Group Privacy:

Thm: Let $D, D' \in \mathcal{D}$ s.t. $\|D \Delta D'\|_1 = k$.

Let $A: \mathcal{D} \to R$ be ε-DP. Then $\forall S \subseteq R$,

$$\Pr[A(D) \in S] \leq e^{k\varepsilon} \Pr[A(D') \in S]$$

Pf: Induction. Let $D = D_0 \sim D_1 \sim D_2 \sim \cdots \sim D_k = D'$.

Claim: $\forall i$, $\Pr[A(D_i) \in S] \leq e^{i\varepsilon} \Pr[A(D_{i-1}) \in S]$

Pf: Base case: $i = 1$

Def of ε-DP ✓

Inductive step: S_{t-1} true for $i-1$.

$$\Pr[A(D_i) \in S] \leq e^{\varepsilon} \Pr[A(D_{i-1}) \in S] \quad \text{(def of } \varepsilon\text{-DP)}$$

$$\leq e^{\varepsilon} \cdot e^{(i-1)\varepsilon} \Pr[A(D) \in S] \quad \text{(induction)}$$

$$= e^{i\varepsilon} \Pr[A(D) \in S] \quad \checkmark$$

Composition Theorems:

Basic (adaptive) composition:

Q: If we run 2 (or more) DP algorithms, is the combination DP?

Setup: Let $A_1 : \mathcal{D} \to R_1$, ε_1-DP

$A_2 : \mathcal{D} \times R_1 \to R_2$, ε_2-DP for every $r \in R_1$!

(adaptive!)

($A_2(\cdot, r) : \mathcal{D} \to R_2$ is ε-DP $\forall r \in R_1$)

Let $A : \mathcal{D} \to R_1 \times R_2$ be the algorithm that outputs $A(D) = (r_1, r_2)$, where

$r_1 = A_1(D)$, $r_2 = A_2(D, r_1)$

Thm: A is $(\varepsilon_1 + \varepsilon_2)$-DP

Pf: Let $D \sim D' \in \mathcal{D}$

Let $(r_1, r_2) \in R_1 \times R_2$

$$\Pr[A(D) = (r_1, r_2)] = \Pr[A_1(D) = r_1] \cdot \Pr[A_2(D, r_1) = r_2]$$

$$\leq e^{\varepsilon_1} \Pr[A_1(D') = r_1] \cdot e^{\varepsilon_2} \Pr[A_2(D', r_1) = r_2]$$

$$= e^{\varepsilon_1 + \varepsilon_2} \Pr[A(D') = (r_1, r_2)] \quad \checkmark$$

Cor: Let $A_1, \dots, A_k$ algorithms s.t. $A_1 : \mathcal{D} \to R_1$ is ε_1-DP, and $A_i : \mathcal{D} \times R_1 \times R_2 \times \dots \times R_{i-1} \to R_i$ is ε_i-DP. Then $A : \mathcal{D} \to R_1 \times R_2 \times \dots \times R_k$ which runs the algorithms in sequence is $\sum_{i=1}^{k} \varepsilon_i$-DP

Pf: induction, previous thm.

Tight in general!

Interpretation: Can build up larger algorithms from DP steps! ε is "privacy budget", so algorithm runs k $\frac{\varepsilon}{k}$-DP algs, interspersed with computations that don't look at input

$\Rightarrow$ Alg is ε-DP!

Note: means that will often want to think of ε as very small, not just small constant.

In particular, $< \frac{1}{k}$

Ex (Adam Smith / Jon Wellner): Lloyd's algorithm for k-means clustering

Prob: Given $x_1, x_2, \dots, x_n \in \mathbb{R}^d$, find k points $y_1, \dots, y_k$ minimizing $\sum_{i=1}^{n} \min_{j=1}^{k} \|x_i - y_j\|_2^2$

(min sum of squared distances to nearest center)

Lloyd's algorithm: (classical approx alg.

First rescale so that all points in $U = \{x \in \mathbb{R}^d : \|x\|_1 \le 1\}$

- init $c_1^{(0)}, c_2^{(0)}, \dots, c_k^{(0)} \in U$.
- for $t = 1$ to T
  - for $j = 1$ to k
    - $S_j = \{i : c_j^{(t-1)}$ is closest $t-1$ center to $x_i\}$
    - $c_j^{(t)} = \frac{1}{|S_j|} \sum_{i \in S_j} x_i$
- return $c_1^{(T)}, \dots, c_2^{(T)}$

What if we want to make it private?

Neighboring database: one datapoint added/removed.

Vectors in $\mathbb{R}^d$: could use Laplace mechanism on final output.

But what is sensitivity of full algorithm?

Idea: use Laplace + composition!

Steps that use private info:

$$\sum_{i \in S_j} x_i, \quad |S_j|$$

- Set $\varepsilon' = \frac{\varepsilon}{2T}$

- For $i = 1$ to T

  - For $j = 1$ to k

    - $S_j = \{i : c_j^{(t-1)}$ closest to $j\}$

    - $n_j = |S_j|$

    - $q_j = \sum_{i \in S_j} x_i$

    - Release $\hat{n}_j = n_j + Y$, where $Y \sim \mathrm{Lap}(\frac{1}{\varepsilon'})$

    - Release $\hat{c}_j = q_j + (Z_1, \ldots, Z_d)$, where $Z_i = \mathrm{Lap}(\frac{1}{\varepsilon'})$ i.i.d.

    - $c_j^{(t)} = \frac{\hat{c}_j}{\hat{n}_j}$ if $\hat{n}_j \geq 1$

      uniform in $[0,1]^d$ o/w

- Return $c_1^{(T)}, \ldots, c_k^{(T)}$

Thm: This alg is ε-DP.

Pf: By basic composition, if each iteration is $2\varepsilon'$-DP, then overall alg. is $2\varepsilon' T = \varepsilon$-DP.

So consider some iteration, and $x \sim x'$ differing in one point.

$h_1, \dots, h_k$ form histogram.

$\Rightarrow$ adding $\mathrm{Lap}\left(\frac{1}{\varepsilon'}\right)$ noise to each is ε'-DP

$\Rightarrow$ $\hat{h}$'s are ε'-DP.

View $a_1, \dots, a_k$ as one long vector of length kd.

Call this vector A_x for x, $A_{x'}$ for x'

In neighboring databases x, x', one datapoint different

$\Rightarrow$ $\|A_x - A_{x'}\|_1 \le 1$ (since each datapoint has ℓ_1-norm bounded by 1)

$\Rightarrow$ by Laplace mechanism, add noise $\mathrm{Lap}\left(\frac{1}{\varepsilon'}\right)$ to every entry is ε'-DP

$\Rightarrow$ by composition, each iteration is $2\varepsilon'$-DP

$\Rightarrow$ Alg is ε-DP

Advanced Composition:

what if allow (ε, δ)-DP?

- Basic composition still holds: if A_i is $(\varepsilon_i, \delta_i)$-DP, get total alg is $\left(\sum_{i=1}^k \varepsilon_i, \sum_{i=1}^k \delta_i\right)$-DP. (exercise). $(k\varepsilon, k\delta)$ if all are (ε, δ)-DP

- Can do better: basically $\sqrt{k}\cdot\varepsilon$!

Thm: If each A_i is (ε, δ)-DP, then $\forall\ \varepsilon, \delta \geq 0$, $\forall\ \delta' > 0$, the composed alg. A is $(\tilde{\varepsilon}, \tilde{\delta})$-DP, where

$$\tilde{\varepsilon} = \varepsilon\sqrt{2k\ln\left(\tfrac{1}{\delta'}\right)} + k\varepsilon\frac{e^\varepsilon - 1}{e^\varepsilon + 1}, \quad \tilde{\delta} = k\delta + \delta'$$

Set ahead for it before starting proof...

Say have an algorithm which runs k (ε,δ)-DP subalgorithms. Want overall alg to be $(\tilde{\varepsilon}, \tilde{\delta})$-DP. How should we set ε, δ as fn of $\tilde{\varepsilon}, \tilde{\delta}$?

Basic composition: $\varepsilon = \frac{\tilde{\varepsilon}}{k}$, $\delta = \frac{\tilde{\delta}}{k}$

Advanced composition: $\delta' = \tilde{\delta}/2 \Rightarrow \delta = \frac{\tilde{\delta}}{2k}$

For ε: For $\varepsilon < 1$, $\frac{e^{\varepsilon}-1}{e^{\varepsilon}+1} \approx \frac{\varepsilon}{2}$

$$\Rightarrow \tilde{\varepsilon} = \Theta\left(\varepsilon\sqrt{k\ln\left(\frac{1}{\delta}\right)} + k\varepsilon^2\right)$$

$\varepsilon^2 k \leq \tilde{\varepsilon} \Rightarrow \varepsilon \leq \sqrt{\frac{\tilde{\varepsilon}}{k}}$

$\varepsilon\sqrt{k\ln(1/\delta)} \geq \tilde{\varepsilon} \Rightarrow \varepsilon \geq \frac{\tilde{\varepsilon}}{\sqrt{k\ln 1/\delta}}$

$\sqrt{k}$ const to compensate!

Note: δ could also be $\delta' = \tilde{\delta}$

still applies to pure DP mechanisms, but turns them into approximate DP!