


---

---

---

---

---



# DP Review

- Defn:  $\forall$  neighboring  $D, D'$ ,  
 $\forall$  set  $S$  of outputs:

$$\Pr[A(D) \in S] \leq e^\epsilon \Pr[A(D') \in S]$$

- Randomized Response:

$$\ln\left(\frac{1+p}{1-p}\right) - DP$$

- Laplace: Output  $f(\bar{x}) + V$

$\epsilon$ -DP, accuracy  $\sim \Delta f / \epsilon$   $\propto \frac{1}{2b} e^{-1V/\epsilon} \frac{-1V/\epsilon}{\Delta f}$

- Exponential: Output  $y \propto e^{\epsilon u(x,y)/\Delta u}$

$\epsilon$ -DP, within  $\sim \frac{\Delta u}{\epsilon}$  of optimal  
(may be hard to sample)

# Nice Application of Exp. Mech. : synthetic data

---

- Let  $D$  be some data set
- E.g. rows with numeric vars/cols  $x_1, x_2, \dots, x_d$
- Pick some stats we'd like to preserve, e.g. avgs of  $x_i$ , correlations of  $x_i, x_j$
- Output of algo will be another dataset  $\hat{D}$

- Now define  $u(D, \hat{D})$   
to measure how well  
 $\hat{D}$  agrees with  $D$   
on preserved stats

• E.g.

$$u(D, \hat{D}) = -\max_i \left\{ \underbrace{|u(x_i)|}_D - \underbrace{\hat{u}(x_i)}_{\hat{D}} \right\}$$

- Draw  $\hat{D}$  from Exp. Mech  
using  $D, \hat{D} \Rightarrow \epsilon$ -DP
- Now publish  $\hat{D}$ !

So we have some good  
general primitives or tools

What about methods for  
combining them to  
obtain richer  
DP algos?

Would like programmability  
for DP.

# Method 1: Post-Processing

- Let  $A(D_1)$  be an  $\epsilon$ -DP algorithm
- Let  $B(A(D_1), D_2, D_3, \dots, D_k)$  be an algo taking  $A(D_1)$  as input, as well as other dataset
- Then  $B$  is  $\epsilon$ -DP in  $D_1$
- DP cannot be "undone", even by a non-DP algo

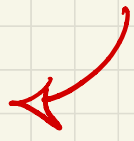
## Method 2: Repetition

- Let  $A_1, A_2, \dots, A_\ell$  be  $\epsilon$ -DP algos
- Then  $B(D) = (A_1(D), \dots, A_\ell(D))$  is  $\ell \cdot \epsilon$ -DP
- $\ell \cdot \epsilon$  is privacy loss from multiple DP computations
- More general: if  $A_i$  is  $\epsilon_i$ -DP, then  $B$  is  $\sum_i \epsilon_i$ -DP

# A Better Repetition Bound

- Say  $A$  is  $(\epsilon, \delta)$ -DP if

$$\Pr[A(D) \in S] \leq e^\epsilon \Pr[A(D') \in S] + \delta$$

If  $\Pr[A(D') \in S] \ll \delta$ ,   
This is costing a lot!

- Then  $A_1, \dots, A_\ell$   $\epsilon$ -DP  $\Rightarrow$

$B(D) = (A_1(D), \dots, A_\ell(D))$  is  
 $(\epsilon', \delta)$ -DP where

$$\epsilon' \sim \sqrt{k \log(1/\delta)} \cdot \epsilon$$



## Method 3: General Composition

Even better: same kind  
of  $(\sqrt{k \log(1/\epsilon)} \epsilon, \delta)$ -DP  
result holds even if

$B$  an adaptive sequence  
of compositions of  
 $\epsilon$ -DP algos!

- True programmability

- Even less privacy loss  
for structured problems

# (Research) Applications of DP

# Application: Machine Learning

- Basically **any** ML method that learns models in a **"statistical"** fashion can be **made DP**
- Linear/logistic regression, decision trees, boosting, neural networks, reinforcement learning, PCA, clustering...
- Not covered: "equation-solving" methods

# Application: Game Theory & Mechanism Design

- Major concern: incentivizing truthfulness
- E.g. bids in an auction (second price, VCG); origins & destinations in Waze
- If you can design your mechanism to be DP, get (approx.) truthfulness for free

# Application: P-hacking & the reproducibility Crisis

---

- Problem: (common) **overfitting** on (shared) datasets
- E.g. CIFAR competitions
- Potential solution: DP leaderboards
- E.g. only publish improvements larger than 1%, can only happen  $\leq 100$  times

# Real-World Deployments

- Apple OS
- Google Chrome  
& open source DP
- 2020 U.S. Census
- COVID-19