1 K-Anonymity

Types of features of a dataset:
- *identifiers*: are unique to an individual, adversary might know it
- *non-identifiers*: are unrelated to the user's identity and **unknown** to the adversary
- *quasi-identifiers*: can be used to partially identify an individual

Definition *(k-anonymity).* A dataset is *k-anonymous* if for each individual, there are at least $k-1$ other individuals in the dataset that have the same value for all "quasi-identifiers."

Typical ways of achieving *k-anonimity*:
- *suppression*: removing values
- *generalization*: replacing a value with a range

Attacks on *k-anonimity*:
- *Homogeneity attack*
- *Background knowledge attack*
- *Composition attack*
- *Downcoding attacks*

2 Subset queries

- each individual's data in the database $D = (z, s)$ is a pair (z_i, s_i) where $z_i \in \mathcal{Z}$ is the identifier and $s_i \in \{0,1\}$ is the private bit. We assume $z_i \neq z_j$ for $i \neq j$, and known to the adversary
- the adversary makes *counting queries* of the form:

$$f_\phi(D) = \frac{1}{n}\sum_{i=1}^{n} \phi(z_i)\cdot s_i = \frac{\phi(z)\cdot s}{n}$$

  for some predicate $\phi : \mathcal{Z} \to \{0,1\}$.
- the curator answers the query by returning $\texttt{Ans}(\phi) = f_\phi(D) \pm e$ where e is some (necessary, or else $\phi(z) = 1$ iff $z = z_i$ recovers s_i) noise.

Definition *(Blatant non-privacy).* The curator's algorithm is *blatantly non-private* if the adversary can, with *high probability* $1 - o(1)$ (for $n \to \infty$) construct a database $\tilde{s} \in \{0,1\}^n$ such that s and $\tilde{s}$ differ in $o(n)$ (sublinear in n) coordinates.

2.a Dinur and Nissim's theorems

Theorem 1. If the adversary can ask 2^n subset queries on a database of size n, and the curator adds noise of size at most E to each query, then the adversary can output a guess $\tilde{s} \in \{0,1\}^n$ such that s and $\tilde{s}$ differ in at most $4En$ positions.

Examples: $E = o(1)$, (i.e. shrinking noise as $n \to \infty$) then the curator's algorithm is blatantly non-private (as $4 \cdot o(1) \cdot n = o(n)$).

Theorem 2. If the adversary can ask n counting queries on a database of size n, and the curator adds noise of size at most

$$E = o\left(\frac{1}{\sqrt{n}}\right) \qquad \left(\lim_{n\to\infty} \sqrt{n}\cdot E = 0\right)$$

to each query, then the curator's algorithm is blatantly non-private.

3 Randomized response

Instead of reporting their true bit x_i each individual instead reports

$$y_i = \begin{cases} x_i & \text{with probability } \frac{1}{2}+\gamma \\ 1-x_i & \text{with probability } \frac{1}{2}-\gamma \end{cases}$$

for some fixed $\gamma \in \left[0, \frac{1}{2}\right]$. Privacy is given by *plausible deniability.*

Utility: let's say we want to compute $p = \frac{1}{n}\sum_{i=1}^{n} x_i$. We can recover an unbiased estimate by measuring

$$\hat{p} = \frac{1}{2\gamma}\cdot\left(\frac{1}{n}\sum_{i=1}^{n} y_i - \frac{1}{2} + \gamma\right) \quad \text{so that} \quad \mathbb{E}[\hat{p}] = p$$

Since $\hat{p}$ is a sum of n independent random variables (shifted Bernoulli) we can apply concentration inequalities (e.g. Chernoff) to show that with high probability

$$|p - \hat{p}| \le \tilde{O}\left(\frac{1}{\gamma\sqrt{n}}\right)$$

where $\tilde{O}(\cdot)$ hides logarithmic factors. If we want an estimate with an additive error at most α we need $n = \tilde{O}\left(\frac{1}{\gamma^2\alpha^2}\right)$.

3.a Differential Privacy of Randomized Response

To analyze the privacy of Randomized Response we consider a mechanism $M(x_1, ..., x_n) = (y_1, ..., y_n)$ that outputs all the randomized responses.

Let $\vec{b} \in \{0,1\}^n$ be a set of randomized responses, $D \sim D'$ neighboring databases that differ in the value of x_j. Then we have:

$$\frac{\Pr\left[M(D) = \vec{b}\right]}{\Pr\left[M(D') = \vec{b}\right]} = \frac{\prod_{i=1}^{n}\Pr[y_i = b_i]}{\prod_{i=1}^{n}\Pr[y'_i = b_i]} = \frac{\Pr[y_j = b_j]}{\Pr[y'_j = b_j]} \le \frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}$$

Where the last bound comes from the highest possible probability of the event $y_j = b_j$ and the lowest for $y'_j = b_j$.

If γ is small (e.g. $\gamma \le \frac{1}{4}$), RR is ε-differentially private for $\varepsilon = O(\gamma)$. (The first-order Maclaurin expansion of $\ln\left(\frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}\right)$ is 4γ).

4 Differential Privacy

Definition *(Neighboring databases).* Two databases D and D' are *neighboring*, denoted $D \sim D'$ if $|D| = |D'|$ and they differ in at most one row (alternative definitions allow for addition or removal of a single row, and are kind of equivalent).

Definition *(Differential Privacy).* A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ is ε-differentially private if, for all $D \sim D'$ and for all events $S \subseteq \mathcal{Y}$,

$$\Pr[M(D) \in S] \le e^\varepsilon \cdot \Pr[M(D') \in S]$$

4.a Properties of Differential Privacy

Theorem *(Post-Processing).* Let $M : \mathcal{X}^n \to \mathcal{Y}$ be ε-differentially private and let $f : \mathcal{Y} \to \mathcal{Z}$ be any (possibly randomized) function. Then $f \circ M$ is ε-differentially private.

Theorem *(Basic Composition).* Let $M = (M_1, M_2, ..., M_k)$ be a sequence of algorithms where M_i is ε_i-differentially private, and the algorithms can be chosen sequentially and adaptively (i.e. M_i can depend on the output of $M_1, ..., M_{i-1}$). Then M is $(\sum_{i=1}^{k}\varepsilon_i)$-differentially private.

Proof *(independent M_i, holds for adaptive case too).* Given two neighboring databases $D \sim D'$ and output $y = (y_1, ..., y_k)$, we have:

$$\prod_{i=1}^{k}\frac{\Pr[M_i(D) = y_i \mid (M_1(D), ..., M_{i-1}(D)) = (y_1, ..., y_{i-1})]}{\Pr[M_i(D') = y_i \mid (M_1(D'), ..., M_{i-1}(D')) = (y_1, ..., y_{i-1})]}$$

$$\le \prod_{i=1}^{k} e^{\varepsilon_i} = \exp\left(\sum_{i=1}^{k}\varepsilon_i\right)$$

Theorem *(Group Privacy).* Let $M : \mathcal{X}^n \to \mathcal{Y}$ be ε-differentially private, and let D and D' be databases that differ in up to k rows. Then, for all events $S \subseteq \mathcal{Y}$,

$$\Pr[M(D) \in S] \le e^{k\varepsilon} \cdot \Pr[M(D') \in S]$$

5 The Laplace Mechanism

Definition *(ℓ_1-Sensitivity).* The ℓ_1-sensitivity of a function $f : \mathcal{X}^n \to \mathbb{R}^k$ is defined as

$$\Delta_1 = \max_{D\sim D'} \|f(D) - f(D')\|_1 = \max_{D,D'}\sum_{i=1}^{k}\left|f(D)_i - f(D')_i\right|$$

where D and D' are neighboring databases.

Definition *(Laplace Distribution).* The Laplace distribution $\texttt{Laplace}(\mu, b)$ with location μ and scale $b > 0$ has:
- density $f(x \mid \mu, b) = \frac{1}{2b}e^{-\frac{|x-\mu|}{b}}$ for all $x \in \mathbb{R}$
- mean μ, variance $2b^2$

Definition *(Laplace Mechanism).* Let $f : \mathcal{X}^n \to \mathbb{R}^k$ be a function with ℓ_1-sensitivity Δ_1, and let $\varepsilon > 0$. The laplace mechanism is defined as:

$$M(D) = f(D) + (\mathcal{Y}_1, ..., \mathcal{Y}_k)$$

where the $\mathcal{Y}_i$ are i.i.d. samples from the $\texttt{Laplace}\left(0, \frac{\Delta_1}{\varepsilon}\right)$ distribution.

Theorem. The Laplace Mechanism is ε-DP.

Proof. Let $D \sim D'$ be neighboring databases. Let $p_D(y)$ and $p_{D'}(y)$ be the probability density functions of the mechanism outputs $M(D)$ and $M(D')$ respectively, evaluated at an arbitrary point $y \in \mathbb{R}^k$.

Let $b = \frac{\Delta_1}{\varepsilon}$.

$$\begin{aligned}
\frac{p_{D(y)}}{p_{D'}(y)} &= \frac{\prod_{i=1}^{k}\frac{1}{2b}\exp\left(-\frac{|y_i - f(D)_i|}{b}\right)}{\prod_{i=1}^{k}\frac{1}{2b}\exp\left(-\frac{|y_i - f(D')_i|}{b}\right)} \\
&= \prod_{i=1}^{k}\frac{1}{2b}\exp\left(\frac{|y_i - f(D')_i| - |y_i - f(D)_i|}{b}\right) \\
&\le \prod_{i=1}^{k}\frac{1}{2b}\exp\left(\frac{|f(D')_i - f(D)_i|}{b}\right) \\
&= \exp\left(\frac{\|f(D')_i - f(D)_i\|_1}{b}\right) \\
&\le \exp\left(\frac{\Delta_1}{b}\right) = \exp(\varepsilon)
\end{aligned}$$

6 Bounding the error of mechanisms

Randomized Response with high probability $1 - o(1)$ has error at most $\tilde{O}\left(\frac{1}{\gamma\sqrt{n}}\right) \approx \tilde{O}\left(\frac{1}{\varepsilon\sqrt{n}}\right)$ for m.

For the Laplace Mechanism we use the tail bound:

$$Z \sim \text{Laplace}(0, b) \wedge t > 0 \Rightarrow \Pr[|Z| \ge tb] \le \exp(-t)$$

Hence, the noise is bounded by $\tilde{O}\left(\frac{k}{\varepsilon n}\right)$

7 Approximate Differential Privacy

Definition *Approximate Differential Privacy.* An algorithm $M : \mathcal{X} \to \mathcal{Y}$ is (ε, δ)-differentially private if, for all neighboring databases D and D' and all events $S \subseteq \mathcal{Y}$ we have

$$\Pr[M(D) \in S] \le e^\varepsilon \cdot \Pr[M(D') \in S] + \delta$$

Note:
- $(\varepsilon, 0)$-DP $= \varepsilon$-DP
- δ can be seen as the "failure probability" of DP
- δ *must* be $o\left(\frac{1}{n}\right)$ to get a meaningful privacy guarantee (TODO: PROOF)
- unlike in pure DP, here we **must** consider all possible subsets of outputs rather than just single outputs

7.a Properties of Approximate DP

Postprocessing: (ε, δ)-DP is closed under postprocessing

Group privacy: approximate DP scales gracefully to larger groups (linear for ε, weirder for δ)

Theorem *(Basic composition of approximate DP).* Let $M = (M_1, M_2, ..., M_k)$ be a sequence of algorithms where M_i is $(\varepsilon_i, \delta_i)$-differentially private, and the algorithms can be chosen sequentially and adaptively. Then M is $(\sum_{i=1}^{k}\varepsilon_i, \sum_{i=1}^{k}\delta_i)$-differentially private.

8 The Privacy Loss Variable

Definition *(Privacy Loss Random Variable).* Let D and D' be two databases, and M be a mechanism. The *privacy loss random variable* $\mathcal{L}_{M(D)\|M(D')}(y)$ is distributed by drawing $y \leftarrow M(D)$ and outputting:

$$\mathcal{L}_{M(D)\|M(D')}(y) := \ln\left(\frac{\Pr[M(D) = y]}{\Pr[M(D') = y)}\right)$$

(for mechanisms with continuous outputs we consider the ratio between densities). If either the numerator or denominator is 0, we say that the privacy loss is infinite.

Why is this useful? We have that

$$\Pr\left[\left|\mathcal{L}_{M(D)\|M(D')}(y)\right| > \varepsilon\right] \le \delta \Rightarrow M \text{ is } (\varepsilon, \delta)\text{-DP}$$

(the converse is not true).

9 The Gaussian Mechanism

Definition *(ℓ_2-Sensitivity).* The ℓ_2-sensitivity of a function $f : \mathcal{X}^n \to \mathbb{R}^k$ is defined as

$$\Delta_2 = \max_{D\sim D'} \|f(D) - f(D')\|_2 = \max_{D,D'}\sqrt{\sum_{i=1}^{k}\left(f(D)_i - f(D')_i\right)^2}$$

where D and D' are neighboring databases.

The ℓ_2 norm is never $>$ the ℓ_2 norm and can be as much as $\sqrt{k}$ times smaller, so it holds that:

$$\Delta_2 \le \Delta_1 \le \sqrt{k}\cdot\Delta_2$$

Definition *(Gaussian Distribution).* The univariate Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2 has density

$$f(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad x \in \mathbb{R}$$

Definition *(Gaussian Mechanism).* Let $f : \mathcal{X}^n \to \mathbb{R}^k$ be a function with l_2-sensitivity Δ_2, and let $\varepsilon, \delta > 0$. The Gaussian mechanism is defined as

$$M(D) = f(D) + (\mathcal{Y}_1, ..., \mathcal{Y}_k)$$

where the $\mathcal{Y}_i$ are i.i.d. samples from $\mathcal{N}\left(0, \frac{2\log(1.25/\delta)\Delta_2^2}{\varepsilon^2}\right)$.

If we can add arbitrary noise, we can achieve any approximate DP guarantee:

Theorem *(Approximate-DP of Gaussian Mechanism).* For any $\varepsilon \le 1$ and $\delta > 0$, the Gaussian mechanism is (ε, δ)-differentially private.

If we fix σ, we still have different approximate DP guarantees that compromise between ε and δ:

Theorem *(Alternative Approximate-DP of Gaussian Mechanism).* For any $\varepsilon > 0$ and $\sigma > 0$, the mechanism that adds Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the output of a function f with ℓ_2-sensitivity Δ_2 is $(\varepsilon, \delta(\varepsilon))$-differentially private. Where $\delta(\varepsilon) = 1.25\exp\left(\frac{-\varepsilon^2\sigma^2}{2\Delta_2^2}\right)$.

Tail bound for Gaussian: If $Z \sim N(0, \sigma^2)$, then for every $t > 0$:

$$\Pr(|Z| > t\cdot\sigma) \le 2\exp(-t^2/2)$$

9.a Answering counting queries with the GM

Theorem *(Advanced composition).* For all $\varepsilon > 0, \delta \in [0,1], \delta' \in (0,1]$, let $M = (M_1, ..., M_k)$ be a sequence of (ε, δ)-differentially private algorithms, where the M_is are potentially chosen sequentially and adaptively. Then M is $\left(\tilde{\varepsilon}, \tilde{\delta}\right)$-differentially private, where

$$\tilde{\varepsilon} = \varepsilon\sqrt{2k\log\left(\frac{1}{\delta'}\right)} + k\varepsilon\frac{e^\varepsilon - 1}{e^\varepsilon + 1} \quad \text{and} \quad \tilde{\delta} = k\delta + \delta'$$

Note that if ε is small (e.g. $\varepsilon \le 1$) from the Maclaurin series we get $\frac{e^\varepsilon - 1}{e^\varepsilon + 1} \approx \frac{\varepsilon}{2} \Rightarrow \tilde{\varepsilon} = \Theta\left(\varepsilon\sqrt{2k\log\left(\frac{1}{\delta'}\right)} + k\varepsilon^2\right)$ where $k\varepsilon^2$ is a lower-order term if ε is small enough. Specifically, if $\varepsilon < \frac{1}{\sqrt{k}}$ and we set $\delta' = \delta$, we get

$$\tilde{\varepsilon} = \Theta\left(\varepsilon\sqrt{k\log\left(\frac{1}{\delta}\right)}\right) \quad \text{and} \quad \tilde{\delta} = \Theta(k\delta)$$

This applies to any (ε, δ)-DP algorithm, proving that with approximate DP the composition grows roughly as $\sqrt{k}$ (instead of linearly as in pure DP).

10 Comparison For Counting Queries

Gaussian Mechanism can be beaten if the data domain $|\mathcal{X}|$ is not too large, e.g. by the *Private Multiplicative Weights* algorithm (as long as the data domain is not exponentially larger than the database size n).

Guarantee	Error per query	Mechanism	Lower bound
Local ε-DP	$\tilde{O}\left(\frac{1}{\varepsilon\sqrt{n}}\right)$	Randomized Response	$\Omega\left(\frac{1}{\varepsilon\sqrt{n}}\right)$
ε-DP	$\tilde{O}\left(\frac{k}{\varepsilon n}\right)$	Laplace	$\Omega\left(\frac{k}{\varepsilon n}\right)$
(ε,δ)-DP	$\tilde{O}\left(\frac{\sqrt{k}}{\varepsilon n}\right)$	Gaussian	$\Omega\left(\frac{\sqrt{k}}{\varepsilon n}\right)$
	$\tilde{O}\left(\frac{\sqrt{\log\lvert\mathcal{X}\rvert\log k}}{\varepsilon n}\right)^{1/2}$	Private Multiplicative Weights	$\tilde{\Omega}\left(\frac{\sqrt{\log\lvert\mathcal{X}\rvert\log k}}{\varepsilon n}\right)^{1/2}$

11 The Exponential Mechanism

Definition *(Selection Problem).* A *selection problem* is specified by:
- A set $\mathcal{Y}$ of possible outcomes (discrete or continuous);
- A score function $q : \mathcal{Y} \times \mathcal{X}^n \to \mathbb{R}$ which measures the "quality" of an output for a given input data set $D \in \mathcal{X}^n$;
- A sensitivity bound Δ such that $q(y;\cdot)$ has sensitivity at most Δ for every $y \in \mathcal{Y}$. That is, for all $y \in \mathcal{Y}$, and all neighboring inputs D and D', we have

$$|q(y;D) - q(y;D')| \le \Delta$$

Definition *(Exponential Mechanism).* Let $(\mathcal{X}, \mathcal{Y}, q, \Delta)$ be a selection problem. Then, the Exponential mechanism returns a random sample from the distribution over $\mathcal{Y}$ defined by

$$\Pr[\mathcal{Y} = y] \propto \exp\Big(\frac{\varepsilon}{2\Delta} q(y;D)\Big)$$

Example *(finite case)*:

$$\Pr[\mathcal{Y} = y] = \frac{\exp(\frac{\varepsilon}{2\Delta} q(y;D))}{\sum_{y' \in \mathcal{Y}} \exp(\frac{\varepsilon}{2\Delta} q(y';D))}$$

i.e. a *softmax* over the scores with a "temperature" of $\frac{2\Delta}{\varepsilon}$:
- small $\varepsilon \Rightarrow$ high temperature $\Rightarrow$ distribution is very uniform (output leaks less information about the scores)
- high $\varepsilon \Rightarrow$ low temperature $\Rightarrow$ difference between the scores is amplified and better approximates a true argmax

The Exponential Mechanism also works for continuous (and infinite) output spaces as long as the normalization constant is finite for all D.

Theorem *(Privacy of the Exponential Mechanism).* The Exponential Mechanism is ε-differentially private.

Lets denote the highest score as

$$\mathrm{OPT}(D) := \max_{y \in \mathcal{Y}} q(y;D)$$

Theorem. Suppose $\mathcal{Y}$ is finite and has size k. Then for every Δ-sensitive score function q, dataset D, and $\delta > 0$, the output of the Exponential Mechanism $M(D)$ satisfies:

$$\Pr_{y \leftarrow M(D)}\left[q(y;D) \le \mathrm{OPT}(D) - \frac{2\Delta\big(\log k + \log\big(\frac{1}{\delta}\big)\big)}{\varepsilon}\right] \le \delta$$

So, we get by with noise on the order of $\tilde{O}\big(\frac{\log k}{\varepsilon}\big)$.

12 Private Machine Learning

We have some distribution $\mathcal{P}$ over *inputs* $\mathcal{X}$ with *labels* $\mathcal{Y}$. We want to learn a model $f_\theta : \mathcal{X} \to \mathcal{Y}$ that will generalize to new data, where $\theta \in \mathbb{R}^d$ are the parameters of the model.

We formalize "generalization" via a *loss function* $\ell(\theta, x, y)$ that measures the error of f_θ on the pair (x, y). We then define the *risk* $\mathcal{L}(\theta)$ of a model as the expected loss over the distribution $\mathcal{P}$:

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(\theta, x, y)]$$

We sample i.i.d. pairs $(x, y) \sim \mathcal{P}$ to build a *training set* $D = \{(x_i, y_i)\}_{i=n}^n$ of size n. The *empirical risk* $\hat{\mathcal{L}}(\theta)$ of f_θ on D is then the average loss over the training set:

$$\hat{\mathcal{L}}(\theta) = \sum_{i=1}^n \ell(\theta, x_i, y_i)$$

Machine learning commonly involves *empirical risk minimization* (ERM), i.e. find $\theta^* \in \Theta$ s.t.

$$\theta^* = \arg\min_{\theta\in\Theta} \hat{\mathcal{L}}_D(\theta)$$

Given a model f_{θ^*} optimized over the training set D, we can evaluate its risk on the distribution $\mathcal{P}$ (e.g. by sampling a new *test* dataset). The risk computed over the training set will usually underestimate the true population risk. This is known as *overfitting* and measured by the *generalization gap*:

$$\left|\mathcal{L}(\theta^*) - \hat{\mathcal{L}}_D(\theta^*)\right|$$

12.a Private Empirical Risk Minimization

Other that standard non-private ERM we have three approaches:
- Output perturbation: We add noise to θ.
- Objective perturbation: We add noise to the objective function.
- Gradient perturbation: We use gradient descent with noise.

12.a.a Output perturbation

- compute the sensitivity of ERM, i.e. of the function $\arg\min_{\theta\in\Theta} \hat{\mathcal{L}}_D(\theta)$
- (if the loss has some strong regularity properties, i.e. if the model is not too complex, there exist bounds on the sensitivity)
- then just add the appropriate amount of Gaussian noise to the output of the ERM algorithm (the model's parameters)

12.a.b Objective perturbation

- instead of solving the exact (or regularized) ERM problem, we add a random term $b^T\theta$ to the optimization objective ($\mathcal{L}$), where b is sampled from a Gaussian distribution
- under suitable assumptions on the loss function, optimizing this with standard ERM is differentially private

12.a.c Gradient Perturbation

Algorithm *Private Projected Gradient Descent.*

$\theta \xleftarrow{\$} \Theta$ — random initialization
for $t = 0, ..., T-1$ **do**
- $g \leftarrow \nabla_{\theta_t}\hat{\mathcal{L}}_D(\theta_t) = \frac{1}{n}\sum_{i=1}^n \nabla_{\theta_t}\ell(\theta_t, x_i, y_i)$ — compute gradient
- $\theta_{t+1} \leftarrow \theta_t - \eta(g + \mathcal{N}(0, \sigma^2 I_{d\times d}))$ — update parameters
- $\theta_t + 1 \leftarrow \Pi_\Theta(\theta_{t+1})$ — project onto set Θ

The only computation that is data dependent is that of the gradient g
$\Rightarrow$ we can add noise to make it DP
$\Rightarrow$ by post-processing, each update step is DP
$\Rightarrow$ by composition the combination of T update steps is DP

The solution to the private ERM problem has an **empirical** (not about the generalization performance) risk roughly $\frac{\sqrt{d}}{\varepsilon}$ larger than the non-private version.

12.b Private Deep Learning

Hard to bound the sensitivity of the ERM output, or of individual gradients. So we enforce that individual gradients are bounded through **gradient clipping**: we *clip* each example's gradient to some maximum ℓ_2 norm C, then compute the average. g is now just the mean of some vectors of norm at most C, so we can get a bound on the ℓ_2 sensitivity.

Setting C:
- too high $\to$ overestimate the sensitivity and add too much noise
- too low $\to$ remove useful information from the gradients and harm optimization

Algorithm *Differentially Private Stochastic Gradient Descent (DP-SGD).*

$\theta \xleftarrow{\$} \Theta$ — random initialization
for $t = 0, ..., T-1$ **do**
- Sample $S \subseteq [n]$ of size m — select a random batch
- $g_i \leftarrow \nabla_{\theta_t}\ell(\theta_t, x_i, y_i), \forall i \in S$ — per-sample gradients
- $g_i \leftarrow C \cdot \frac{g_i}{\max\big(C, \|g_i\|_2\big)}, \forall i \in S$ — clip gradients
- $\overline{g} \leftarrow \frac{1}{|m|}\sum_{i\in S} g_i$ — aggregate gradients
- $g \leftarrow \nabla_{\theta_t}\hat{\mathcal{L}}_D(\theta_t) = \frac{1}{n}\sum_{i=1}^n \nabla_{\theta_t}\ell(\theta_t, x_i, y_i)$ — compute gradient
- $\theta_{t+1} \leftarrow \theta_t - \eta(\overline{g} + \mathcal{N}(0, \sigma^2 I_{d\times d}))$ — update parameters

Privacy. $\|g_i\|_2 \le C \Rightarrow \Delta_2 \le \frac{2C}{m}$, so we can add Gaussian noise of variance $\sigma^2 = \frac{2\log(\frac{1.25}{\delta})}{\varepsilon^2} \cdot \big(\frac{2C}{m}\big)^2$ to get a (ε, δ)-DP guarantee for each gradient step.

With basic composition, after T steps we would get a total privacy budget that grows as $(\varepsilon T, \delta T)$, which would be unfeasible.

With advanced composition we get that DP-SGD is $\Big(\varepsilon\sqrt{T\log\big(\frac{1}{\delta}\big)}, \delta T\Big)$-DP, which is still too much.

Lemma *(Amplification by subsampling).* Let M be an algorithm that is (ε, δ)-differentially private. Let M' be the algorithm that first samples an ρ-fraction of the examples from the dataset uniformly at random, and then applies M to these m examples. Then M' is (ε', δ')-differentially private, for $\varepsilon' \approx \rho\varepsilon$ and $\delta' = \rho\delta$. (Actually, $\varepsilon' = \ln(1 + (e^\varepsilon - 1)\cdot\rho) \approx \rho\varepsilon$ for small ε).

Let $\rho = \frac{m}{n}$ be the fraction of the dataset that we subsample at each step. Then by the above lemma we get that the privacy budget grows as $O\Big(\varepsilon\rho\sqrt{T\log\big(\frac{1}{\delta}\big)}, \delta\rho T\Big)$. Since ρ is typically very small when training neural networks, this is much better than before.

Advanced tools let us get a privacy bound for DP-SGD of roughly $O\big(\varepsilon\rho\sqrt{T}, \delta\big)$ for appropriate choices of parameters, which gives significant savings as we expect very small δ and $T\rho \gg 1$.

13 Membership Inference

Definition *(Confusion Matrix).* We define:

$$\mathrm{TPR} = \Pr[b' = 1 \mid b = 1] \qquad \mathrm{TNR} = \Pr[b' = 0 \mid b = 0]$$
$$\mathrm{FPR} = \Pr[b' = 1 \mid b = 0] \qquad \mathrm{FNR} = \Pr[b' = 0 \mid b = 1]$$

Theorem. Let $\varepsilon > 0$ and $\delta \in [0, 1]$. A mechanism M is (ε, δ)-DP *if and only if* for all adversaries in the membership inference game

$$e^\varepsilon \cdot \mathrm{FNR} \ge \mathrm{TNR} - \delta \quad \wedge \quad e^\varepsilon \cdot \mathrm{FPR} \ge \mathrm{TPR} - \delta$$

Corollary. A mechanism M is (ε, δ)-DP *if and only if* for all adversaries in the membership inference game,

$$e^\varepsilon \ge \max\left(\frac{\mathrm{TNR} - \delta}{\mathrm{FNR}}, \frac{\mathrm{TPR} - \delta}{\mathrm{FPR}}\right)$$

13.a DP Lower-bounds from MI Attacks

We know that if we add Gaussian noise with $\sigma = \tilde{O}\Big(\frac{\sqrt{k}}{n}\Big)$ to each query, we get DP with $\varepsilon = O(1)$. The following theorem shows that this is also necessary.

Theorem. Let M be a Gaussian mechanism for computing the mean of a database $D = \{\vec{x}_1, ..., \vec{x}_n\}$ of n vectors from the domain $\{0, 1\}^k$:

$$M(D) = \frac{1}{n}\sum_{i=1}^n \vec{x}_i + \mathcal{N}\big(\vec{0}, \sigma^2 \cdot I_{k\times k}\big)$$

Assume $\sigma = o\Big(\frac{\sqrt{k}}{n}\Big)$ and $\delta = o(1)$. Then this mechanism is <u>*not*</u> $(O(1), \delta)$-DP.

Using more involved arguments, one can show that *any* mechanism that answers k counting queries needs error $\Omega\Big(\frac{\sqrt{k}}{n}\Big)$ per query to achieve a constant privacy budget ε.

13.b Deep Neural Networks

A deep neural network is a function of linear layers passed through a softmax function σ to obtain a probability distribution:

$$f(x) = \sigma(f_k \circ ... \circ f_1(x))$$

They are trained by gradient descent to minimize a loss function ℓ:

$$\theta_{t+1} \leftarrow \theta_t - \eta\sum_{(x,y)\in B} \nabla_\theta \ell(\theta_t, x, y)$$

These networks commonly overfit hence $\hat{\mathcal{L}}_{D(\theta)} \ll \mathcal{L}(\theta)$.

That means that we can perform the following membership testing attack for a sample x and its correct class y for some τ:

$$\mathcal{A}_{\mathrm{global}(\theta,x,y)} = \begin{cases} 1 \text{ if } \ell(\theta, x, y) \le \tau \\ 0 \text{ otherwise} \end{cases}$$

But this is not a good attack.

13.c Per-Sample Likelihood Ratios

In black-box attacks we only have access to the loss value for a sample, so View $= \ell(\theta, x, y)$.

We can estimate two distributions of the loss over depending on if x was included in the training set by:

$$\mathrm{Loss}_{\mathrm{IN}} = \left\{\ell(\theta, x, y) \middle| \begin{matrix} D_{\mathrm{atk}} \sim \mathcal{P} \\ \theta^{\mathrm{IN}} \leftarrow M(D_{\mathrm{atk}} \cup \{x\}) \end{matrix}\right\}$$

$$\mathrm{Loss}_{\mathrm{OUT}} = \left\{\ell(\theta, x, y) \middle| \begin{matrix} D_{\mathrm{atk}} \sim \mathcal{P} \\ \theta^{\mathrm{IN}} \leftarrow M(D_{\mathrm{atk}}) \end{matrix}\right\}$$

We fit two gaussian distributions to these sets $\mathcal{N}(\mu_{\mathrm{IN}}, \sigma^2_{\mathrm{IN}})$ and $\mathcal{N}(\mu_{\mathrm{OUT}}, \sigma^2_{\mathrm{OUT}})$.

If the observed loss is ℓ the test can now be done by computing:

$$\Lambda(\ell) = \frac{\Pr[\ell \mid H_0]}{\Pr[\ell \mid H_1]} = \frac{\mathcal{N}(\mu_{\mathrm{IN}}, \sigma^2_{\mathrm{IN}})(\ell)}{\mathcal{N}(\mu_{\mathrm{OUT}}, \sigma^2_{\mathrm{OUT}})(\ell)}$$

and accepting if $\Lambda(\ell) \le \tau$ for some treshold τ.

14 Background

Union Bound: $\Pr[\cup_i A_i] \le \sum_i \Pr[A_i]$

Markov's Inequality: For $X \ge 0$ and $a > 0, \Pr[X \ge a] \le \frac{\mathbb{E}[X]}{a}$

Chebyshev's Inequality: For a random variable X with mean μ, variance σ^2, for any $k > 0$, $\Pr[|X - \mu| \ge k\sigma] \le \frac{1}{k^2}$

Chernoff / Hoeffding Bound: Let $X_1, ..., X_n$ be independent random variables taking values in $[a, b]$. Let $X = \sum_i X_i$ and $\mu = \mathbb{E}[X]$. Then:

$$\Pr[|X - \mu| \ge t] \le 2\exp\left(-\frac{2t^2}{n(b-a)^2}\right)$$

$$\Pr\left[\left|\frac{X}{n} - \frac{\mu}{n}\right| \ge t\right] \le 2\exp\left(-\frac{2nt^2}{(b-a)^2}\right)$$

If we remove te absolute value, we can remove the factor 2 in front of the exponential.

Bernoulli: $\mathrm{Ber}(p)$: $\mathbb{E}[X] = p$, $\mathrm{Var}[X] = p(1-p)$.

Binomial: $\mathrm{Bin}(n, p)$: $\Pr[X = k] = \binom{n}{k}p^k(1-p)^{n-k}$, $\mathbb{E}[X] = np$, $\mathrm{Var}[X] = np(1-p)$.