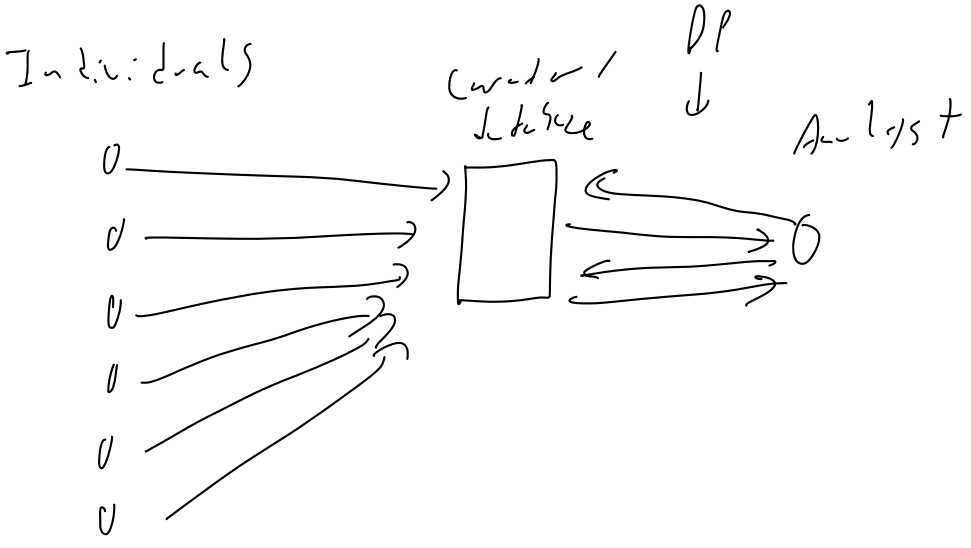
Local DP:

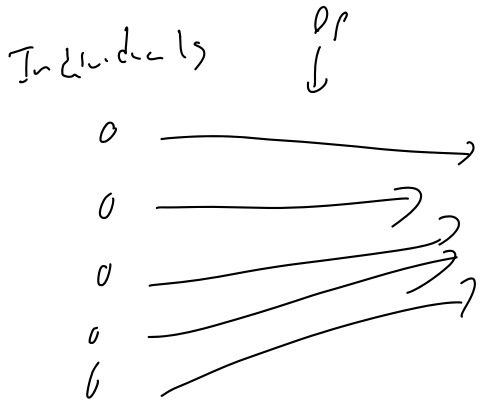
So far: centralized DP. "Trusted curator".



Analyst interacts w/ curator to preserve privacy, but curator knows all true values!

- what if don't trust curator to have your data?
- what if curator gets hacked?

Local DP: don't even trust curator!



Formal def: "local randomizer", "local random coins", etc.

Completely equivalent to every individual using DP on dataset of size 1!

Def: If every individual uses an (ε, δ)-DP alg, then combined alg/mechanism/protocol is (ε, δ)-local DP

Thm: (ε, δ)-local DP $\Rightarrow$ (ε, δ)-DP

Pf: Parallel Composition!

Variants: - interactive vs. non-interactive

Pros: - super distributed, no trust, used in practice!

Cons: - often much worse accuracy!

Easy example: sum of binary values

- Each $x_i \in \{0,1\}$, return $X = \sum_{i=1}^{n} x_i$
- Central DP: $\Delta f = 1 \Rightarrow \tilde{X} = X + \mathrm{Lap}(1/\varepsilon)$: ε-DP

$$\mathbb{E}[|X - \tilde{X}|] = O(1/\varepsilon)$$

- Local DP: only possible algorithm is randomized response!
- RR: outputs $\tilde{x}_i = \begin{cases} x_i & \text{w/prob. } \frac{e^\varepsilon}{e^\varepsilon + 1} \\ 1 - x_i & \text{w/prob } \frac{1}{e^\varepsilon + 1} \end{cases}$

Claim: ε-local DP.

Pf: Consider user i.

$$\frac{\Pr[\tilde{x}_i = 1 \mid x_i = 1]}{\Pr[\tilde{x}_i = 1 \mid x_i = 0]} = \frac{\frac{e^\varepsilon}{e^\varepsilon + 1}}{\frac{1}{e^\varepsilon + 1}} = e^\varepsilon$$

$$\frac{\Pr[\tilde{x}_i = 0 \mid x_i = 0]}{\Pr[\tilde{x}_i = 0 \mid x_i = 1]} = e^\varepsilon \quad \checkmark$$

Given $\tilde{x}_i$'s, what should (untrusted) curator do?

Want

$$\mathbb{E}\left(\sum_{i=1}^n \tilde{x}_i\right) = \sum_{i : x_i = 1} \frac{e^\varepsilon}{e^\varepsilon + 1} + \sum_{i : x_i = 0} \frac{1}{e^\varepsilon + 1}$$

$$\Rightarrow \left(e^\varepsilon + 1\right) \mathbb{E}\left[\sum_{i=1}^n \tilde{x}_i\right] = \sum_{i : x_i = 1} e^\varepsilon + \sum_{i : x_i = 0} 1$$

$$= \sum_{i=1}^n 1 + \sum_{i : x_i = 1} (e^\varepsilon - 1)$$

$$= n + \sum_{i : x_i = 1} (e^\varepsilon - 1)$$

$$\Rightarrow \frac{\left(e^\varepsilon + 1\right) \mathbb{E}\left(\sum_{i=1}^n \tilde{x}_i\right) - n}{e^\varepsilon - 1} = \sum_{i : x_i = 1} 1$$

$$\Rightarrow \mathbb{E}\left[\sum_{i=1}^n \left(\frac{e^\varepsilon + 1}{e^\varepsilon - 1} \tilde{x}_i - \frac{1}{e^\varepsilon - 1}\right)\right] = \sum_{i : x_i = 1} 1 = \#1\text{'s}.$$

So curator computes

How good is this?

Let $X = \#1$'s, $\tilde{X} = \sum_{i=1}^{n} \left(\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1} \tilde{x}_i - \frac{1}{e^{\varepsilon}-1} \right)$

Variance: $E[(X-\tilde{X})^2] \approx \frac{n}{\varepsilon^2}$ (when ε small) $=\sigma^2$

$\Rightarrow E[|X-\tilde{X}|] \geq \frac{\sqrt{n}}{\varepsilon}$, tight via Chebyshev/Chernoff

So constant expected loss in central, $\sqrt{n}$ in local!

Private Selection:

Each $x_i \in [d]$, return $\underset{j \in [d]}{\text{argmax}} \left(\sum_{i : x_i = j} 1 \right)$

Central: RNM: add $\text{Lap}(1/\varepsilon)$ to each count, return max.

works if diff b/ max, second $\geq \frac{\log d}{\varepsilon}$

Local: write each el't of $[d]$ as $\{0,1\}^{\log d}$, run

RR on each of $\log n$ bits.

Composition: set $\varepsilon' = \frac{\varepsilon}{\log d}$

$\Rightarrow$ expected error of each bit $\approx \frac{\sqrt{n} \log d}{\varepsilon}$

$\Rightarrow$ need every bit to have $\geq \frac{\sqrt{n} \log d}{\varepsilon}$ more of correct value.

Could be pretty unlikely even if lots more of correct val.!

Ex: $d = 4$ 00, 01, 10, 11 $\frac{4}{10}$ 10 $\frac{3}{10}$ 01 $\frac{3}{10}$ 11

$\Rightarrow$ final bit will be set to 1 incorrectly!

- "One-hot" encoding: each elt of $[d]$ written as $\{0,1\}^d$

  $x \in d \rightarrow \{0, 0, \ldots 0, 1, 0, \ldots 0\}$ (the 1 at position x)

RR on each bit: don't need composition! Like RNM

$\Rightarrow$ expected error of each bit $\approx \frac{\sqrt{n}}{\varepsilon}$

$\Rightarrow$ ok if diff b/w largest, second largest $\geq \frac{\sqrt{n} \log d}{\varepsilon}$

More thoughts:

- Connections to "statistical query algs"
- Used significantly in industry!
- Google: RAPPOR
  - "Permanent" RR
  - Bloom filter to map each $x \in X \rightarrow \{0,1\}^k$
  - Issue: assumes value never changes!
- Apple: based on "Count sketch" alg. to reduce comm. cost
- Microsoft:
  - more tricks to reduce comm. cost
  - what if values change over time, but not by much?

- e.g. how long spent in app for day
- discretize!
  - too fine: too much re-randomization
  - too coarse: lose utility
- Main result: 1-bit mean estimation w/ coarse buckets, no real loss in accuracy!

- In graphs, LEDP pretty powerful!
  - one edge changes ⇒ ~~two~~ nodes have diff. local input
  - but not arbitrarily different!
  - Interactivity ~~super~~ helpful.
  - Not just RR!

Beyond Local:

- Crypto: replace trusted curator by secure Multiparty Computation! (MPC)
  - basically works, but need all users to be available to run protocol
  - computational assumptions, PKI (public key infra)

Next time: "Shuffle model"