(ε, δ)-Differential Privacy

Definition
Given ε, δ ≥ 0, a probabilistic query Q: X^n → R is (ε, δ)-differentially private iff for all adjacent database b_1, b_2 and for every S ⊆ R:

\[ \Pr[Q(b_1) \in S] \leq \exp(\varepsilon) \Pr[Q(b_2) \in S] + \delta \]
Multiple queries

**Question:** how much perturbation do we have if we want to answer $n$ counting queries with Laplace under $\varepsilon$-DP?
Question: how much perturbation do we have if we want to answer $n$ counting queries with Laplace under $\varepsilon$-DP?

Using standard composition we have as a max error

$$O\left(\frac{n}{\varepsilon_{\text{global}} n}\right) = O\left(\frac{1}{\varepsilon_{\text{global}}}\right)$$

Notice that if we don’t renormalize this is of the order of

$$O\left(\frac{n}{\varepsilon_{\text{global}}}\right)$$

bigger than the sample error.
Advanced Composition

**Question:** how much perturbation do we have if we want to answer n queries under \((\varepsilon, \delta)\)-DP?

Using advanced composition we have as a max error

\[
O\left(\frac{1}{\varepsilon_{\text{global}} \sqrt{n}}\right)
\]

If we don’t renormalize this is of the order of

\[
O\left(\frac{\sqrt{n}}{\varepsilon_{\text{global}}}\right)
\]

comparable to the sample error.

[DworkRothblumVadhan10, SteinkeUllman16]
Answering multiple queries

We have seen several methods to answer a single query:
- Randomized Response
- Laplace Mechanism
- Exponential Mechanism

And methods to answer multiple queries with small error:
- Standard composition - we can answer $\sqrt{n}$ queries.
- Advanced composition - we can answer $n$ queries.

**Question:** Can we do better?
SmallDB: Answering multiple linear queries

Algorithm 5 Pseudo-code for SmallDB

1: function SMALLDB\(D, Q, \epsilon, \alpha\)
2: \hspace{0.5cm} Let \(m = \frac{\log|Q|}{\alpha^2}\)
3: \hspace{0.5cm} Let \(u: \mathcal{X}^n \times \mathcal{X}^m \rightarrow \mathbb{R}\) be defined as:
4: \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} \hspace{0.5cm} u(D, D_i) = -\max_{q \in Q} |q(D) - q(D_i)|
5: \hspace{0.5cm} Let \(D' \leftarrow \mathcal{M}_E(D, u, \epsilon)\)
6: \hspace{0.5cm} return \(D'\)
7: end function
Finally, we may now state the utility theorem for SmallDB.

Theorem 4.5. By the appropriate choice of $\alpha$, letting $y$ be the database output by $\text{SmallDB}(x, Q, \varepsilon, \alpha^2)$, we can ensure that with probability $1 - \beta$:

$$
\max_{f \in Q} |f(x) - f(y)| \leq \frac{16 \log |X| \log |Q| + 4 \log \left( \frac{1}{\beta} \right)}{\varepsilon \alpha^3}.
$$

(4.2)

Equivalently, for any database $x$ with

$$
\|x\|_1 \geq \frac{16 \log |X| \log |Q| + 4 \log \left( \frac{1}{\beta} \right)}{\varepsilon \alpha^3}
$$

with probability $1 - \beta$: $\max_{f \in Q} |f(x) - f(y)| \leq \alpha$.

Answering multiple queries

We have seen several methods to answer a single query:
- Randomized Response
- Laplace Mechanism
- Exponential Mechanism

And methods to answer multiple queries with small error:
- Standard composition - we can answer $\sqrt{n}$ queries.
- Advanced composition - we can answer $n$ queries.

If we allow coordinating noise among different queries we can answer an exponential number of queries.
Data Release: IDC
Data Release: IDC

\[ \{Q_1, \ldots, Q_n\} \]
Data Release: IDC

\{Q_1, ..., Q_n\}
Iterative Database Construction

Given a database $D$, a set of queries $\{Q_1, ..., Q_n\}$ and a target accuracy $\alpha$, it generates a sequence $D_1, ..., D_n$ of synthetic DBs such that:

- it gives increasingly better approximations of $D$,
- $D_{t+1}$ is generated by $D_t$ using only one query $Q_t$ maximizing the difference:
  \[ |Q_t(D) - Q_t(D_t)| \geq k \]
- $D_n$ satisfies the target accuracy $\alpha$. 
Private Data Release

\{Q_1, ..., Q_n\}

Differential Privacy: the idea

A. Haeberlen

Promising approach: Differential privacy

USENIX Security (August 12, 2011)

Private data

N(illness, >1955)?

826±10

N(brain tumor, 05-22-1955)?

3 ±700

Noise

Differential Privacy:
Ensuring that the presence/absence of an individual has a negligible statistical effect on the query's result.

Trade-off between utility and privacy.
Private Iterative Database Construction

Given a database $D$, a set of queries $\{Q_1, \ldots, Q_n\}$ and a target accuracy $\alpha$ it generates a sequence $D_1, \ldots, D_n$ of synthetic DBs such that:

- it gives increasingly better approximations of $D$,
- $D_{t+1}$ is privatelly generated by $D_t$ using only one query $Q_t = Q_t + \text{noise}$ maximizing the difference:
  $$| Q_t(D) - Q_t(D_t) | \geq k$$
- $D_n$ satisfies the target accuracy $\alpha$. 
MWEM
Multiplicative Weight Exponential Mechanism

An algorithm for IDC for linear queries based on:
- the Exponential mechanism to select the query maximizing the difference,
- the Multiplicative Weight update rule to update the database.
Dataset as a distribution over the universe

The algorithm views a dataset $D$ as a distribution over rows $x \in \mathcal{X}$.

$$D(x) = \frac{\#\{i \in [n] : d_i = x\}}{n}$$

Then,

$$q(D) = \mathbb{E}_{x \rightarrow D}[q(x)]$$

We denote by $D_0$ the uniform distribution over $\mathcal{X}$.

| $D_0$ | 1/n | 1/n | 1/n | 1/n | 1/n | 1/n | ...... | ...... | ...... | 1/n |
Select the query with the exponential Mechanism Update part using the MW algorithm and a threshold obtained with Laplace.

Let's now fix $i$, we want to bound the ratio of the probability that $i$ is selected when running RNM on $D$ and $D_{\hat{\epsilon}}$. Now let $r_j$ be the noise drawn from Laplace for the round $j$ and $r_i$ be the noise drawn from Laplace for all the rounds except the $i$-th one.

The algorithm views a dataset $D$ as a distribution over rows $x \in X$. $D(x) = \#\{i \in [n]: d_i = x\}$

We denote by $D_0$ the uniform distribution over $X$.

Algorithm 9: Pseudo-code for MWEM

1: function MWEM($D$, $q_1$, ..., $q_m$, $T$, $\epsilon$, $D_0$)
2: for $i \leftarrow 1, \ldots, T$ do
3: $u_i(D, q) = |q(D_{i-1}) - q(D)|$
4: $\hat{q} \leftarrow \text{ExpMech}(D, u_i, n\epsilon/2T)$
5: $m_i \leftarrow \hat{q}(D) + \text{Lap}(T/n\epsilon)$
6: $D_i(x) = D_{i-1}(x) \times \exp(\hat{q}(x) \frac{m - \hat{q}(D_{i-1})}{2})$
7: $D_i = \text{renormalize}(D_i)$
8: end for
9: return $\text{avg}_{i<T} D_i$
10: end function

Theorem 1.13. MWEM satisfies $\epsilon$-di-\text{ernetial privacy.}$

Theorem 1.14. For any database $D$, set of linear queries $Q$, $T \in \mathbb{N}$, and $\epsilon > 0$, if $\hat{D} = \text{MWEM}(D, Q, T, \epsilon)$, then

$$\Pr[\max_{q \in Q} |q(\hat{D}) - q(D)| > 2 \frac{\log(|X|)}{T} + 10 \frac{T}{\log(|Q|)} \leq \epsilon, \epsilon]$$
MW Intuition

\[ D_i(x) = D_{i-1}(x) \times \exp(q(x) \frac{(q(D) - q(D_{i-1}))}{2n}) \]

For a given query q:

- If \( q(D) \gg q(D_{i-1}) \), we should scale up the weights on records contributing positively, and scale down the ones contributing negatively,

- If \( q(D) \ll q(D_{i-1}) \), we should scale down the weights on records contributing positively, and scale up the ones contributing negatively.
Let’s consider counting queries.

\[ \hat{q}_1(x_1) \quad \hat{q}_1(x_2) \quad \hat{q}_1(x_3) \quad \hat{q}_1(x_4) \quad \hat{q}_1(x_5) \quad \hat{q}_1(x_6) \ldots \ldots \ldots \ldots \ldots \hat{q}_1(x_{|x|}) \]

\[ \begin{array}{cccccccccc}
0 & 1 & 1 & 0 & 1 & 0 & \ldots & \ldots & \ldots & 1 \\
\downarrow & \uparrow & \uparrow & \downarrow & \uparrow & \downarrow & \ldots & \ldots & \ldots & \uparrow \\
\end{array} \]

\[ D_1 \]

\[ \begin{array}{cccccccccc}
1/n & e^{1/2n}/n & e^{1/2n}/n & 1/n & e^{1/2n}/n & 1/n & \ldots & \ldots & \ldots & e^{1/2n}/n \\
\end{array} \]

Let’s assume \( \hat{q}_1(D) - \hat{q}(D_0) = 1 \) then \( D_1(x_j) = D_0(x_j) \times \exp\left( \frac{q_1(x_j)}{2n} \right) \)
Let's consider counting queries.

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D0</td>
<td>1/n</td>
<td>1/n</td>
<td>1/n</td>
<td>1/n</td>
<td>1/n</td>
<td>1/n</td>
<td>1/n</td>
<td>1/n</td>
<td>1/n</td>
</tr>
<tr>
<td>q1(x1)</td>
<td>q1(x2)</td>
<td>q1(x3)</td>
<td>q1(x4)</td>
<td>q1(x5)</td>
<td>q1(x6)</td>
<td>……</td>
<td>……</td>
<td>……</td>
<td>q1(x</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>……</td>
<td>……</td>
<td>……</td>
<td>1</td>
</tr>
</tbody>
</table>

Let's assume \( \hat{q}_1(D) - \hat{q}(D_0) = -1 \) then \( D_1(x_j) = D_0(x_j) \times \exp(\frac{-q_1(x_j)}{2n}) \)
Algorithm 9 Pseudo-code for MWEM

1: function MWEM(D, q1, ..., qm, T, ε, D0)
2:     for i ← 1, ..., T do
3:         ui(D, q) = |q(Di−1) − q(D)|
4:         ̂q ← ExpMech(D, ui, nε/2T)
5:         mi ← ̂q(D) + Lap(T/nε)
6:         Di(x) = Di−1(x) × exp(̂q(x)(m−̂q(Di−1))/2)
7:         Di = renormalize(Di)
8:     end for
9:     return avgi<T Di
10: end function
Theorem 1.13. MWEM satisfies $\epsilon$-differential privacy.
**Accuracy Theorem:** MWEM achieves max-error:

\[
\alpha = O \left( \frac{\sqrt{\log |X| \cdot \log(1/\delta) \cdot \log |Q|}}{\varepsilon n} \right)^{1/2}.
\]

**Accuracy Theorem:** SmallDB achieves max-error:

\[
\alpha = O \left( \frac{\log |Q| \log |X|}{\varepsilon n} \right)^{1/3}.
\]
Keep a distribution over the databases, and search for a query which maximize the error, to be used in the update rule.
A dual approach

Keep a distribution over the queries, and search for a record which maximize the error, to be used in the update rule.

\{Q_1, \ldots, Q_n\}
In general we want to consider a set $Q$ of queries $q_1, \ldots, q_k$ closed under negation.

We denote by $Q_0$ the uniform distribution over $Q$:

\[
\begin{array}{cccccccc}
\frac{1}{|Q|} & \frac{1}{|Q|} & \frac{1}{|Q|} & \frac{1}{|Q|} & \frac{1}{|Q|} & \frac{1}{|Q|} & \ldots & \ldots & \ldots & \frac{1}{|Q|}
\end{array}
\]
Algorithm 11 Pseudo-code for DualQuery

1: function DualQuery(D, Q, C, s, α, Q₀)
2:  for i ← 1, . . . , C do
3:      Sample s queries q₁, . . . , qₛ from Q
4:      Find xᵢ such that
5:      \( \left( \frac{1}{s} \sum_j q_j^s(x_i) \right) \geq \left( \max_x \frac{1}{s} \sum_j q_j^s(x) \right) - \frac{\alpha}{4} \)
6:      \( Q_i(q) = Q_{i-1}(q) \times \exp(-\alpha \frac{q(x_i) - q(D)}{2n}) \)
7:      \( Q_i = \text{renormalize}(Q_i) \)
8:  end for
9:  return \( \bigcup_{i<C} x_i \)
10: end function
Theorem 1.15. DualQuery is differentially private for:

\[ \epsilon = \frac{\alpha T(T - 1)s}{4n} \]
Algorithm 11 Pseudo-code for DualQuery

1: function DualQuery($D$, $Q$, $T$, $s$, $\alpha$, $Q_1$)  
2: Sample $s$ queries $q_1^s$, ..., $q_s^s$ from $Q$  
3: Find $x_1$ such that  
4: \[
\left( \frac{1}{s} \sum_j q_j^s(x_1) \right) \geq \left( \max_x \frac{1}{s} \sum_j q_j^s(x) \right) - \alpha/4
\]
5: for $i \leftarrow 2, \ldots, T$ do  
6: \[
u_i(D, q) = \sum_{j=1}^{i-1} q(x_j) - q(D)\]
7: Sample $s$ queries $q_1^s$, ..., $q_s^s$ as  
8: \[
q_k^s \leftarrow \text{ExpMech}(D, u_i, \frac{\alpha(i-1)}{n})
\]
9: Find $x_i$ such that  
10: \[
\left( \frac{1}{s} \sum_j q_j^s(x_i) \right) \geq \left( \max_x \frac{1}{s} \sum_j q_j^s(x) \right) - \alpha/4
\]
11: end for  
12: return $\bigcup_{i \leq T} x_i$
13: end function
### DualQuery - Accuracy

<table>
<thead>
<tr>
<th>Theorem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy Theorem</strong>: DualQuery achieves max-error:</td>
<td>[ \alpha = O \left( \log^{1/2} \frac{Q \cdot \log^{1/6}(1/\delta) \cdot \log^{1/6}(2</td>
</tr>
<tr>
<td><strong>Accuracy Theorem</strong>: SmallDB achieves max-error:</td>
<td>[ \alpha = O \left( \log</td>
</tr>
<tr>
<td><strong>Accuracy Theorem</strong>: MWEM achieves max-error:</td>
<td>[ \alpha = O \left( \frac{\sqrt{\log</td>
</tr>
</tbody>
</table>
Algorithm 11 Pseudo-code for DualQuery

1. function **DualQuery**(*D*, *Q*, *T*, *s*, *α*, *Q₁*)
2. Sample *s* queries *q*_s¹*, ..., *q*_s^n* from *Q*
3. Find *x*_1 such that
4. \[ \left( \frac{1}{s} \sum_j q_j^s(x_1) \right) \geq \left( \max_x \frac{1}{s} \sum_j q_j^s(x) \right) - \alpha/4 \]
5. for *i* ← 2, ..., *T* do
6. \[ u_i(D, q) = \sum_{j=1}^{i-1} q(x_j) - q(D) \]
7. Sample *s* queries *q*_1^s*, ..., *q*_s^s* as
8. \[ q_k^s \leftarrow \text{ExpMech}(D, u_i, \frac{\alpha(i-1)}{n}) \]
9. Find *x*_i such that
10. \[ \left( \frac{1}{s} \sum_j q_j^s(x_i) \right) \geq \left( \max_x \frac{1}{s} \sum_j q_j^s(x) \right) - \alpha/4 \]
11. end for
12. return \( \bigcup_{i \leq T} x_i \)
13. end function
Example k-way marginals

Let's consider the universe domain \( \mathcal{X} = \{0, 1\}^d \) and let's consider \( \vec{v} \in \{1, \bar{1}, \ldots, d, \bar{d}\}^k \) with \( 1 \leq k \leq d \) and

\[
q_{\vec{v}}(x) = q_{v_1}(x) \land q_{v_1}(x) \land \cdots \land q_{v_k}(x)
\]

where \( q_j(x) = x_j \) and \( q_j(x) = \neg x_j \)

We call a conjunction or k-way marginal the associated counting query

\[
q_{\vec{v}} : \mathcal{X}^n \rightarrow [0, 1]
\]

We can create a corresponding integer program problem.
Example 3-way marginals

\[
\max \sum_i c_i + \sum_j d_j
\]

with \(\forall \hat{u}_i = q_{abc} : x_a + x_b + x_c \geq 3c_i\)

\(\forall \hat{v}_j = q_{abc} : (1 - x_a) + (1 - x_b) + (1 - x_c) \geq d_j\)

\(x_i, c_i, d_i \in \{0, 1\}\)