Introduction to Differential Privacy

Björn Bebensee
bebensee@bi.snu.ac.kr

Seoul National University

November 14, 2019

Contents

Motivation

Imagine we have a database of users. We want to publish some anonymous statistics about the users.

Motivation

Imagine we have a database of users. We want to publish some anonymous statistics about the users.

This is a hard problem!

Famously researchers uniquely identified 99% of users in the "anonymized" Netflix Prize dataset in 2007.

Motivation

Given: a database of users

Compute: statistics over all users

Constraint: protect the information of each single user

Adversary: Potentially malicious actor who either

- is able to query the *statistical database* (*interactive setting*)
- only has access to the published statistics (*noninteractive setting*)

Differential Privacy

Differential privacy is a framework of statistical techniques.
It is used to

(1) compute statistical queries on user inputs
(2) protect each individuals' privacy while doing so

Allows a trade-off between utility and user privacy.

Differential Privacy

Definition

A randomized function $\mathcal{K}$ gives ϵ-Differential Privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,

$$\frac{\Pr[\mathcal{K}(D_1) \in S]}{\Pr[\mathcal{K}(D_2) \in S]} \leq e^{\epsilon}$$

An algorithm is ϵ-DP if it does not depend on any single entry in the dataset but rather (probably) gives the same output even if you remove any single entry.

Differential Privacy

How can we achieve ϵ-Differential Privacy?

Differential Privacy

Idea: Inject random noise to protect the privacy of individuals.

Example: What are the number of users with a bad credit rating?

Ground truth $N = 21$, X random variable with Laplace distribution. Return $N + X$.

Differential Privacy

Idea: Inject random noise to protect the privacy of individuals.

Example: What are the number of users with a bad credit rating?

Ground truth $N = 21$, X random variable with Laplace distribution. Return $N + X$.

Problem

Privacy losses accumulate. An adversary can estimate the ground truth given enough queries. More queries correspond to laxer privacy guarantees.

Differential Privacy

To address this define a function's sensitivity as follows:

Definition

For a function $f : D \to R^k$, the sensitivity of f is

$$\Delta f = \max_{D_1, D_2} ||f(D_1) - f(D_2)||_1$$

for all datasets D_1, D_2 differing in at most one element.

Then, for a query function f, return

$$f(x) + \text{Lap}(\Delta f/\epsilon)$$

with *privacy loss* ϵ, i.e. the variance depends on the sensitivity and the privacy loss.

Problems

However, there are some problems with (centralized) DP:

1. Still requires trust in a central authority!
2. Distributed setting: inputs are connected to identifiers such as IP address in logs etc.

Problems

However, there are some problems with (centralized) DP:

1. Still requires trust in a central authority!
2. Distributed setting: inputs are connected to identifiers such as IP address in logs etc.

Goal

- Compute user statistics in a distributed setting, w/o trust in central authority
- Efficient computation, low communication cost (important on mobile devices)

Randomized response

Idea for the distributed setting: randomized response

Survey technique introduced in 1965 to get accurate statistics on sensitive topics.

Example: Have you used drugs this month?

Randomized response

1. Participants toss a coin
2. Answer truthfully if coin comes up heads
3. If tails, participant tosses a 2nd coin:
   - "Yes" if tails
   - "No" if heads

True answer if first coin heads or 50% chance on second coin.
Answer truthfully ∼75% of the time.

Randomized response: analysis

Given positive answers X, true number of "Yes" answers:

$$Y = 2(X - 0.25)$$

Result: plausible deniability for participants, accurate statistics

Local Differential Privacy

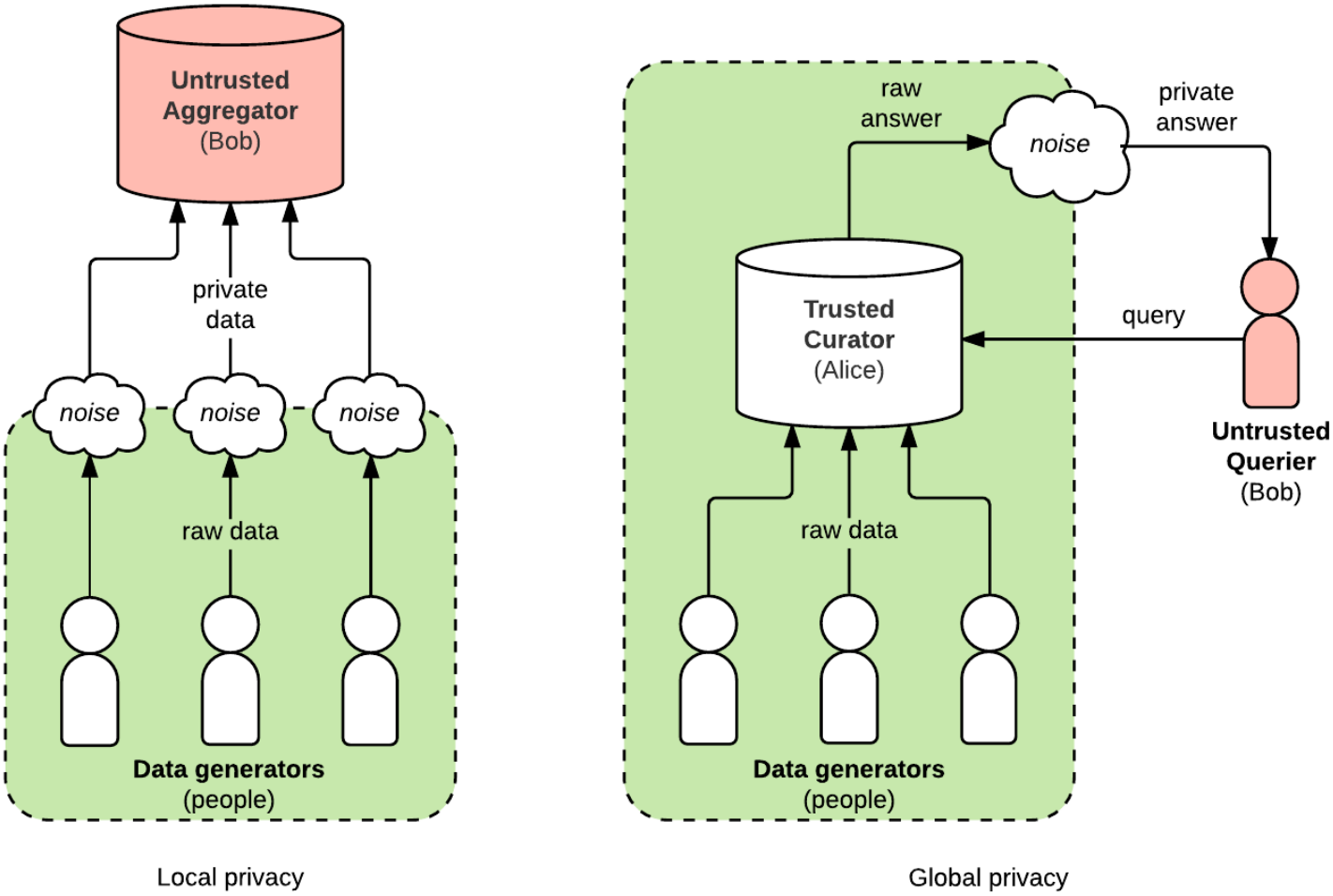

Figure: Local differential privacy vs. differential privacy

Local Differential Privacy

Definition

We say that an algorithm π satisfies ϵ-Local Differential Privacy where $\epsilon > 0$ if and only if for any input v and v'

$$\forall y \in Range(\pi) : \frac{Pr[\pi(v) = y]}{Pr[\pi(v') = y]} \leq e^{\epsilon}$$

where $Range(\pi)$ denotes every output of the algorithm π.

For the randomized response example: survey participants are given ϵ-LDP guarantee with $\epsilon = ln(0.75/(1 - 0.75)) = ln(3)$

Challenges in the local model

Challenges

It is much harder to construct protocols for the local model:

- more difficult to maintain low error-bound
- maintain low communication cost (ideally only a few bits)

Challenges in the local model

Challenges

It is much harder to construct protocols for the local model:

- more difficult to maintain low error-bound
- maintain low communication cost (ideally only a few bits)

centralized model: add Laplace noise of magnitude $O(1/\epsilon)$, independent of number of participants

local model: lower-bound $\Omega(\sqrt{n}/\epsilon)$, dependent on number of participants

Key problems in LDP

There are a number of different problems in LDP. We will look at a few of them:

- Frequency estimation
- Heavy hitter identification
- Private spatial data collection

Frequency Oracles

Definition

Given a domain $\mathcal{D}$ a *frequency oracle* (FO) is a protocol which estimates the frequency of an element $d \in \mathcal{D}$.

Core problem of LDP: locally private frequency estimation

FO framework

Define *pure* FO protocols by a composition of **encoding** and **perturbation** and **support**.

Encoding: Encoding of each value in the domain $\mathcal{D}$

Perturbation: Random perturbation of encoded values

Support: Mapping of each output y to the set of inputs that support the output value y

The true frequency $c(i)$ of a value i can then be estimated given the encoding, perturbation and support.

Direct Encoding

Generalization of randomized response with coin flip.
Encode$(v) = v$. Perturb with:

$$\Pr[\text{Perturb}(x) = i] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + |\mathcal{D}| - 1}, & \text{if } i = x \\ q = \frac{1-p}{|\mathcal{D}|-1} = \frac{1}{e^\epsilon + |\mathcal{D}| - 1}, & \text{if } i \neq x \end{cases}$$

The support function is $\text{Support}_{\text{DE}}(i) = \{i\}$.

Unary Encoding

Value v is encoded as a bit vector where only the v-th position equals 1 and all other positions equal 0. Given probabilities p, q the perturbed output B' is computed as follows:

$$\Pr[B'[i] = 1] = \begin{cases} p, & \text{if } B[i] = 1 \\ q, & \text{if } B[i] = 0 \end{cases}$$

with optimal parameters $p = \frac{1}{2}$ and $q = \frac{1}{e^{\epsilon}+1}$ and the support function $\text{Support}_{\text{UE}}(B) = \{i \mid B[i] = 1\}$.

Heavy hitter identification

Goal

Estimate the frequency of the most common domain elements (*heavy hitters*) while guaranteeing ϵ-LDP.

For small domains: simply estimate frequency of all domain elements using FO protocol

Computationally infeasible for large domains.

Heavy hitter identification

Definition

Set of n users each holding an input $x_i \in \mathcal{D}$, *distributed database* $S = (x_1, \ldots, x_n)$ consisting of all users' inputs. A domain element $x \in \mathcal{D}$ is Δ-heavy if its multiplicity in S is at least Δ.

Goal

Find all Δ-heavy elements (i.e. *heavy hitters*) for Δ as small as possible.

Informally: Δ-heavy if there are at least Δ users who hold the input x. Then Δ is also referred to as the protocol's *error*.

Heavy hitter identification

Solution

Use efficient FO protocol, minimize queries necessary to identify the most frequent items.

One approach: use binary prefix tree, prune all subtrees that cannot be prefixes of heavy hitters

Private spatial data collection

Services such as *Google Maps* and *Waze* benefit from user location data to identify popular locations and to create traffic congestion maps.

Private spatial data collection

Goal

We want to maintain users' privacy and be able to give privacy guarantees while collecting useful spatial data.

Unfortunately: domain too big to obtain accurate results while maintaining ϵ-LDP

Chen et al. introduce *personalized local differential privacy*

Private spatial data collection

Definition

Given the personalized privacy specification (τ, ϵ) of a user u, a randomized algorithm π satisfies (τ, ϵ)-personalized local differential privacy for u, if for two locations $l, l' \in \tau$ and any $O \subseteq Range(A)$,

$$\frac{\Pr[\pi(l) \in O]}{\Pr[\pi(l') \in O]} \leq e^{\epsilon}$$

where the probability space is over the coin flips of π.

τ determines user's *safe region* which they do not mind revealing (i.e. "I am in New York state" but not more fine-grained than that)

PLDP is a generalized version of ϵ-LDP as $\tau = \mathcal{L}$ for a location universe $\mathcal{L}$ implies regular ϵ-LDP.

Private spatial data collection

Solution

- Each user chooses safe region τ
- Perturb reported locations using local randomizer which guarantees (τ, ϵ)-PLDP.
- Partition users into clusters with same safe region to minimize error
- Estimate user counts for all locations by combining the estimates from the clusters accordingly

Practical deployments of DP

Local Differential Privacy is still a very active area of research.

Where is (Local) Differential Privacy in use today?

Practical deployments of DP

In Google Chrome (RAPPOR):

- collecting user statistics
- gaining insights in common settings, how Chrome is used
- find common homepages

Basic RAPPOR uses a variant of Unary Encoding as its FO protocol

Practical deployments of DP

On iOS and MacOS:

- for general usage statistics
- to find popular emojis
- to find new trending words that are not in the dictionary yet
- etc.

Client side: randomized response on bit vector encodings

Aggregation: uses *count-mean sketch with Hadamard transform*, requires only a single bit (!) to be transmitted by the client

Practical deployments of DP

Criticism: lack of transparency in Apple's implementation, weak privacy guarantees, high privacy loss of $\epsilon = 16$ per day

However: important first steps to adoption of (local) differential privacy have been taken, privacy guarantees can be given at scale

Thank you for your attention.