Privacy Amplification

- Subsampling
- Shuffling
- Iteration

Amplification by Subsampling

Always run $\underline{S}$GD in practice

Select small mini-batch.

Take two datasets, differ in one entry.

Usually differing point is not used.

Better privacy.

Claim Let A be (ε, δ)-DP. Let A' be the following: on inputs of size n, take uniform subset S with $|S| = m \leq n$, and run $A(x_S)$. Then A' is (ε', δ')-DP for $\varepsilon' = \ln\left(1 + (e^{\varepsilon} - 1)\frac{m}{n}\right)$ & $\delta' = \frac{m}{n}\delta$.

for $\varepsilon, \frac{m}{n}$ small, have $1+(e^{\varepsilon}+1)\frac{m}{n} \approx 1+(1+\varepsilon-1)\frac{m}{n}$

$$\approx 1+\varepsilon\frac{m}{n}$$

$$\ln\left(1+\varepsilon\frac{m}{n}\right) \approx \varepsilon\frac{m}{n}.$$

Math for amplification by subsampling is <u>annoying</u>!

Many variants: subsets, sampling w/replacement, w/o replacement, etc.

Try to give a simple picture.

Analyze randomized response

$$RR_{\varepsilon}(x) = \begin{cases} x & \text{wp } \frac{e^{\varepsilon}}{e^{\varepsilon}+1} \\ \text{flip } x & \text{wp } \frac{1}{e^{\varepsilon}+1} \end{cases}$$

write subsampled version

similar version

$$RR'_{\varepsilon}(x) = \begin{cases} x & \text{wp } \frac{1+\varepsilon}{2+\varepsilon} \\ \text{flip } x & \text{wp } \frac{1}{2+\varepsilon} \end{cases}$$

equivalent to RR':

$$RR''_\varepsilon(x) = \begin{cases} x & \text{wp } \frac{\varepsilon}{2+\varepsilon} \\ \mathrm{Unif}(\{0,1\}) & \text{wp } \frac{2}{2+\varepsilon} \end{cases}$$

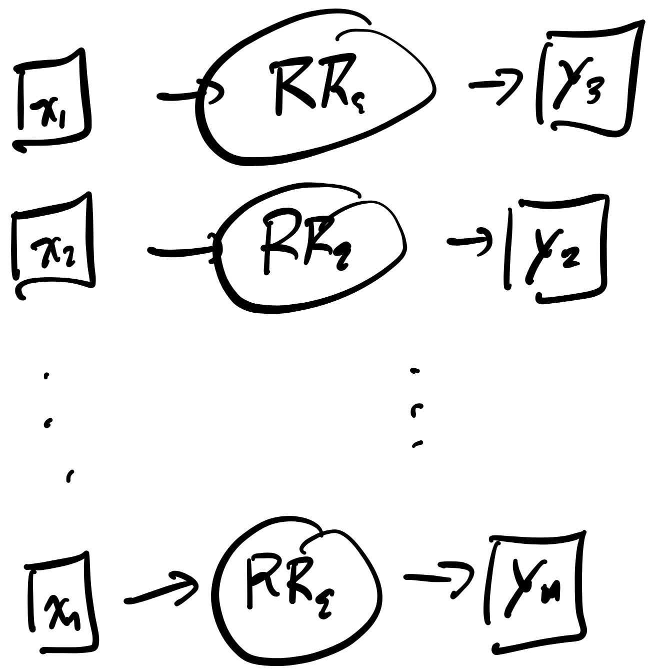
$$\Pr[RR''_\varepsilon(x) = x] = \cdots = \Pr[RR'_\varepsilon(x) = x]$$

$$A(x) = \begin{cases} RR''_\varepsilon(x) & \text{wp } p \\ \mathrm{Unif}(\{0,1\}) & \text{wp } 1-p \end{cases}$$

$$= \begin{cases} x & \text{wp } \frac{\varepsilon p}{2+\varepsilon} \\ \mathrm{Unif}(\{0,1\}) & \text{otherwise} \end{cases} \qquad \approx RR_{\varepsilon p}(x)$$

Shuffling & Local DP

n people running Randomized Response



Example of Local DP, where each individual releases their own noisy version of data.

Usually: less trust, less accuracy.

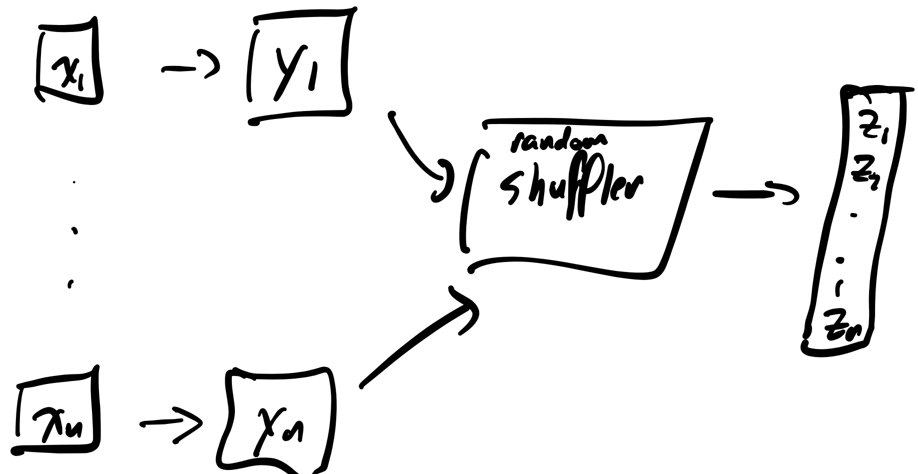
Compare: with Laplace noise, can estimate

$$\hat{u} = \sum_{i=1}^{n} x_i \pm O\left(\frac{1}{\varepsilon}\right)$$

Informally, with RR_ε get $\hat{u} = \sum_{i=1}^{n} x_i \pm O\left(\frac{\sqrt{n}}{\varepsilon}\right)$

!! ⌢

Under LDP, adversary knows to focus on output y_i.
What if they don't know which one was y_i?



Does this give better privacy? Yes!

<u>Theorem (Informal)</u> If each user runs RR_ε with $\varepsilon \le 1$, then $\forall \delta > 0$ the shuffled protocol is (ε', δ)-DP with $\varepsilon' = O\left(\varepsilon \frac{\sqrt{\log 1/\delta}}{\sqrt{n}}\right)$.

Kills $\sqrt{n}$ error term!

Intermediate trust assumption, shuffler is very simple.

Amplification by Iteration

(Actual math is complicated to state.)

Consider running some DP-SGD variant in a one-pass setting, so each data point is seen only once.

For concreteness, maybe it's like:

$A(x)$

① For $t = 1, \ldots, T$

② $\quad \tilde{g}_t \leftarrow \frac{1}{b} \sum_{i=tb}^{tb+b} \nabla \ell(\theta; x_i, y_i) + \mathcal{N}(0, \sigma^2 \mathbb{I})$

③ $\quad \theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$

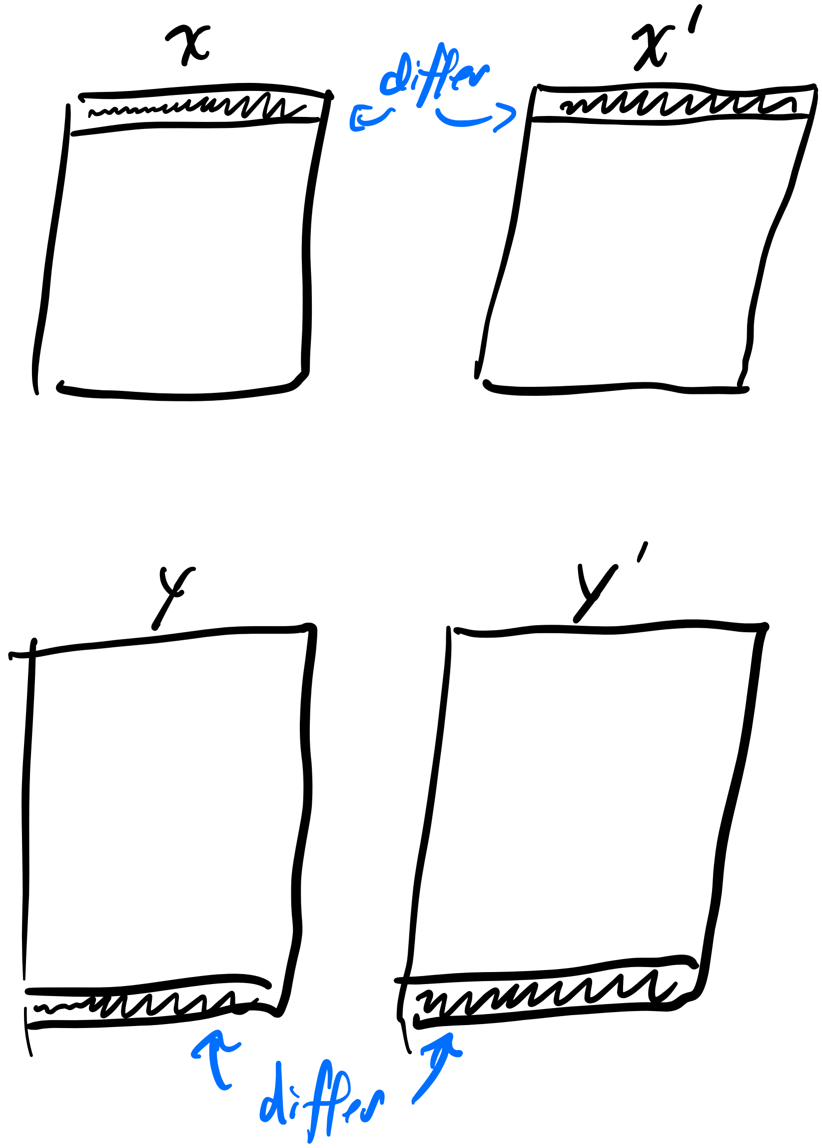
④ End For

⑤ Return θ_T

With $n = Tb$ examples overall and batch size b.

Also note this algorithm only returns the final iterate.

Now consider four datasets: x, x', y, y'



ie, we have $x \sim x'$ and $y \sim y'$

Intuition: It should be harder to distinguish $\boxed{A(x) \text{ vs } A(x')}$ compared to distinguishing $\boxed{A(y) \text{ vs } A(y')}$ because the last elements should be more influential on θ_T.

In other words, the extra iterations might buy some other privacy in the setting of x vs x'.

How could we use this?

One example would be to add less noise earlier in the optimization process.