PDS, 24.11.21

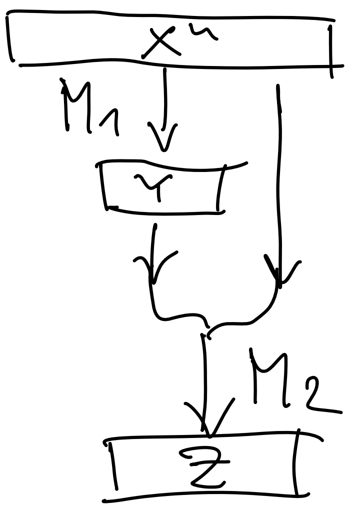
[6.4]

a) Postprocessing preserves D.P.

b) Sequential composition

Let $M_1 : \mathcal{X}^n \longrightarrow \mathcal{Y}$

$M_2 : \mathcal{X}^n \times \mathcal{Y} \longrightarrow \mathcal{Z}$



Thm: If M_1 is ε_1-dp. and M_2 is ε_2-dp. then $M_2 \circ M_1$ is $(\varepsilon_1+\varepsilon_2)$-dp.

Pf: For $z \in \mathcal{Z}$, any X^n and $\bar{X}^n$ s.t. $X^n \sim \bar{X}^n$

$$\mathbb{P}\left[M_2(M_1(X^n), X^n) = z\right]$$

$$= \sum_y \underbrace{\mathbb{P}\left[M_2(y, X^n) = z\right]}_{\varepsilon_2\text{-d.p.}} \cdot \underbrace{\mathbb{P}\left[M_1(X^n) = y\right]}_{\varepsilon_1\text{-d.p.}}$$

$$\leq \sum_y e^{\varepsilon_2} \mathbb{P}\left[M_2(y, \bar{X}^n) = z\right] \cdot e^{\varepsilon_1} \mathbb{P}\left[M_1(\bar{X}^n) = y\right]$$

$$= e^{\varepsilon_1 + \varepsilon_2} \cdot \mathbb{P}\left[M_2(M_1(\bar{X}^n), \bar{X}^n) = z\right]$$

For k different, but always ε-d.p. algorithms $M_1, \dots, M_k$, the composition is $k\varepsilon$-d.p.

$\Rightarrow$ privacy budget

c) Group privacy

What if b positions change from X^n to $\bar{X}^n$?

$$X^n = X_0^n \sim X_1^n \sim X_2^n \dots \sim X_b^n = \bar{X}^n$$

There exists a sequence of neighboring vectors.

Thm: For ε-d.p. M, let X^n and $\bar{X}^n$ differ in b entries. Then for all $Y \subseteq \mathcal{Y}$ it holds

$$P[M(X^n) \in Y] \leq e^{b \cdot \varepsilon} P[M(\bar{X}^n) \in Y]$$

6.5) D.P. and private machine learning

- How does d.p. data influence learning?
- How does ML on d.p. data impact the privacy of data?

ML extracts statistical evidence from a dataset, even if dataset is d.p.

Ex. Medical study considers $\underbrace{\text{attributes}}_{QI}$ of patients and $\underbrace{\text{diseases}}_{S}$.

Study reveals correlation between QI of smoking and S of lung cancer.

Ex. Dataset

QI	3	-2	9	2.71828	-6
S	4	-1	10	3.71828	-5

ML learns that "$S = QI + 1$"

ML predicts that for $QI = 2.71828$,
$S = 3.71828 \ldots$?

ML alg. does not violate privacy.
But revealing the datapoint 2.71...
in question and in dataset as QI
was a privacy violation.

Ex. Predict pregnancy before person is
aware of it.

Ex. Netflix dataset: not made properly
private and dataset itself
violated d.p.

6.6) D.P. in practice

- Today used by many companies online
- Especially Google, MSFT, Apple

Practical concerns

1) Obtain setting, environment values, keywords for analytics

   $\Longrightarrow$ Bitstrings, char. strings are not numerical

2) Values change little over time

   $\Longrightarrow$ simpl d.p. collection would reveal too much

3) Collect data efficiently

   $\Longrightarrow$ Encoding methods

Model

Local D.P.

Users

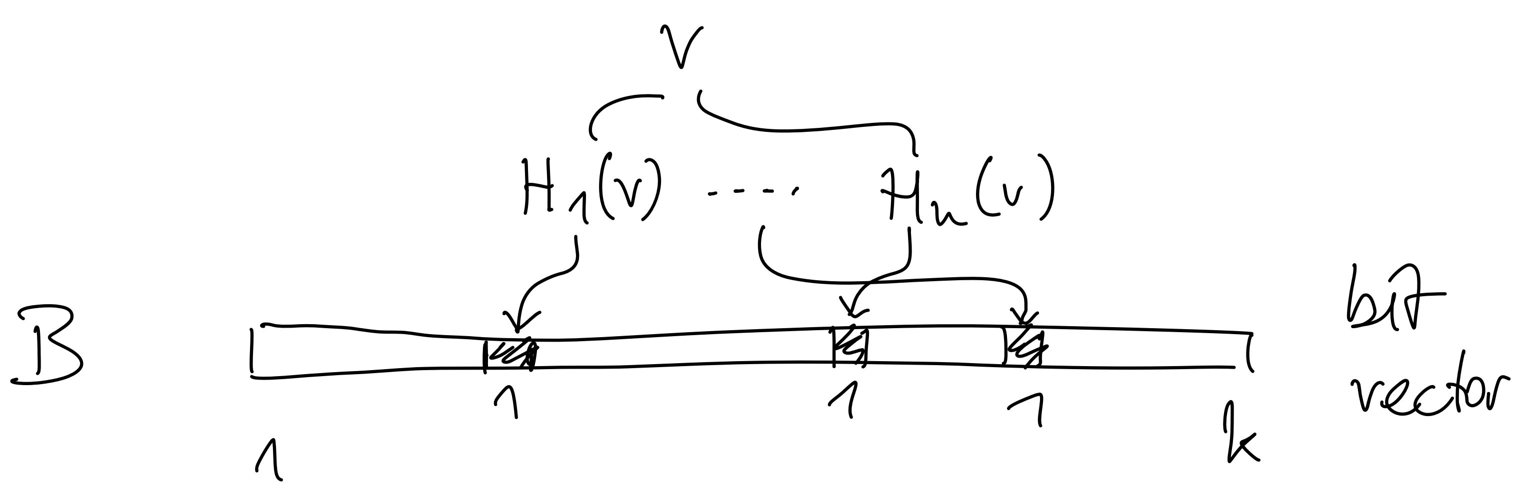
$$\begin{tikzcd} X_1 \arrow[d, "M"] & \cdots & X_n \arrow[d, "M"] \\ Y_1 & \cdots & Y_n \end{tikzcd}$$

$$\underbrace{Y_1 \quad \cdots \quad Y_n}_{} \xrightarrow{M'} Z$$

Central collector

Solution to 1: Bloom filters

Turn string (or a number) into a <u>log-k-bit</u> string using h different hash functions, $H_1, \dots, H_h$.



For each $\ell = 1, \dots, h$: set bit ℓ in B to 1.

Bloom filter may contain false positives, but if some $\tilde{v}$ was stored in B, then B always reports that it contains $\tilde{v}$.

* B is a bit vector, to which one can apply randomized response.

* Since B changes not often, it cannot be sent repeatedly many times.
  (If one would randomize it independently, this noise would be filtered using statistics.)

Solution to 2: Memoization

- Do not run ε-d.p. local M for each report.
  But compute once
  B' using M_L a local d.p. mech.

Then report value, send

B'' computed using M_L

from B'.

Then collector applies statistics.

B' is called memoized

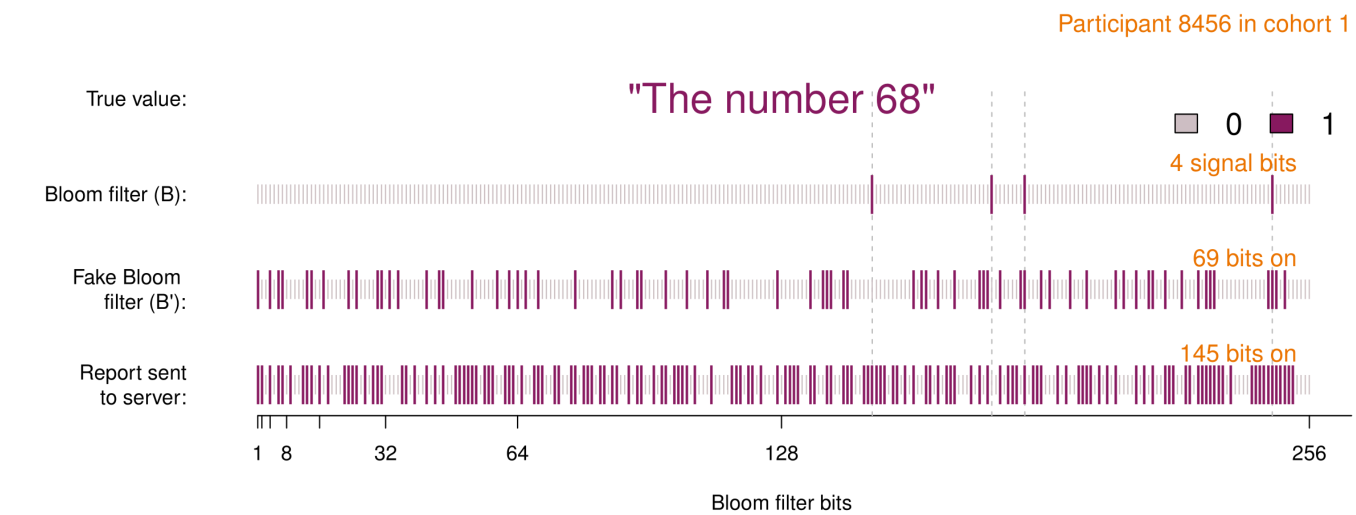


Figure 1: Life of a RAPPOR report: The client value of the string "The number 68" is hashed onto the Bloom filter B using h (here 4) hash functions. For this string, a Permanent randomized response B' is produces and memoized by the client, and this B' is used (and reused in the future) to generate Instantaneous randomized responses S (the bottom row), which are sent to the collecting service.

from RAPPOR paper. [EPK14]

↳ by Google, deployed in Chrome

- daily reports .. up to 30 min
- 100 metrics, each is 2-d.p.

- repeatedly collected, until budget of ~4.4 is exhausted

- Collecting from 14M clients, reveals a value only if shared by 14'000 clients

Solution to 3: efficient data collection

Each user reports $X_i \in [0, m]$,

for $i = 1, \ldots, n$

Local Lap. mech.

$$Y_i = X_i + \mathrm{Lap}\left(\frac{m}{\varepsilon}\right)$$

sending one bit $Y_i \in \{0, 1\}$

$$Y_i = \begin{cases} 1 & \text{w/prob. } \frac{1}{e^\varepsilon + 1} \cdot \frac{X_i}{m} \left(\frac{e^\varepsilon - 1}{e^\varepsilon + 1}\right) \\ 0 & \text{otherwise} \end{cases}$$

by Microsoft in Windows (≥ 10)

... for details, see paper [BKY17].