brazil \hspace{1cm} \textsuperscript{2} Macquarie University, Australia \hspace{1cm} \textsuperscript{3} UNSW, Australia
Differential Privacy

* Add statistical noise to prevent inferences about individuals’ data.
* Noise tuning: \( P(y \mid x) \leq e^\epsilon P(y \mid x') \)
  where x, x’ are datasets differing in one individual.
* Designed to protect against membership inference attacks.

* Also want to preserve useful information (inferences about z given y)
Local Differential Privacy

* Add statistical noise to prevent inferences about individuals’ data.
* Noise tuning: \( P(y \mid x) \leq e^{\epsilon} P(y \mid x') \)
  where \( x, x' \) are different data values of an individual.
* Designed to protect against attribute inference attacks.

* Also want to preserve useful information (observation \( z \) vs true statistic.)
How to tune the noise to optimise the privacy-utility trade-off?

Assumption: Increasing epsilon (decreasing privacy) causes utility to increase — monotonicity

Our result: Utility is not always monotonic on epsilon in general privacy workflows
Quantitative Information Flow

Differential privacy: \( C_{x,y} \leq e^\epsilon C_{x',y} \)

\[ C = \begin{bmatrix} 2/3 & 1/6 & 1/6 \\ 1/3 & 1/3 & 1/3 \\ 1/6 & 1/6 & 2/3 \end{bmatrix} \]

Information flow channel

\(-\log_4\text{-differentially private}\)

Refinement: \( A \sqsubseteq B \) means channel A is more useful than channel B to a Bayesian analyst equipped with any prior and any loss function.

Theorem (Coriaceous): \( A \sqsubseteq B \) iff there exists a channel W s.t. \( A \cdot W = B \)

where \( \cdot \) is matrix multiplication (post-processing)
General Privacy Workflows

How to tune the noise to optimise the privacy-utility trade-off?

OR: If I change epsilon, how does utility change?
OR: What is the relationship between epsilon and refinement?

i.e. if $\epsilon \geq \epsilon'$ then is it true that $P \cdot C_e \cdot P' \subseteq P \cdot C_{\epsilon'} \cdot P'$?
Properties of Refinement

**Theorem**: Given channels \( C, C' \) it holds that

\[
C \sqsubseteq C' \implies \epsilon(C) \geq \epsilon(C')
\]

(i.e. Epsilon is always monotonic on utility)

**Theorem**: If channels \( C, C' \) belong to the same “family” it holds that

\[
\epsilon(C) \geq \epsilon(C') \implies C \sqsubseteq C'
\]

(i.e. Utility is monotonic on epsilon).

**Note**: This was proven for KRR and Geometric “families” only.

Properties of Refinement

**Theorem**: Given channels $C$, $C'$ and a pre-processing step $P$ it holds that

$$C \subseteq C' \implies P \cdot C \subseteq P \cdot C'$$

**Corollary**: If $C$, $C'$ are both KRR or both Geometric then

$$\epsilon(C) \geq \epsilon(C') \implies P \cdot C \subseteq P \cdot C'$$

i.e. Increasing epsilon also increases utility (monotonicity holds).

Properties of Refinement

Theorem*: Given channels $C$, $C'$ and a post-processing step $P$ then

$$C \sqsubseteq C' \iff C \cdot P \sqsubseteq C' \cdot P$$

Corollary: If $C$, $C'$ are both KRR or both Geometric then

$$\epsilon(C) \geq \epsilon(C') \iff C \cdot P \sqsubseteq C' \cdot P$$

BUT: Can’t apply this result directly to our local DP workflow…

Privacy Workflows

**Issue:** Noise is added to each individual but post-processing is done on the combination of individuals.

Not $C_\epsilon \cdot P$ but n-fold composition of $C_\epsilon$ followed by $P$
Kronecker Composition

Personal data + Noise

\[
\begin{bmatrix}
0 & 1 \\
3/4 & 1/4 \\
1/4 & 3/4
\end{bmatrix}
\]

“Kronecker product”
Kronecker Composition

Properties of Kronecker product:

**Associativity:** \((A \otimes B) \otimes C = A \otimes (B \otimes C)\)

**Bilinearity:** \(A \otimes (B + C) = A \otimes B + A \otimes C\) and \((B + C) \otimes A = B \otimes A + C \otimes A\)

**Product respecting:** \((A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D)\) if \(A \cdot C\) and \(B \cdot D\) are defined.

**Invertibility:** If \(A, B\) are invertible, then so is \(A \otimes B\). The inverse is \((A \otimes B)^{-1} = A^{-1} \otimes B^{-1}\)

But \(\otimes\) is not commutative in general.
Kronecker Composition

**Theorem:** Given channels $C$, $C'$ then

$$C \subseteq C' \implies C \otimes^N \subseteq C' \otimes^N$$

**Corollary:** Given channels $C$, $C'$ then

$$\epsilon(C) \geq \epsilon(C') \implies \epsilon(C \otimes^N) \geq \epsilon(C' \otimes^N)$$

**Corollary:** Given channels $C$, $C'$ both KRR or both Geometric then

$$\epsilon(C) \geq \epsilon(C') \implies C \otimes^N \subseteq C' \otimes^N$$
Non-Monotonicity in Theory

**Theorem**: Given noise-adding channels $C$, $C'$ in the same family,

$$
e(C) \geq e(C') \implies C \sqsubseteq C' \implies C^\otimes N \sqsubseteq C'^\otimes N \implies C^\otimes N \cdot P \sqsubseteq C'^\otimes N \cdot P$$

i.e. There exist cases where we can get more privacy and more utility.
Non-Monotonicity in Practice

But... is this a problem for utility measures that are meaningful?

Yes!

Utility measured using mean absolute error

Loss(x, y) = |y - x|

Data source: https://github.com/propublica/compas-analysis
Conditions for Monotonicity

Under what conditions does the following hold?

\[ C^\otimes N \subseteq C'^\otimes N \implies C^\otimes N \cdot P \subseteq C'^\otimes N \cdot P \]

**Theorem:** Given channels \( C, C' \) s.t. \( C \subseteq C' \) with witness \( W \), and a post-processor \( P \), the following are sufficient conditions for monotonicity of utility on epsilon:

1. \( P \) has a left inverse \(-1P\)
2. \( P \cdot -1P \cdot W \cdot P = W \cdot P \)
Monotonicity Results

**Theorem:** If $R$, $R'$ are KRR mechanisms then

\[ R^\otimes N \subseteq R'^\otimes N \implies R^\otimes N \cdot P \subseteq R'^\otimes N \cdot P \]

whenever $P$ is a “counting” query.

**Corollary:** If $R$, $R'$ are KRR mechanisms then

\[ \epsilon(R) \geq \epsilon(R') \implies R^\otimes N \cdot P \subseteq R'^\otimes N \cdot P \]

whenever $P$ is a “counting” query.
Non-Monotonicity Results

Many negative results:

**Theorem:** If $R$, $R'$ are “KRR” channels then

$$R^\otimes N \subseteq R'^\otimes N \iff R^\otimes N \cdot P \subseteq R'^\otimes N \cdot P$$

whenever $P$ is a “sum” query.

Also does not hold in general for Geometric channels for counting or sum $P$. 
### Statistical Postprocessors

1. “Counting” query - deterministic channel $T$

<table>
<thead>
<tr>
<th>( R^\otimes 2 )</th>
<th>(yes, yes)</th>
<th>(yes, no)</th>
<th>(no, yes)</th>
<th>(no, no)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(yes, yes)</td>
<td>3/4 × 3/4</td>
<td>3/4 × 1/4</td>
<td>1/4 × 3/4</td>
<td>1/4 × 1/4</td>
</tr>
<tr>
<td>(yes, no)</td>
<td>3/4 × 1/4</td>
<td>3/4 × 3/4</td>
<td>1/4 × 1/4</td>
<td>1/4 × 3/4</td>
</tr>
<tr>
<td>(no, yes)</td>
<td>1/4 × 3/4</td>
<td>1/4 × 1/4</td>
<td>3/4 × 3/4</td>
<td>3/4 × 1/4</td>
</tr>
<tr>
<td>(no, no)</td>
<td>1/4 × 1/4</td>
<td>1/4 × 3/4</td>
<td>3/4 × 1/4</td>
<td>3/4 × 3/4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( T )</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(yes, yes)</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(yes, no)</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(no, yes)</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(no, no)</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( R^\otimes 2 ) ( T )</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(yes, yes)</td>
<td>1/4 × 1/4</td>
<td>3/4 × 1/4 + 1/4 × 3/4</td>
<td>3/4 × 3/4</td>
</tr>
<tr>
<td>(yes, no)</td>
<td>1/4 × 3/4</td>
<td>3/4 × 3/4 + 1/4 × 1/4</td>
<td>3/4 × 1/4</td>
</tr>
<tr>
<td>(no, yes)</td>
<td>3/4 × 1/4</td>
<td>1/4 × 1/4 + 3/4 × 3/4</td>
<td>1/4 × 3/4</td>
</tr>
<tr>
<td>(no, no)</td>
<td>3/4 × 3/4</td>
<td>1/4 × 3/4 + 3/4 × 1/4</td>
<td>1/4 × 1/4</td>
</tr>
</tbody>
</table>
2. “Sum” query - deterministic channel S

<table>
<thead>
<tr>
<th>R</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1/2</td>
<td>1/4</td>
<td>1/4</td>
</tr>
<tr>
<td>1</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>2</td>
<td>1/4</td>
<td>1/4</td>
<td>1/2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>02</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Main Takeaways:

- Refinement is a useful tool for reasoning about utility in differential privacy
- Kronecker products allow reasoning about utility in local differential privacy contexts
- Utility is not always monotonic on epsilon!

Future work:

- Study of monotonicity in machine learning contexts eg. DP-SGD.