October 29: Differential Privacy

Scribe: Jordan Kellerstrass

1 Overview

Differential Privacy (DP) is a more rigorous definition of privacy in the context of accessing statistical datasets. The defining goal is to hide the data of a specific user such that whether or not s/he participated in the database is unknowable with high probability. This is accomplished by adding noise relative to the sensitivity of the query before returning results, where sensitivity is the likelihood of learning something new about an individual. Differential Privacy is needed because absolute disclosure prevention is proven impossible and a privacy policy such as *only ask statistical queries* turns out to be non-trivial. Differential Privacy is a feature of algorithms that interact with datasets.

2 Main Ideas

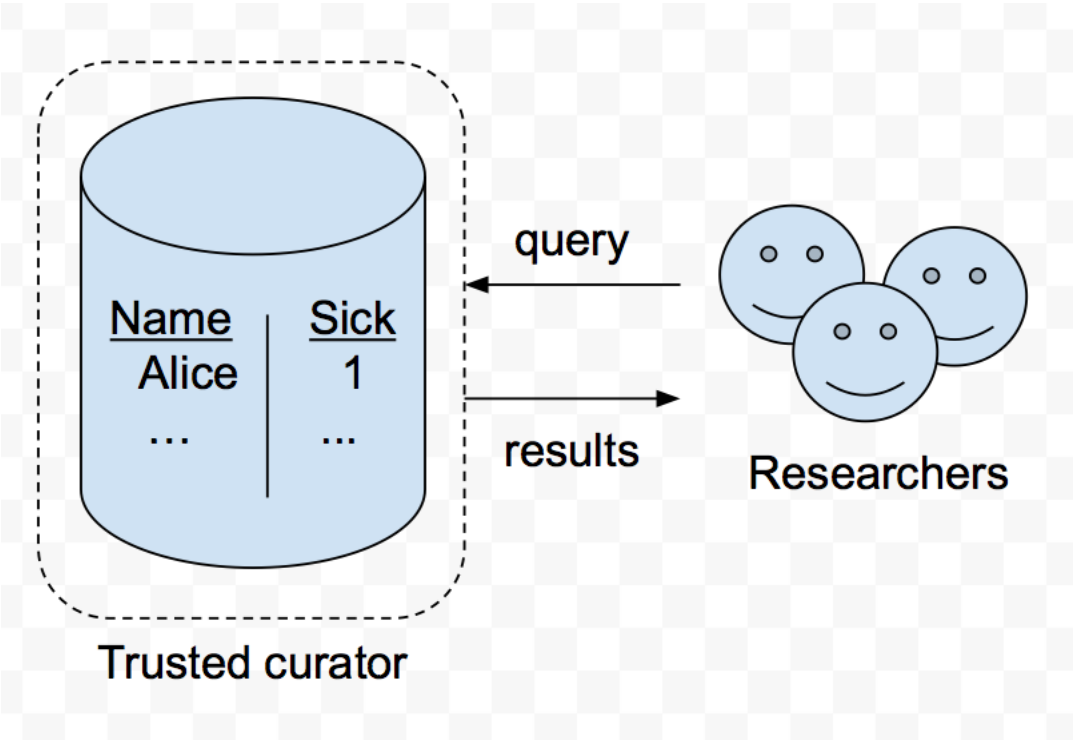

Figure 1: Trusted curator facilitates differential privacy.

2.1 Definitions

- Auxilary datasets D_1 and D_2 differ by exactly one element
- k-anonymizations and deidentification are other mechanisms of protecting k entries in a sample data set. The impossibility result shows that if this type of sanitation results in either insufficient privacy or insufficiently useful data.
- Noise looks like data and is artificially generated to hide revealing information. Noise is added to query results proportional to the most one row could pull the actual result in either direction.
- $\mathcal{K}_f$ is the mechanism that adds noise and gives results.
- ϵ represents the degree to which differential privacy is achieved.
  - Noise $\times \frac{1}{\epsilon} = \epsilon$-differential privacy
  - A larger value of ϵ denotes a less differentially private query result.
  - ϵ degrades nicely in that ϵ_{total} is equal to the sum of all ϵ_{query} added together for a given user or organization.
  - A database curator can give users or organization 'privacy budgets' in terms of ϵ, so that when their cumulative ϵ_{query}s add up to their privacy budget, they will no longer have access to execute new queries on the data.
- L-1 sensitivity describes how much an attribute (query result) of a dataset will be affected by changing, adding, or subtracting one entry.
  - L1-sensitivity of a function f, where f: $\mathcal{D} \rightarrow R^d$, is $\Delta f = max\|f(D_1) - f(D_2)\|_1$
  - if f is a counting function, Δf=1
  - if f returns a value in [0,100], Δf=100

From the paper: "A randomized function K gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(K)$,

$\Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S]$"

This probability holds if you switch D_1 and D_2 and applies to group privacy, although since statistics are meant to reveal information about large groups, privacy degrades as group size increases.

2.2 How to Achieve Differential Privacy

ϵ-differential privacy is achieved by $\mathcal{K}_f$ computing $f(X)$, adding scaled symmetric exponential distribution noise with variance σ^2 to each coordinate of $f(X)$, which is relative to the L1-sensitivity (see 2.1) of the query function f. Note that X refers to the database, but computation of the sensitivity of $f(X)$ is independent of X.

This is described by the density function, where a is the provided answer to $f(X)$:

$$\Pr[\mathcal{K}_f(X) = a] \propto \exp(-\|f(X) - a\|_1/\sigma)$$

From the paper, "For $f : \mathcal{D} \rightarrow R^d$, the mechanism $\mathcal{K}_f$ gives $(\Delta f/\sigma)$-differential privacy."

This can be done with any function, although only some functions will remain useful.

3 Example

Consider census data collection, which promises to protect privacy without actually defining what privacy is. Unsanitized algorithms could execute a differencing attack such as, how many members of X have Y, and then how many members of X except for A have Y? Both answers would return a seemingly aggregated response, but together result in a clear privacy breach. The fundamental law of information recovery states that overly accurate estimations of too many statistics is blatantly non-private, even if traditionally identifying information such as name, birthday, SSN, etc. are excluded. Implementing differential privacy mitigates this issue as long as ϵ is managed adequately.

4 Caveats

A known issue with DP mechanisms is that by executing many adjacent queries, one can infer most of the information that DP is designed to conceal. As such, an organization or individual can be given a privacy budget in terms of total ϵ. Once the user has depleted his/her privacy budget, further DB accesses are rejected. Still, nothing prevents multiple users from talking to each other to learn information they weren't intended to have access to.

By definition DP does not account for auxiliary information used to learn something about a person in relation to statistics from the database. Instead, DP provides that within some small margin, disclosures are just as likely whether or not someone participated in the database.

5 Presentations

5.1 PINQ: Privacy Integrated Queries

PINQ is a data analysis programming interface that enforces differential privacy, namely without placing trust in the expertise or diligence of the user. The implementation is based on C#'s LINQ and acts as a protective layer in between raw data and the analysts or programmers accessing it in a protected way. The data source provider specifies the differential privacy requirements. When an analyst accesses the data, PINQ provides the differentially private transformations and aggregations implementation of the query written in and executed by LINQ.

5.2 DJoin: Differentially Private Join Queries over Distributed Databases

DJoin performs join queries from multiple databases (such as travel patterns and disease outbreaks, data managed by different companies) such that the end result has the same degree of noise as the component steps. This is accomplished by de-noising intermediate query results, combining them, and then reintroducing the noise. The main idea is accomplished by multi-party computation.