MODULE 2:
DIFFERENTIAL PRIVACY
Module 2: Differential Privacy

• Differential Privacy Definition

• Basic Algorithms
  – Laplace & Exponential Mechanism
  – Randomized Response

• Composition Theorems
Differential Privacy

For every pair of inputs that differ in one row

\(D_1\) \(\quad\) \(D_2\)

For every output …

\(O\)

Adversary should not be able to distinguish between any \(D_1\) and \(D_2\) based on any \(O\)

\[
\log \left( \frac{\Pr[A(D_1) = O]}{\Pr[A(D_2) = O]} \right) < \varepsilon \quad (\varepsilon > 0)
\]

[Dwork ICALP 2006]
Why pairs of datasets *that differ in one row*?

For every pair of inputs that differ in one row:

- $D_1$
- $D_2$

Simulate the presence or absence of a single record

For every output …

- $O$
Why *all* pairs of datasets ...?

For every pair of inputs that differ in one row

\[ D_1 \quad D_2 \]

Guarantee holds no matter what the other records are.

For every output ...

\[ O \]
Why *all* outputs?

Why *all* outputs?

A(D₁) = O₁

\[ P [ A(D₁) = O₁ ] \]

\[ P [ A(D₂) = O_k ] \]

Set of all outputs
Should not be able to distinguish whether input was $D_1$ or $D_2$ no matter what the output.

Worst discrepancy in probabilities.
Privacy Parameters

For every pair of inputs that differ in one row

\[ \Pr[A(D_1) = O] \leq e^\varepsilon \Pr[A(D_2) = O] \]

Controls the degree to which \( D_1 \) and \( D_2 \) can be distinguished. Smaller the \( \varepsilon \) more the privacy (and better the utility)
Outline of the Module 2

• Differential Privacy

• Basic Algorithms
  – Laplace & Exponential Mechanism
  – Randomized Response

• Composition Theorems
Can deterministic algorithms satisfy differential privacy?
Deterministic Algorithms do not satisfy differential privacy

Space of all inputs

Space of all outputs (at least 2 distinct outputs)
Deterministic Algorithms do not satisfy differential privacy

Each input mapped to a distinct output.
There exist two inputs that differ in one entry mapped to different outputs.

\[ Pr > 0 \]

\[ Pr = 0 \]
Random Sampling …

… also does not satisfy differential privacy

\[
\Pr[D_2 \rightarrow O] = 0 \implies \log\left(\frac{\Pr[D_1 \rightarrow O]}{\Pr[D_2 \rightarrow O]}\right) = \infty
\]
Output Randomization

- Add noise to answers such that:
  - Each answer does not leak too much information about the database.
  - Noisy answers are close to the original answers.
Laplace Mechanism

Privacy depends on the $\lambda$ parameter

$$h(\eta) \propto \exp(-\eta / \lambda)$$

Mean: 0,

Variance: $2 \lambda^2$

Laplace Distribution – $\text{Lap}(\lambda)$

Mean: 0,

Variance: $2 \lambda^2$
How much noise for privacy?

[Sensitivity]

Consider a query $q: I \rightarrow R$. $S(q)$ is the smallest number s.t. for any neighboring tables $D$, $D'$,

$$|q(D) - q(D')| \leq S(q)$$

[Thm]

If sensitivity of the query is $S$, then the following guarantees $\varepsilon$-differential privacy.

$$\lambda = \frac{S}{\varepsilon}$$
Sensitivity: COUNT query

- Number of people having disease
- Sensitivity = 1

- Solution: $3 + \eta$, where $\eta$ is drawn from Lap$(1/\varepsilon)$
  - Mean = 0
  - Variance = $2/\varepsilon^2$
Sensitivity: SUM query

• Suppose all values \( x \) are in \([a, b]\)

• Sensitivity = \( b \)
Privacy of Laplace Mechanism

- Consider neighboring databases \( D \) and \( D' \)
- Consider some output \( O \)

\[
\frac{\Pr[A(D) = O]}{\Pr[A(D') = O]} = \frac{\Pr[q(D) + \eta = O]}{\Pr[q(D') + \eta = O]}
\]

\[
= \frac{e^{-|O - q(D)|/\lambda}}{e^{-|O - q(D')|/\lambda}}
\]

\[
\leq e^{S(q)/\lambda} = e^\varepsilon
\]
Utility of Laplace Mechanism

• Laplace mechanism works for any function that returns a real number

• Error: \( E(\text{true answer} - \text{noisy answer})^2 \)

\[
= \text{Var}( \text{Lap}(S(q)/\varepsilon) ) \\
= 2 \cdot S(q)^2 / \varepsilon^2
\]
Exponential Mechanism

• For functions that do not return a real number …
  – “what is the most common nationality in this room”: Chinese/Indian/American…

• When perturbation leads to invalid outputs …
  – To ensure integrality/non-negativity of output
Exponential Mechanism

Consider some function \( f \) (can be deterministic or probabilistic):

How to construct a differentially private version of \( f \)?
Exponential Mechanism

• Scoring function $w: \text{Inputs} \times \text{Outputs} \rightarrow R$

• $D$: nationalities of a set of people
• $\#(D, O)$: # people with nationality $O$
• $f(D)$: most frequent nationality in $D$
• $w(D, O) = |\#(D, O) - \#(D, f(D))|$
Exponential Mechanism

• Scoring function $w: \text{Inputs} \times \text{Outputs} \rightarrow \mathbb{R}$

• Sensitivity of $w$

$$\Delta_w = \max_{O \& D, D'} |w(D, O) - w(D, O')|$$

where $D, D'$ differ in one tuple
Exponential Mechanism

Given an input $D$, and a scoring function $w$,

Randomly sample an output $O$ from $Outputs$ with probability

$$\frac{e^{\frac{\varepsilon}{2\Delta}}w(D,O)}{\sum_{Q \in Outputs} e^{\frac{\varepsilon}{2\Delta}}w(D,Q)}$$

- Note that for every output $O$, probability $O$ is output $> 0$. 
Randomized Response (a.k.a. local randomization)

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disease (Y/N)</td>
<td>Disease (Y/N)</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

With probability $p$, report the true value.

With probability $1-p$, report the flipped value.
Differential Privacy Analysis

• Consider 2 databases D, D’ (of size M) that differ in the jth value
  \[ D[j] \neq D'[j]. \text{ But, } D[i] = D'[i], \text{ for all } i \neq j \]

• Consider some output O

\[
\frac{P(D \rightarrow O)}{P(D' \rightarrow O)} \leq e^\varepsilon \iff \frac{1}{1 + e^\varepsilon} < p < \frac{e^\varepsilon}{1 + e^\varepsilon}
\]
Utility Analysis

- Suppose $n_1$ out of $N$ people replied “yes”, and rest said “no”
- What is the best estimate for $\pi = \text{fraction of people with disease} = Y$?
  
  \[
  \hat{\pi} = \frac{n_1/n - (1-p)}{(2p-1)}
  \]

- $E(\hat{\pi}) = \pi$
- $\text{Var}(\hat{\pi}) = \frac{\pi(1 - \pi)}{n} + \frac{1}{n(16(p - 0.5)^2 - 0.25)}$

Sampling \hspace{1cm} Variance due to coin flips
Laplace Mechanism vs Randomized Response

Privacy

• Provide the same $\epsilon$-differential privacy guarantee

• Laplace mechanism assumes data collected is trusted

• Randomized Response does not require data collected to be trusted
  
  – Also called a Local Algorithm, since each record is perturbed
Laplace Mechanism vs Randomized Response

Utility

• Suppose a database with $N$ records where $\mu N$ records have disease $= Y$.

• Query: # rows with Disease=Y

• Std dev of Laplace mechanism answer: $O(1/\varepsilon)$

• Std dev of Randomized Response answer: $O(\sqrt{N})$
Outline of the Module 2

• Differential Privacy

• Basic Algorithms
  – Laplace & Exponential Mechanism
  – Randomized Response

• Composition Theorems
Why Composition?

• Reasoning about privacy of a complex algorithm is hard.

• Helps software design
  – If building blocks are proven to be private, it would be easy to reason about privacy of a complex algorithm built entirely using these building blocks.
A bound on the number of queries

• In order to ensure utility, a statistical database must leak some information about each individual
• We can only hope to bound the amount of disclosure
• Hence, there is a limit on number of queries that can be answered
Dinur Nissim Result

• A vast majority of records in a database of size $n$ can be reconstructed when $n \log(n)^2$ queries are answered by a statistical database …

… even if each answer has been arbitrarily altered to have up to $o(\sqrt{n})$ error.
Sequential Composition

- If $M_1, M_2, \ldots, M_k$ are algorithms that access a private database $D$ such that each $M_i$ satisfies $\varepsilon_i$-differential privacy,

then the combination of their outputs satisfies $\varepsilon$-differential privacy with $\varepsilon = \varepsilon_1 + \ldots + \varepsilon_k$. 
Privacy as Constrained Optimization

• Three axes
  – Privacy
  – Error
  – Queries that can be answered

• E.g.: Given a fixed set of queries and privacy budget $\varepsilon$, what is the minimum error that can be achieved?
Parallel Composition

• If $M_1, M_2, \ldots, M_k$ are algorithms that access disjoint databases $D_1, D_2, \ldots, D_k$ such that each $M_i$ satisfies $\varepsilon_i$-differential privacy,

then the combination of their outputs satisfies $\varepsilon$-differential privacy with $\varepsilon = \max \{ \varepsilon_1, \ldots, \varepsilon_k \}$
Postprocessing

- If $M_1$ is an $\varepsilon$-differentially private algorithm that accesses a private database $D$, then outputting $M_2(M_1(D))$ also satisfies $\varepsilon$-differential privacy.
Case Study: K-means Clustering
Kmeans

- Partition a set of points $x_1, x_2, \ldots, x_n$ into $k$ clusters $S_1, S_2, \ldots, S_k$ such that the following is minimized:

$$
\sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|_2^2
$$

Mean of the cluster $S_i$
Kmeans

Algorithm:

• Initialize a set of $k$ centers
• Repeat
  Assign each point to its nearest center
  Recompute the set of centers
  Until convergence …

• Output final set of $k$ centers
Differentially Private Kmeans

• Suppose we fix the number of iterations to T

• In each iteration (given a set of centers):
  1. Assign the points to the new center to form clusters
  2. Noisily compute the size of each cluster
  3. Compute noisy sums of points in each cluster
Differentially Private Kmeans

• Suppose we fix the number of iterations to $T$

  If each iteration uses $\varepsilon/T$ privacy budget, total privacy loss is $\varepsilon$

• In each iteration (given a set of centers):

  1. Assign the points to the new center to form clusters

  2. Noisily compute the size of each cluster

  3. Compute noisy sums of points in each cluster
Differentially Private Kmeans

• Suppose we fix the number of iterations to $T$ (Each iteration uses a privacy budget of $\frac{\epsilon}{T}$)

• In each iteration (given a set of centers):
  1. Noisily compute the size of each cluster
  2. Compute noisy sums of points in each cluster
  3. Recompute new clusters based on 1. and 2.

Sensitivity = 1

Sensitivity = size of domain = $|\text{dom}|$

Postprocessing (no impact on privacy)
Differentially Private K-means

• Suppose we fix the number of iterations to $T$ (Each iteration uses a privacy budget of $\frac{\varepsilon}{T}$)

• In each iteration (given a set of centers):

  1. Noisily compute the size of each cluster

  2. Compute noisy sums of points in each cluster

  3. Recompute new clusters based on 1. and 2.
Differentially Private Kmeans

• Suppose we fix the number of iterations to $T$ (Each iteration uses a privacy budget of $\varepsilon / T$)

• In each iteration (given a set of centers):

  1. Noisily compute the size of each cluster
  2. Compute noisy sums of points in each cluster
  3. Recompute new clusters based on 1. and 2.

$$\text{Laplace}(2T/\varepsilon)$$

$$\text{Laplace}(2T \cdot |\text{dom}| / \varepsilon)$$

$$\text{Compute exactly}$$
Results \((T = 10\) iterations, random initialization)\)

**Original Kmeans algorithm**

**Laplace Kmeans algorithm**

- Even though we noisily compute centers, Laplace kmeans can distinguish clusters that are far apart.

- Since we add noise to the sums with sensitivity proportional to \(|\text{dom}|\), Laplace k-means can’t distinguish small clusters that are close by.