Lecture 15: Differential Privacy

MIT - 6.893
Fall 2020
Henry Corrigan-Gibbs

Plan

* Recap: Private Aggregation

* Privacy Problems

* DP Defn

* Laplace Mechanism

* Issues in practice

Logistics

* HW5 out now
  due Nov 13 @ 5pm

* Granick visit on
  Wednesday

PLEASE DO READINGS AND BRING Qs!

Should be great.

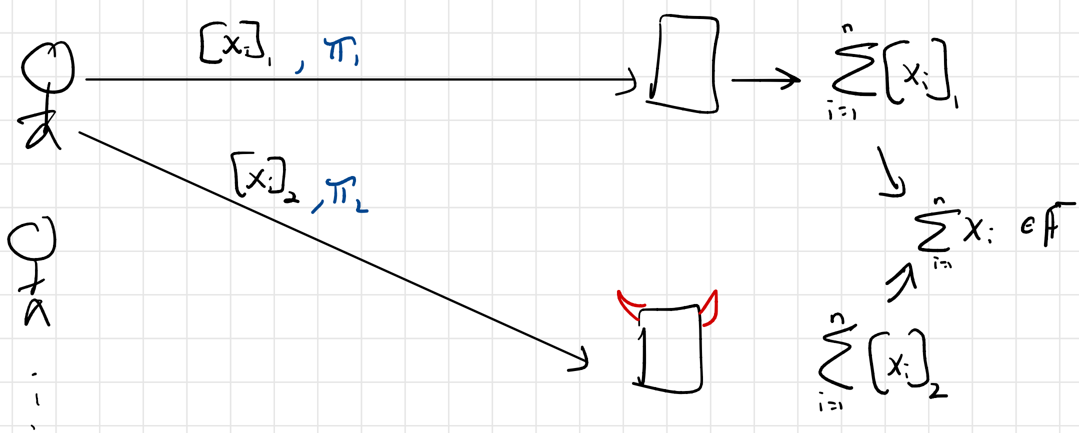
Recap: Private Aggregation

Many clients with values $x_1, \ldots, x_n \in \mathbb{F}$

Servers want to know $\sum_{i=1}^{n} x_i \in \mathbb{F}$.

↳ Covers surprisingly many applications
Boston wage gap, telemetry, ...

Simple protocol ... "Degenerate MPC"



Provided that ≥ 1 server is honest,
Servers learn $\sum_{i=1}^{n} x_i$ and "nothing else"

↳ When you are worried about clients corrupting output
can use ZK proof on secret-shared data to have
client prove that its value satisfies a "validity predicate"
e.g. $\in \{0, 1, \ldots, 10\}$.

↳ Correctness only holds if both servers are honest...

Differential Privacy

In the private-aggregation system we just saw, as long as ≥ 1 server honest, servers "learn nothing more" than sum of clients' inputs.

Q: Do the servers still learn too much?

Example: Use private-agg system for private survey of MIT 1st-yrs
"Have you ever broken Covid rules?"

n students, inputs $x_1, \dots, x_n \in \{0,1\}$

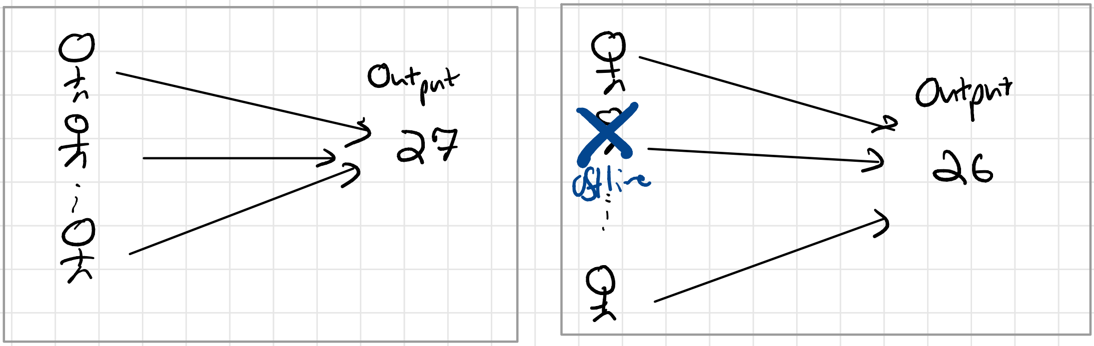
output $\sum_{i=1}^{n} x_i = n$

There's only one possible choice of inputs that explains this output! Output reveals all inputs!

What happened? Something went wrong here. But system worked as intended.

A more realistic example...

Private agg system used every day to answer MIT survey Q



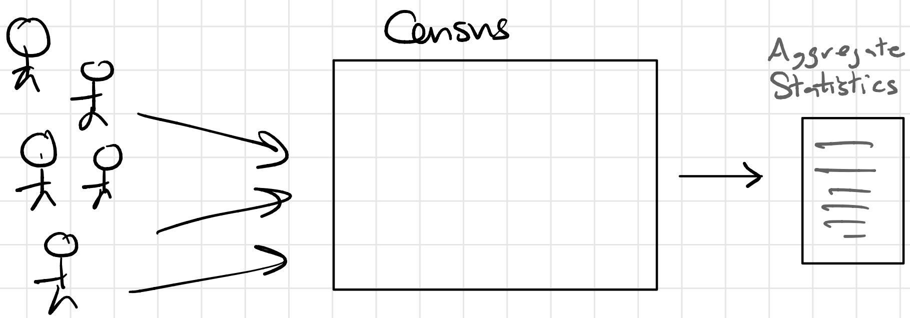
Day 0 Day 1

→ Under reasonable assumption about stability of data, servers (or anyone who sees output) learns one client's private value.

Again, something seems wrong...
But what?

Another example: U.S. Census

"Best possible privacy" (assuming you trust the Census Bureau)



(E.g. From Michael Hawes talk on D.P in census)

Public Data

Name	Age	Zip
...	32	02139

Census Data

Age	Zip	# children
32	02139	___

On its own, the census data isn't problematic, but when combined with other data (side info) it is.

Another Example: AOL Data Set

In Aug 2006, AOL published a data set of search queries

ID	Query
9243	Cheese store Somerville
9243	Bike trails near Somerville
9243	Private information retrieval
9243	???

→ Stripping names meaningless ... in many (all?) cases it is trivial to identify a person from their search queries.

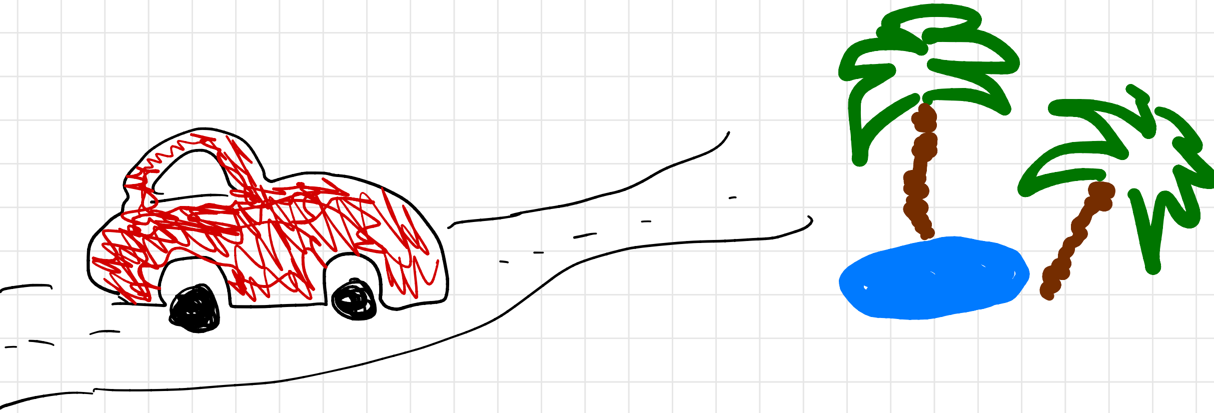
In each of these cases, a protocol or system or person published some data that was "harmful" to privacy.

Cryptographic protocols (e.g. MPC) typically focus on "the HOW" – once you decide which fn you want to publish, how do you do this without leaking anything else or trusting a central server or ....

Differential privacy focuses on "the WHAT."
What fns of private data are safe to output?

↳ Irrespective of how we accomplish this!

Analogy (from Omer Reingold)



MPC is the mechanism (the car)

"How do n parties compute function $f(x_1, \ldots, x_n)$ of their private data while "leaking" nothing else?"

D.P. is about the goal (the destination)

"Is $f(\cdot)$ a safe fn to compute?"

Breakout Rooms

Q: How would you reason about which fns of private data are safe to relesy?

A definition?

Two parts to the study of DP

(You'll often hear these conflated/confused)

1. Definitions

* Very robust principled way to capture badness or leakage as the result of revealing fns of private data.

↪ Almost obvious in retrospect (in the best way)

2. Mechanisms

* Technical means to construct systems that publish data while respecting DP defn's.

(See Dwork–Roth Book for lots of useful background and advanced tools.)

The Bad News :(

With much of crypto, we have our cake and eat it too... not so here.

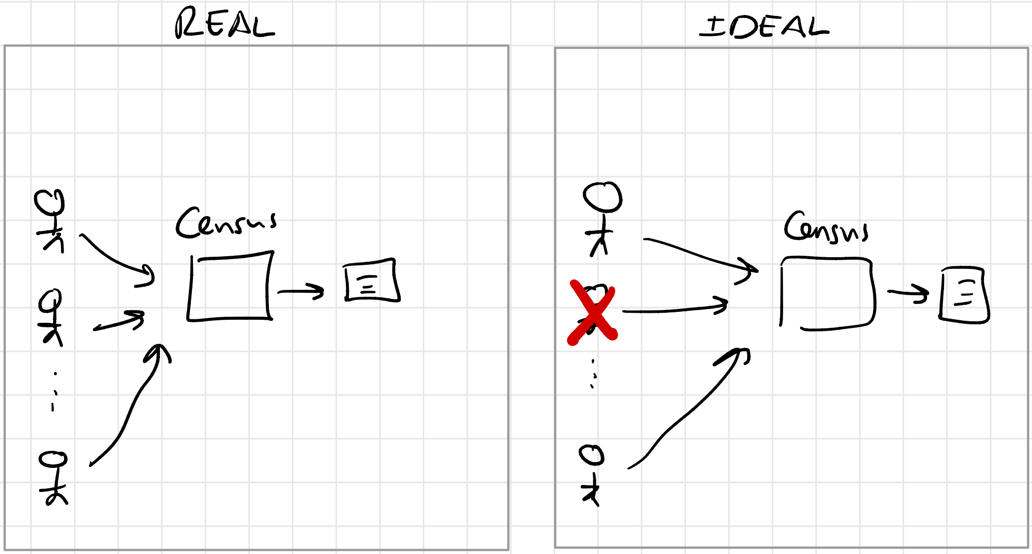
More utility

More privacy

There is an explicit trade-off... can max out both at the same time.

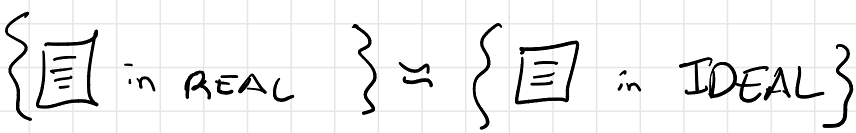
Privacy parameter ε captures this trade-off.

Idea of D.P. Def'n



Your data not included in census results

Would like that...



In D.P. these worlds are "similar" but NOT cryptographically indistinguishable. (in some sense).

↳ If they were, no point contributing your data at all!

(Dwork, McSherry, Nissim, Smith 2006)

Formalism

Mechanism $\mathcal{M} : \underbrace{\mathcal{X}^n}_{n\text{-row DB}} \longrightarrow \underbrace{\mathcal{Y}}_{\text{output}}$

(e.g. $\mathcal{X} = \{0,1\}$ and $\mathcal{M}$ outputs sum of values)

Two DBs D, D' are "neighboring" if they differ in at most one row.

$$\boxed{\begin{array}{l}\text{Mechanism satisfies } \varepsilon\text{-DP if } \forall \text{ pairs of "neighboring databases" } D, D' \text{ and} \\ \text{every set of values } S \subseteq \mathcal{Y}, \\ \qquad \Pr_{\mathcal{M}}[\mathcal{M}(D) \in S] \le e^{\varepsilon} \cdot \Pr_{\mathcal{M}}[\mathcal{M}(D') \in S].\end{array}}$$

Typically, take $\varepsilon =$ small constant. (0.1, 1, 5)

Intuition about Defn

Say that some outputs of mechanism are bad for you (e.g. grad student salary). If bad output was going to happen w.p. $\le p$ if you don't participate, then participating increases chance of bad event to $\le p \cdot e^{\varepsilon}$.

$\Rightarrow$ When $\varepsilon > 10$, the guarantees start to really break down.

DP is a strong notion of privacy:

* for <u>all</u> pairs of DBs (worst case)
* no computational limits
* no small chance of failure ...

Mechanism can fail to satisfy DP b/c there exists a single non-realistic pair of DBs that give very diff outputs

Robust!

If M is ε-DP, $F \circ M$ is ε-DP

↳ Post processing

If M_1 is ε_1-DP, M_2 is ε_2-DP, → Composition

$(M_1 \| M_2)$ is $(\varepsilon_1 + \varepsilon_2)$-DP.

If DBs differ at k rows & M is ε-DP, outputs on differing DBs are $k\varepsilon$-DP → Group privacy

Sanity Check: Mechanism that Outputs Sum is not DP for $\varepsilon < \infty$?

Take $D = \{0, 0, \dots, 0\}$
$D' = \{0, 0, \dots, 1\}$ neighboring

$S = \{0\}$

$$\underbrace{\Pr[M(D) \in S]}_{1} \leq e^{\varepsilon} \cdot \underbrace{\Pr[M(D') \in S]}_{0}$$

$\Rightarrow$ Cannot satisfy D.P. with this mechanism.

Laplace Mechanism ... 98% of DP you will encounter

A way to output any fn $f: \mathcal{X}^n \to \mathcal{Y}$ with D.P.

For simplicity, say $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i$

$x_i \in \{0, 1\}$

Laplace Mechanism for sum

$$M(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i + \underbrace{Lap(1/\varepsilon)}_{\text{noise}}$$

Good: Now can satisfy ε-DP

surprise that it's SAT at all

mean: 0

variance: $2/\varepsilon^2$

$Lap(1/\varepsilon)$ PDF: $\frac{\varepsilon}{2} e^{-\varepsilon \cdot |x|}$

"Heavy tailed distribution"

Bad: Noisy answer

... expect error $\approx 1/\varepsilon$.

Noise is inherent (see sum example)

As $\varepsilon \to 0$, noise $\to$ BIG

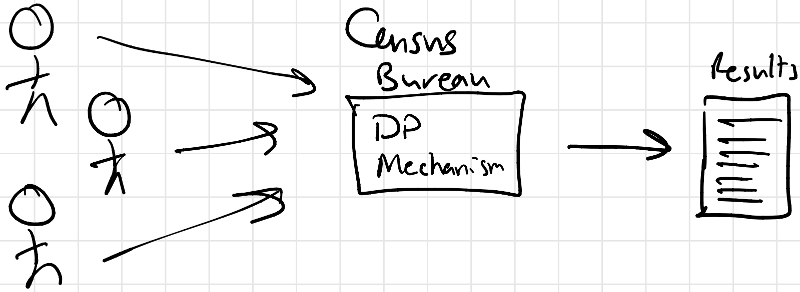
When reading about apps of DP,
the two questions you should ask are:

1) What is ε? Over time?

2) What are "neighboring" DBs
in this setting?
(e.g. cell phone data)
calls vs #s vs users.

3) Local or central?
↳ More about use than defn

Central Model (Think: U.S. Census)

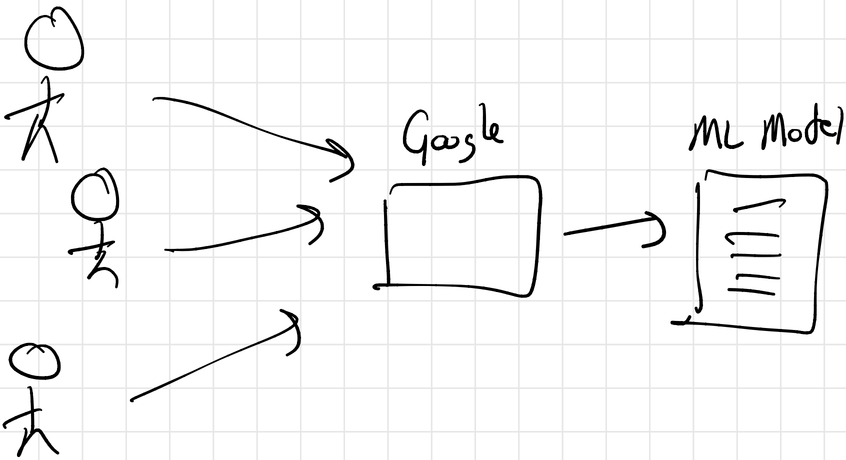


Pro: + Easy to implement (?)

+ Fewer changes to existing processes

Con: − Census sees all of your data

↪ No privacy w.r.t. census



Difficulties using DP in practice

→ As you release more statistics, effective $\varepsilon \to$ BIG (sums up). Very quickly, the privacy guarantee becomes vacuous.

→ No good way to "reset" privacy budget

→ Non-sensical outputs. E.g. in census, cities w/ negative population.

→ Data consistency: Need marginals to add up, etc.

→ Analyzing complex mechanisms (e.g. ML training) is very difficult.

→ What is the right value of ε?

Take away:

DP is one powerful and important defn of privacy.
It doesn't solve all of our problems.
It doesn't always perfectly capture true privacy leakage.
But it is the best we have so far.