STATS/BIODS 352: Practical Privacy-preservation with Differential Privacy and OpenDP
Part II

James Honaker
james@hona.kr

Meta and School of Engineering & Applied Sciences
Harvard University

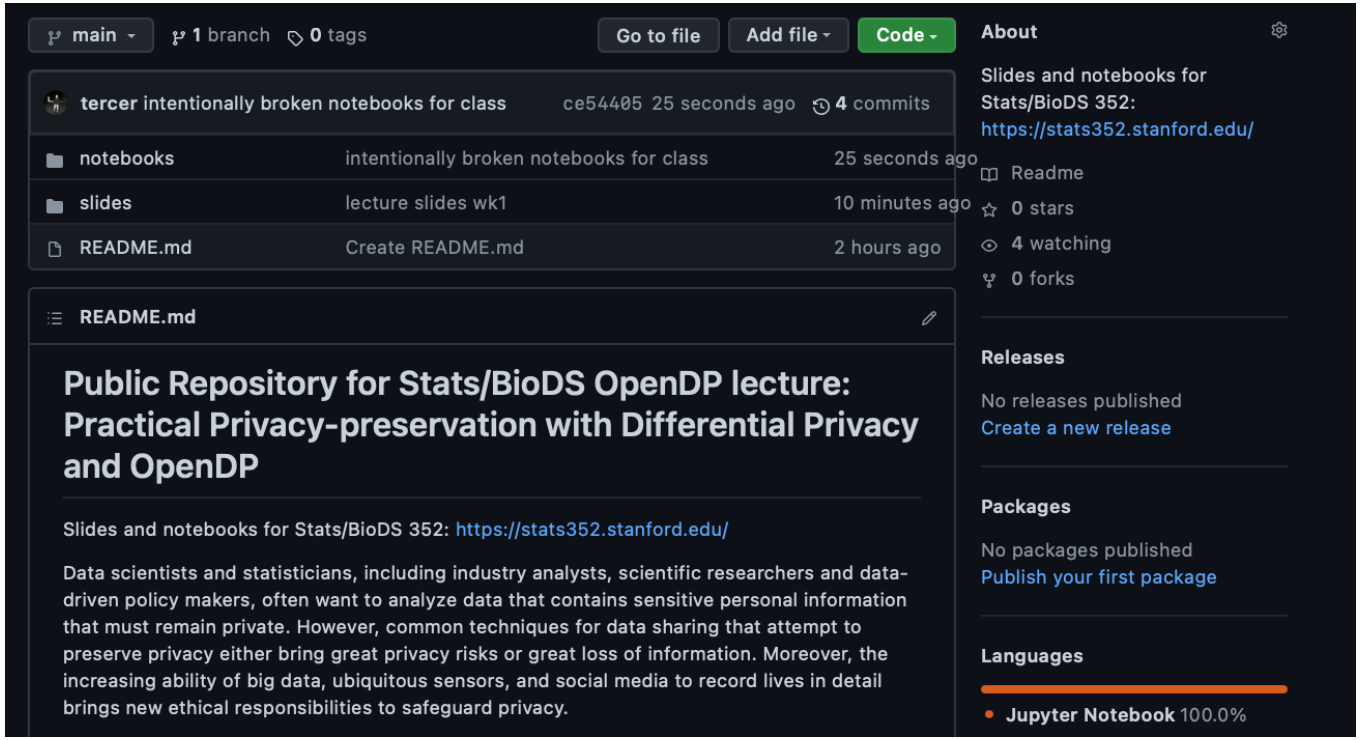
April 14, 2022

- **Privacy Attacks**
  - Linkage Attacks
  - Reconstruction Attacks
- **Differential Privacy**
  - Definition
  - Adding Noise (Laplace mechanism)

– WEEK 2 –

- **More Foundations**
  - Counts, Medians, Standard Deviations.
  - Choosing Privately (Exponential mechanism)
  - Differentially Private Optimization (DP-SGD)
- **Implementations**
  - OpenDP (Open source library of DP components for statistical releases)
  - Opacus (Open source DP optimization for PyTorch ML)

GitHub Repository



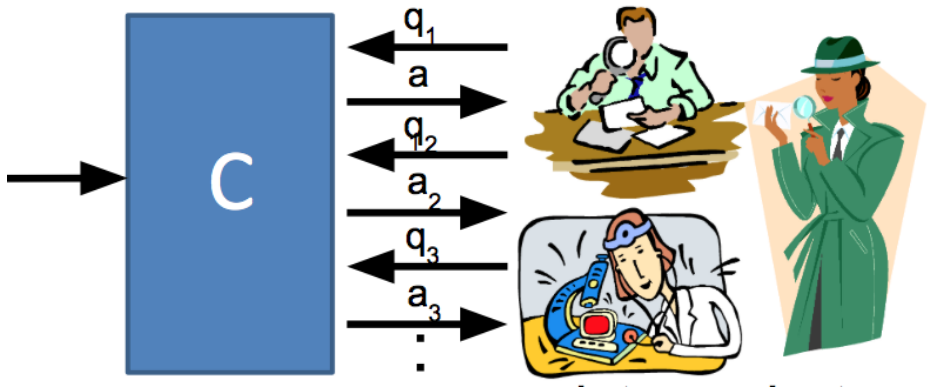
https://github.com/opendp/stats352/

Differential Privacy

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
M	A		N
M	O		N
M	O		Y
F	A		N
M	B		Y

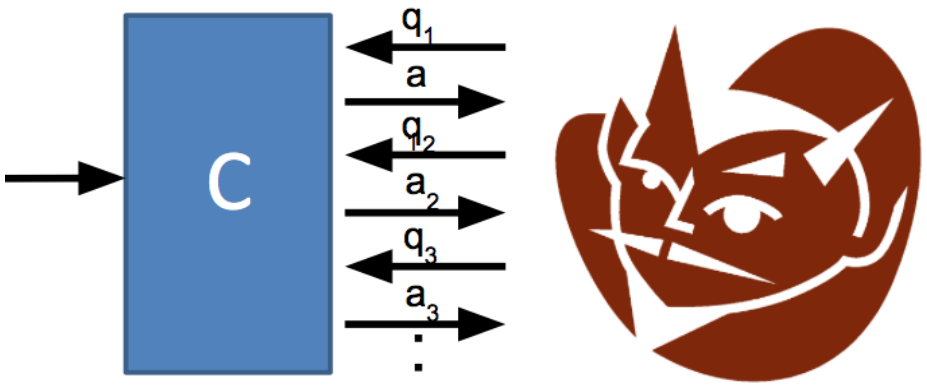


Data Repository Curator Analysts

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
M	A		N
M	O		N
M	O		Y
F	A		N
M	B		Y

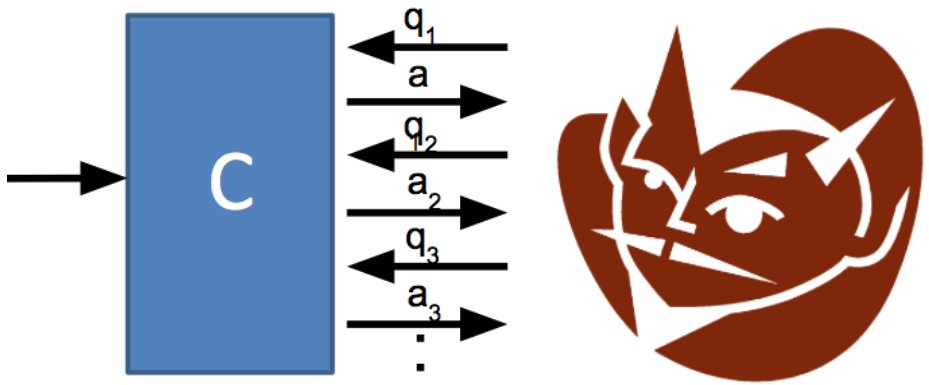


Data Repository Curator Adversary

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
M	A		N
M	O		N
M	O		Y
F	A		N
M	B		Y

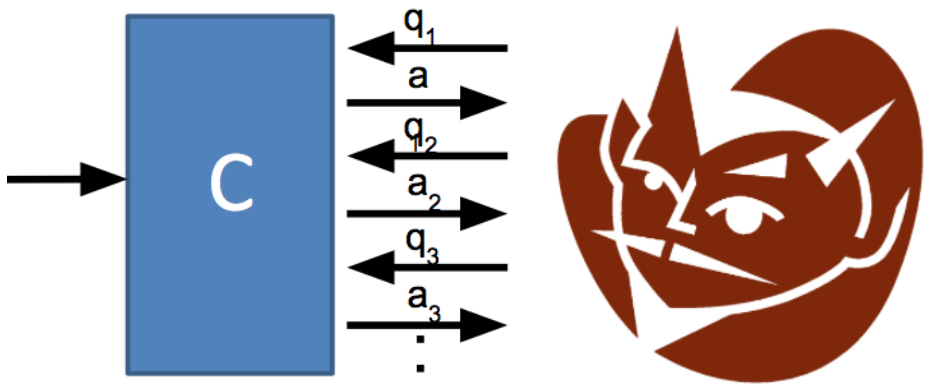


Data Repository Curator Adversary

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
M	O		N
M	O		Y
F	A		N
M	B		Y

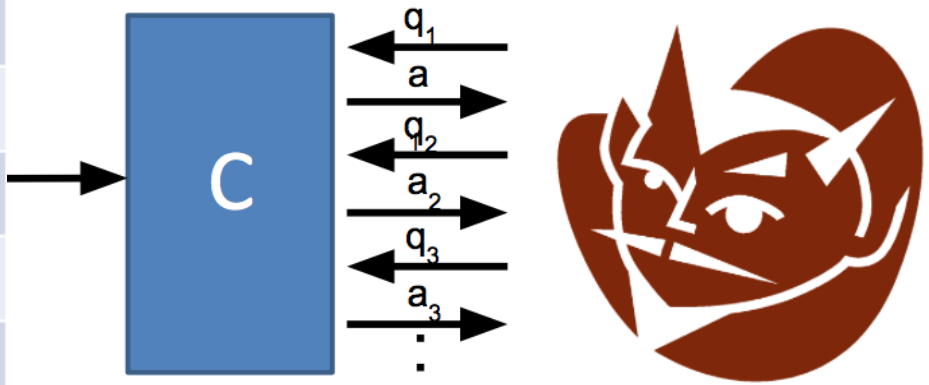


Data Repository Curator Adversary

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
F	A		Y
M	O		N
M	O		Y
F	A		N
M	B		Y



Data Repository Curator Adversary

Differential Privacy

M is ϵ-DP if

$$Pr[M(D, q) \in T] \leq (1 + \epsilon) Pr[M(D', q) \in T], \qquad \forall\, T, q.$$

Differential Privacy

M is ϵ-DP if

$$Pr[M(D, q) \in T] \leq (1 + \epsilon) Pr[M(D', q) \in T], \qquad \forall\, T, q.$$

- D, D' Neighbouring datasets

Differential Privacy

M is ϵ-DP if

$$Pr[M(D,q) \in T] \leq e^{\epsilon} Pr[M(D',q) \in T], \qquad \forall\, T, q.$$

- D, D' Neighbouring datasets
- M Mechanism that Maps from data to result

Differential Privacy

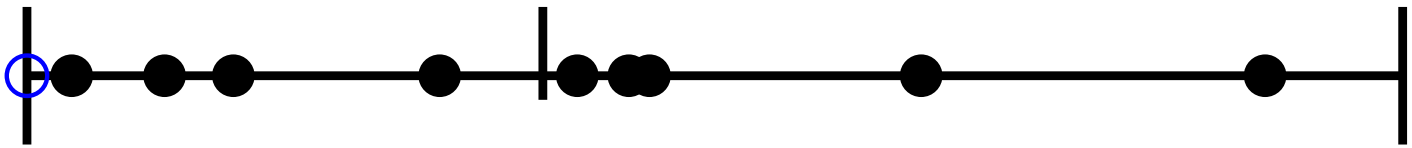
M is ϵ-DP if

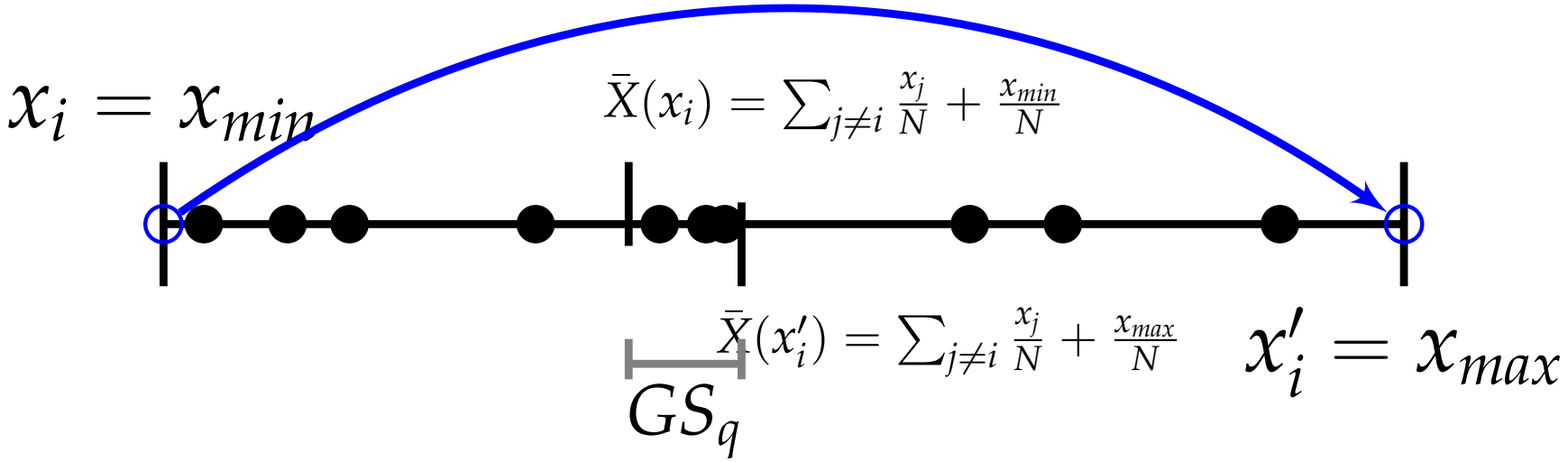
$$Pr[M(D,q) \in T] \leq e^{\epsilon} Pr[M(D',q) \in T], \qquad \forall\, T, q.$$

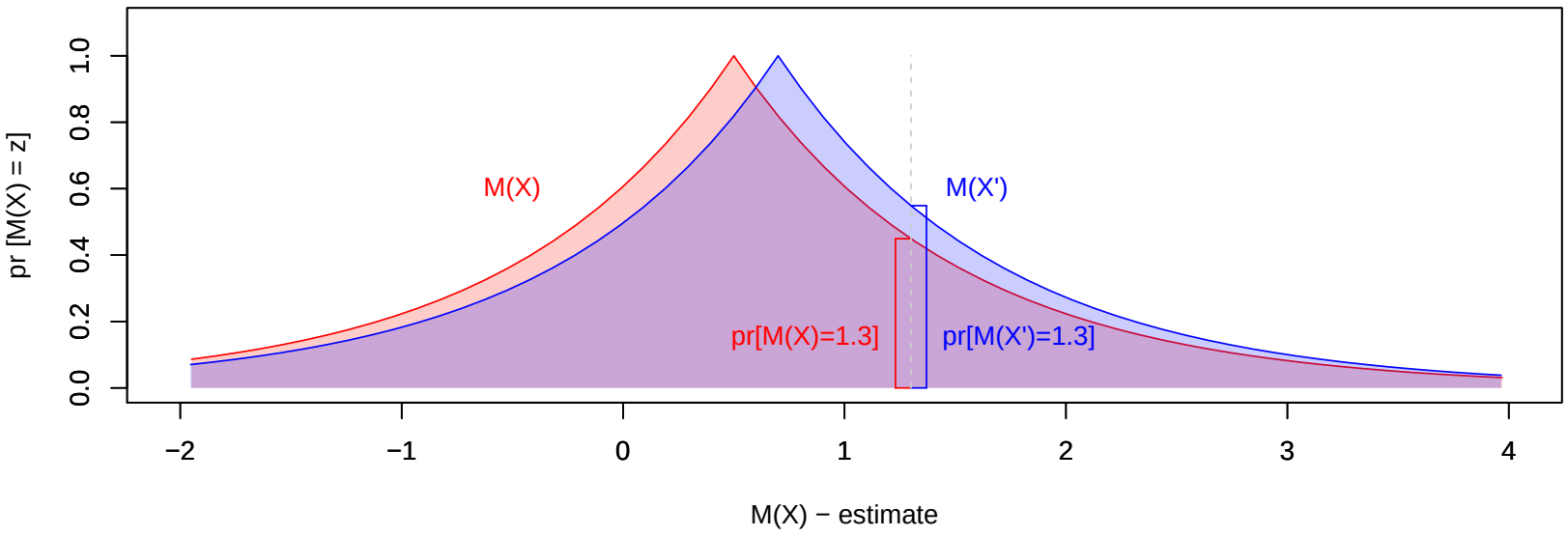
- D, D' Neighbouring datasets
- M Mechanism that Maps from data to result
- q Query
- T Set providing a decision rule

$x_i = x_{min}$

$\bar{X}(x_i) = \sum_{j \neq i} \frac{x_j}{N} + \frac{x_{min}}{N}$

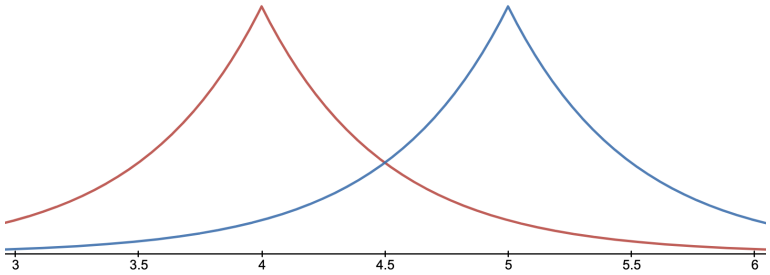



$x_i = x_{min}$
$\bar{X}(x_i) = \sum_{j \neq i} \frac{x_j}{N} + \frac{x_{min}}{N}$
$\bar{X}(x_i') = \sum_{j \neq i} \frac{x_j}{N} + \frac{x_{max}}{N}$
$x_i' = x_{max}$
GS_q

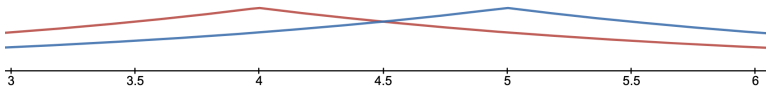

M(X)
M(X')
pr[M(X)=1.3]
pr[M(X')=1.3]
pr [M(X) = z]
0.0
0.2
0.4
0.6
0.8
1.0
−2
−1
0
1
2
3
4
M(X) − estimate

Privacy vs Accuracy

More **noise** gives more **privacy** (smaller ϵ) and more **error**

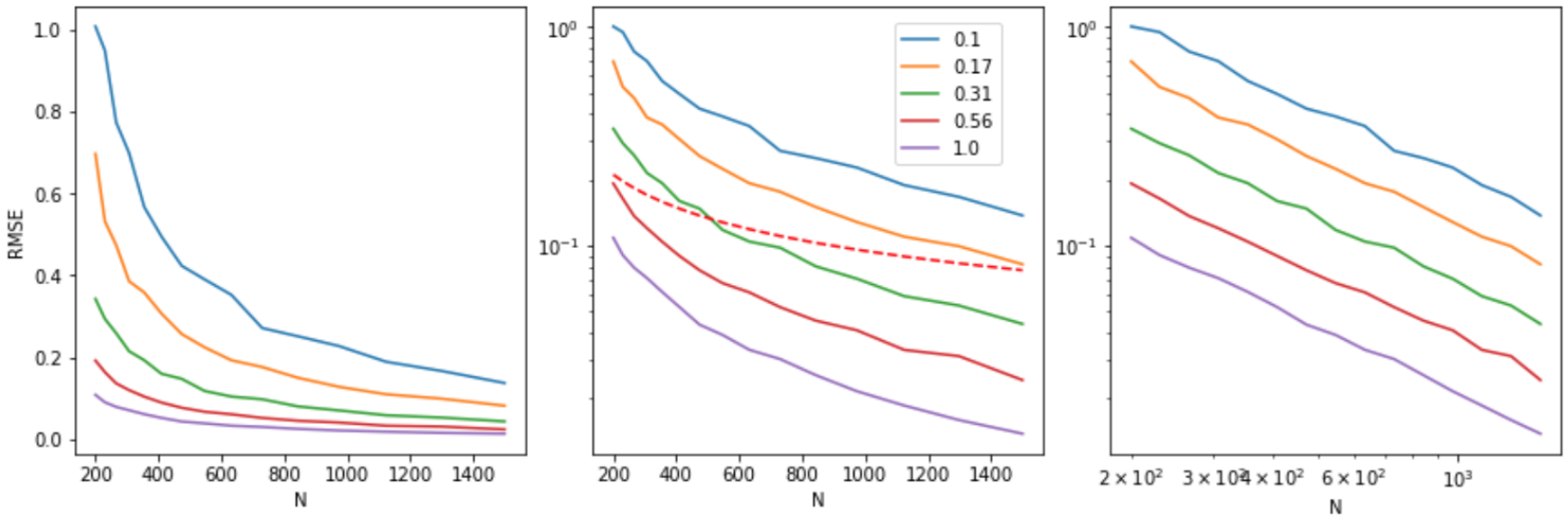


$\epsilon = 2$



$\epsilon = 0.5$

Error from Laplace over ϵ, N



see `laplace_mechanism_and_opendp.ipynb`

Key Differential Privacy Properties:

1. **Post-processing:** Any function on a DP release is still DP
2. **Programability:** Any combination of DP releases is also DP.
3. **Composition:** Privacy loss can be summed across DP releases.

Basic Composition

How to count privacy for multiple queries?
Claim: If $\forall i = 1, \ldots, k$, $A_i :\rightarrow \mathbb{R}^d$ satisfies (ϵ, δ)-DP, then $B(D) = A_1(D), A_2(D), \ldots, A_k(D)$ satisfies $(k\epsilon, k\delta)$-DP.
Example:

- q_1 : histogram of the age, (ϵ, δ)-DP.
- q_2: proportion of the male, (ϵ, δ)-DP.

$q = (q_1, q_2)$ satisfies $(2\epsilon, 2\delta)$-DP.

Advanced Composition

Can we do better? Yes!
Claim: If $\forall i = 1, \ldots, k$, $A_i :\rightarrow \mathbb{R}^d$ satisfies (ϵ, δ)-DP, then $B(D) = A_1(D), A_2(D), \ldots, A_k(D)$ satisfies

$$\epsilon\sqrt{k \ln(1/\delta')} + e^{\epsilon} - 1k, k\delta + \delta'\text{-DP},$$

for any $\delta' > 0$.

Remark 1: when ϵ is small, privacy increases with a sublinear rate!
This is quite common in machine learning.
Remark 2: still loose for specific algorithms, e.g., Gaussian mechanism.

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function
- **Privacy Loss Parameter** - Formal measure of privacy loss (e.g., ϵ, δ, ρ)

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function
- **Privacy Loss Parameter** - Formal measure of privacy loss (e.g., ϵ, δ, ρ)
- **Privacy Loss Budget** - Bound on cumulative privacy loss

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function
- **Privacy Loss Parameter** - Formal measure of privacy loss (e.g., ϵ, δ, ρ)
- **Privacy Loss Budget** - Bound on cumulative privacy loss
- **Composition** - Parameters across releases add up

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function
- **Privacy Loss Parameter** - Formal measure of privacy loss (e.g., ϵ, δ, ρ)
- **Privacy Loss Budget** - Bound on cumulative privacy loss
- **Composition** - Parameters across releases add up
- **Post Processing** - Any function on a DP release is still DP

Pereira et al. US Broadband Coverage Data Set: A DP Data Release

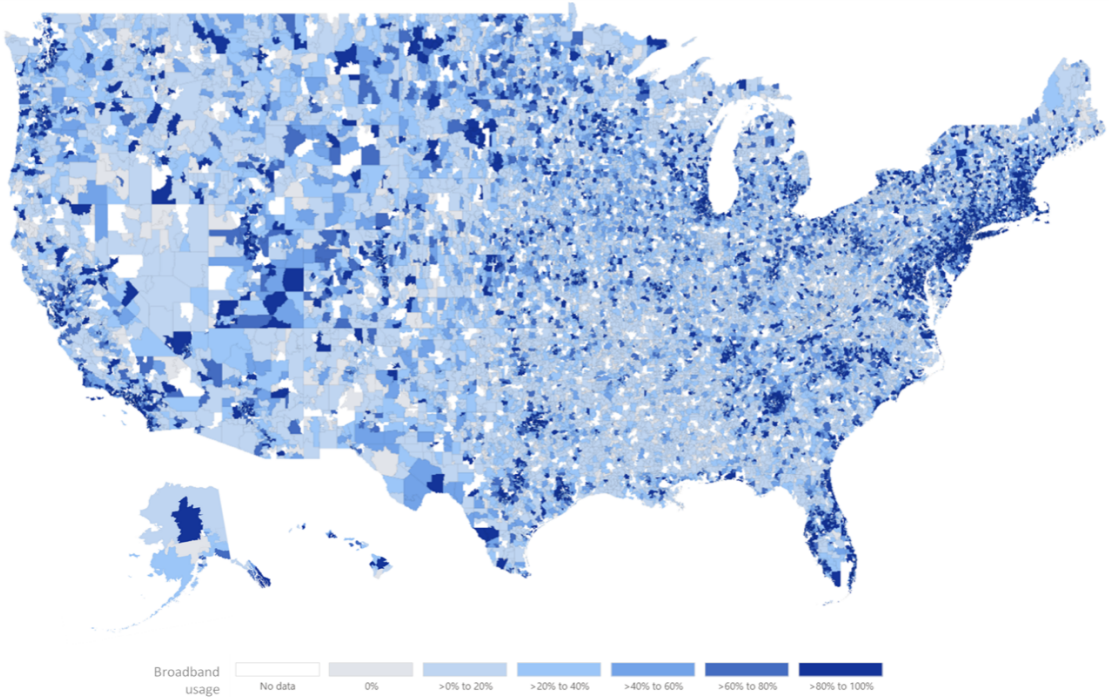
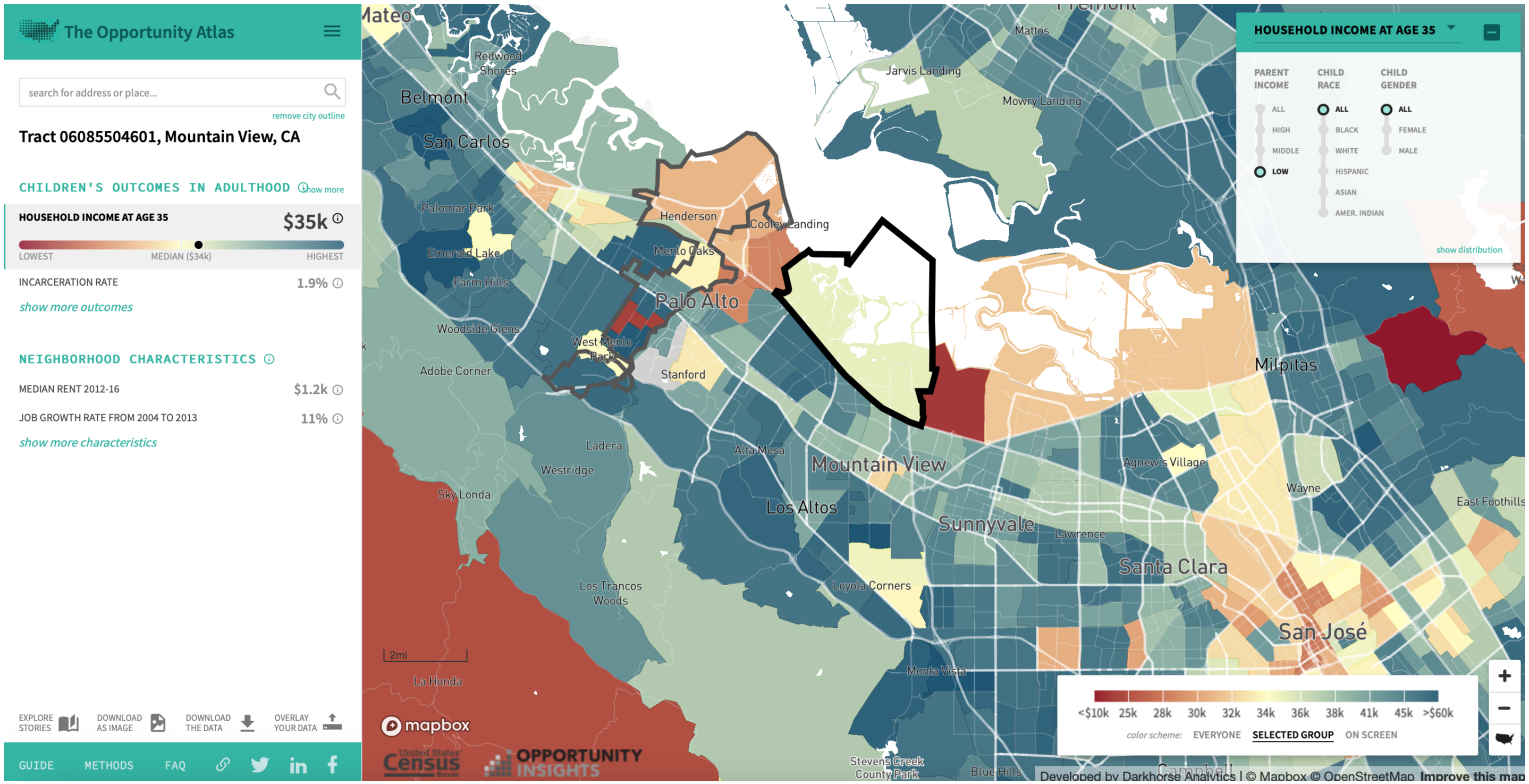


Fig. 1. Map of the United States by postal codes with indicators of broadband coverage.

From Microsoft Services, we query the following Windows telemetry data:
– L_z: Counts of devices connecting to Microsoft Services with internet speed lower than 25Mbps in zip code z.
– H_z: Counts of devices connecting to Microsoft Services with internet speed greater or equal than 25Mbps in zip code z.
Additionally, we query Microsoft Services for the following data:
– M_z: Counts of devices utilizing Microsoft Services in zip code z.
– O_z: Counts of devices not utilizing Microsoft Services in zip code z.


The Opportunity Atlas
search for address or place...
remove city outline
Tract 06085504601, Mountain View, CA
CHILDREN'S OUTCOMES IN ADULTHOOD
show more
HOUSEHOLD INCOME AT AGE 35
$35k
LOWEST
MEDIAN ($34k)
HIGHEST
INCARCERATION RATE
1.9%
show more outcomes
NEIGHBORHOOD CHARACTERISTICS
MEDIAN RENT 2012-16
$1.2k
JOB GROWTH RATE FROM 2004 TO 2013
11%
show more characteristics
EXPLORE STORIES
DOWNLOAD AS IMAGE
DOWNLOAD THE DATA
OVERLAY YOUR DATA
GUIDE
METHODS
FAQ
HOUSEHOLD INCOME AT AGE 35
PARENT INCOME
ALL
HIGH
MIDDLE
LOW
CHILD RACE
ALL
BLACK
WHITE
HISPANIC
ASIAN
AMER. INDIAN
CHILD GENDER
ALL
FEMALE
MALE
show distribution
<$10k 25k 28k 30k 32k 34k 36k 38k 41k 45k >$60k
color scheme: EVERYONE SELECTED GROUP ON SCREEN
Belmont
San Carlos
Palo Alto
Mountain View
Los Altos
Sunnyvale
Santa Clara
San José
Milpitas
Stanford
Menlo Oaks
Henderson
Cooley Landing
Jarvis Landing
Mowry Landing
Alta Mesa
Agnews Village
Wayne
East Foothills
Lawrence
Loyola Corners
Monta Vista
Westridge
Ladera
Sky Londa
Los Trancos Woods
La Honda
Adobe Corner
Woodside Glens
Palomar Park
Emerald Lake
Farm Hills
Redwood Shores
2mi
mapbox
Census
OPPORTUNITY INSIGHTS
Developed by Darkhorse Analytics | © Mapbox © OpenStreetMap Improve this map

Implementations

Count

Let $q(x)$ be $\#\{i : x \in B\}$.
Then GS(q) = 1.

Count

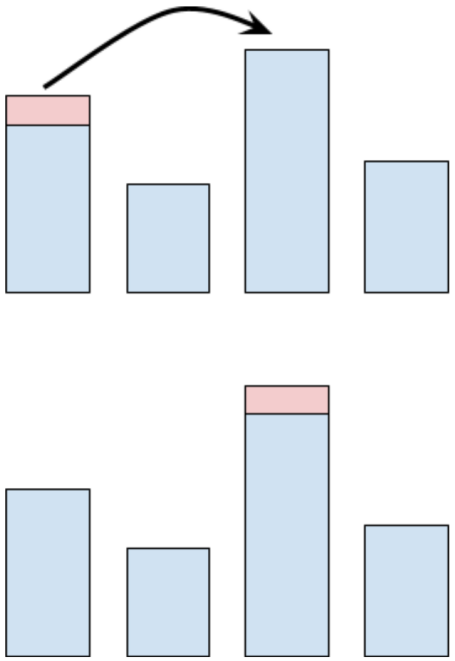
Let $q(x)$ be $\#\{i : x \in B\}$.
Then GS(q) = 1.

Let $B_1, \cdots, B_k$ be disjoint bins.
Let $q = \{q_1, \cdots, q_k\}$ as $q_i = \#\{i : x \in B_i\}$.
Then GS(q) = 2. $(\forall k)$

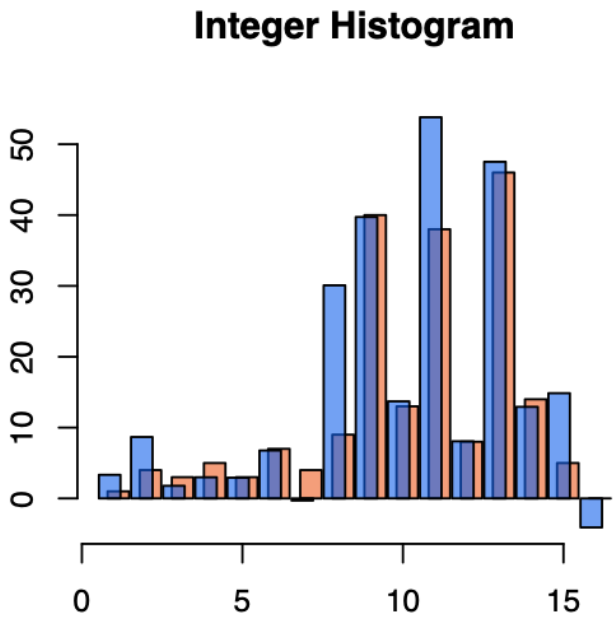
Count

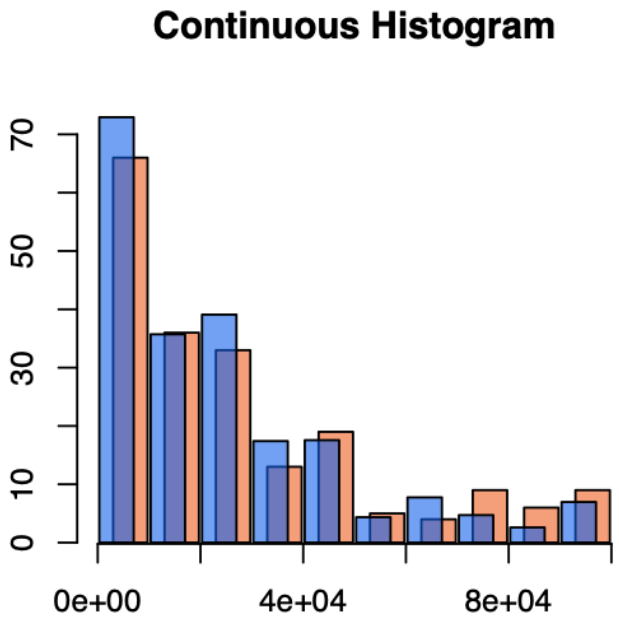
Let $q(x)$ be $\#\{i : x \in B\}$.
Then GS(q) = 1.

Let $B_1, \cdots, B_k$ be disjoint bins.
Let $q = \{q_1, \cdots, q_k\}$ as $q_i = \#\{i : x \in B_i\}$.
Then GS(q) = 2. $(\forall k)$



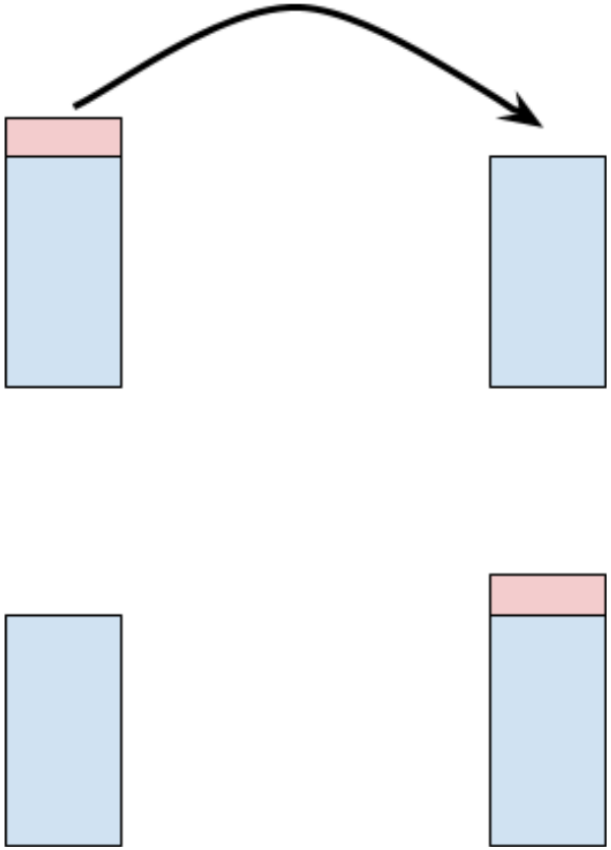
Count


Integer Histogram
0
10
20
30
40
50
0
5
10
15


Continuous Histogram
0
10
30
50
70
0e+00
4e+04
8e+04

Median

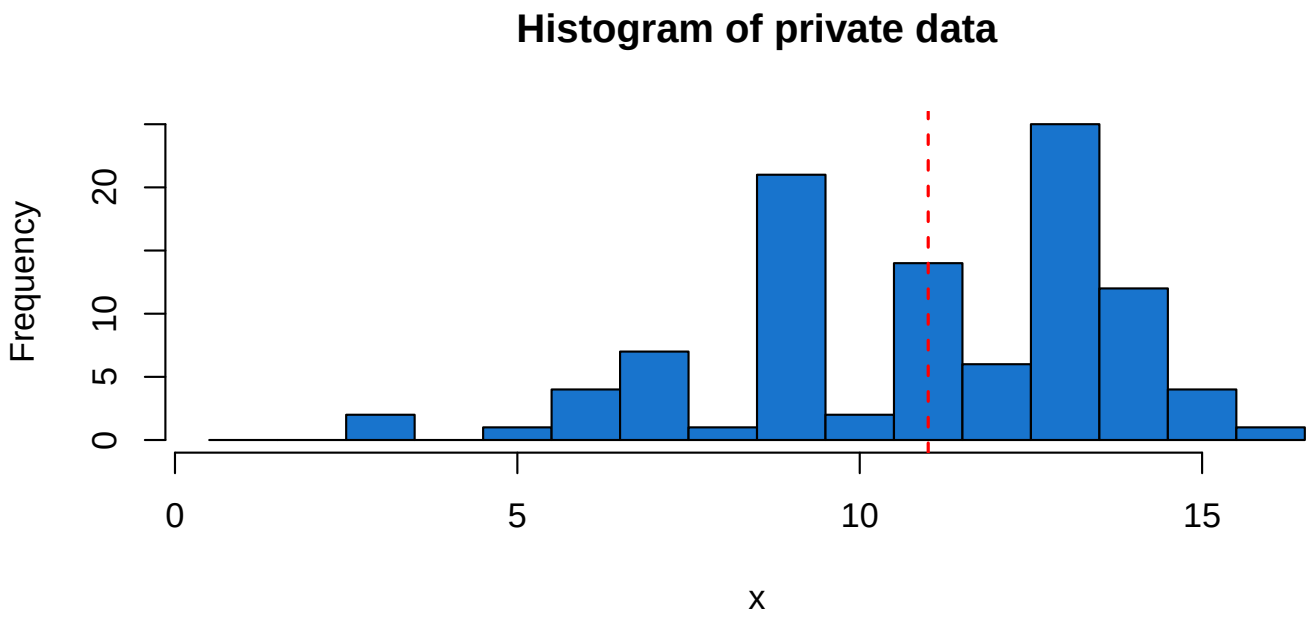
Let $q = y_{n/2}$ where Y is the ordering of X.
Then GS(q) = $x_{max} - x_{min}$

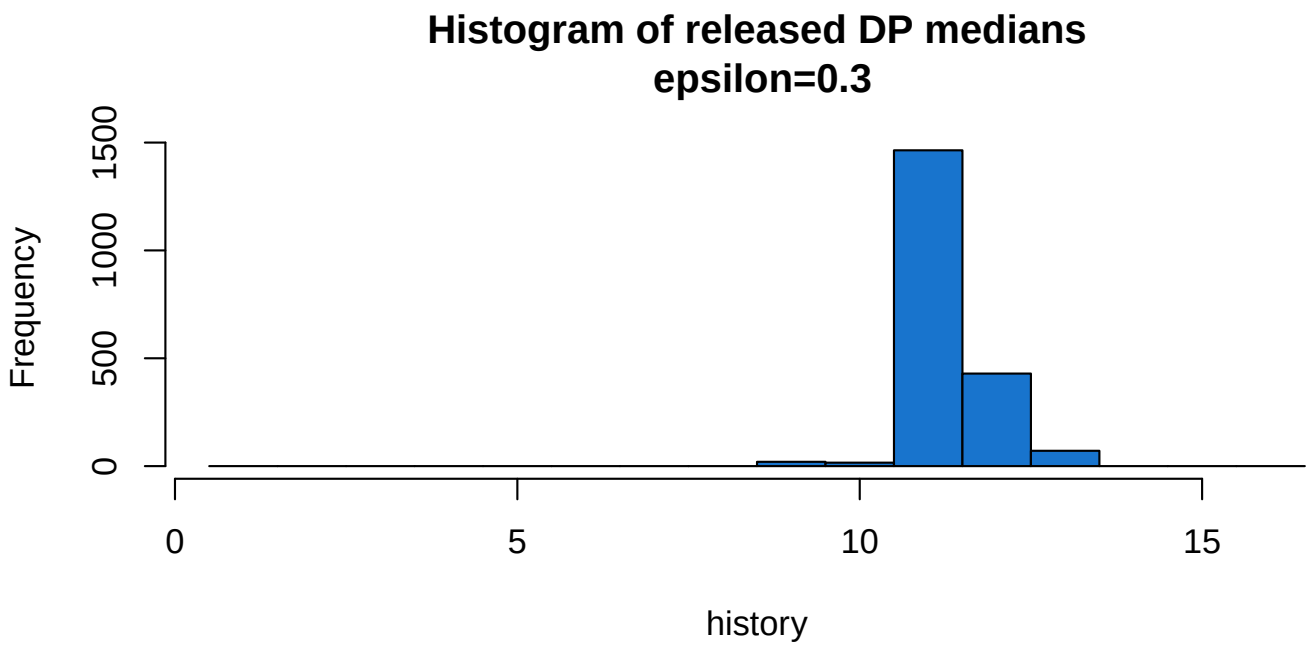


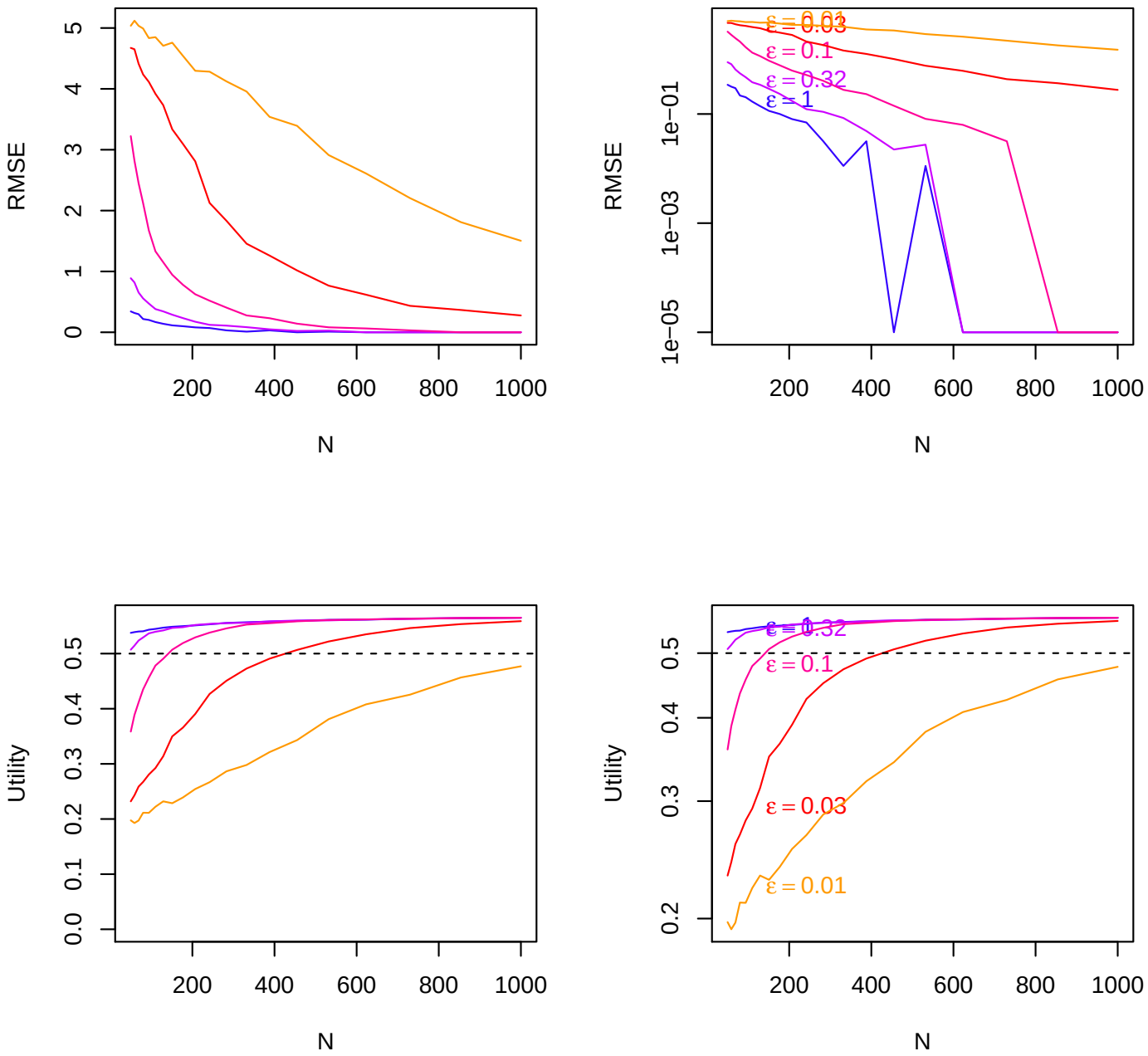
Exponential Mechanism for the Median

- Say $\mathcal{X} = \{1, 2, \cdots, M\}$.
- $M(x)$: output $y \in \mathcal{X}$ with prob $\propto \exp(\epsilon \cdot u(x, y)/2)$
  Where $u(x, y) = \min\{\#\{i : x_i \leq y\}, \#\{i : x_i \geq y\}\}$.
- Note that true median y^* has $u(x, y^*) \geq n/2$.
- Can show that for all x, with high probability,

$$u(x, M(x)) \geq n/2 - O(log(M)/\epsilon)$$


Histogram of private data
Frequency
0
5
10
20
0
5
10
15
x


Histogram of released DP medians
epsilon=0.3
Frequency
0
500
1000
1500
0
5
10
15
history


RMSE
N
0
1
2
3
4
5
200
400
600
800
1000
RMSE
N
1e−01
1e−03
1e−05
ε = 0.01
ε = 0.03
ε = 0.1
ε = 0.32
ε = 1
Utility
N
0.0
0.1
0.2
0.3
0.4
0.5
Utility
N
0.2
0.3
0.4
0.5
ε = 0.32
ε = 0.1
ε = 0.03
ε = 0.01

Variance

Tracking sensitivity for complex combinations of transformations/chaining of functions is difficult.

Theorem

The sensitivity of the sample estimated variance of N observations is:

$$GS(s^2) = \frac{n-1}{n^2}\left(y_{max} - y_{min}\right)^2$$

$$\Rightarrow GS(s) \approx \frac{y_{max} - y_{min}}{\sqrt{n}}$$

Proof.

We can rewrite the sample variance as:

$$
\begin{aligned}
s^2 &= \frac{1}{n}\sum_i (\bar{y} - y_i)^2 && (1)\\
&= \bar{y}^2 - 2\bar{y}\frac{1}{n}\sum_i y_i - \frac{1}{n}\sum_i y_i^2 && (2)\\
&= \left(\frac{1}{n}\sum_i y_i^2\right) - \bar{y}^2 && (3)\\
&= \frac{1}{n}\sum_i y_i^2 - \left(\frac{1}{n}\sum_i y_i\right)^2 && (4)\\
&= \frac{1}{n}\left[y_j^2 + \sum_{i\neq j} y_i^2\right] - \frac{1}{n^2}\left(y_j + \sum_{i\neq j} y_i\right)^2 && (5)\\
&= \frac{1}{n}\left[y_j^2 + \sum_{i\neq j} y_i^2\right] - \frac{1}{n^2}\left(y_j^2 + 2y_j\sum_{i\neq j} y_i + \left(\sum_{i\neq j} y_i\right)^2\right) && (6)\\
&= \frac{1}{n}\left[\sum_{i\neq j} y_i^2 - \frac{1}{n}\left(\sum_{i\neq j} y_i\right)^2\right] + \frac{1}{n^2}\left((n-1)y_j^2 - 2y_j\sum_{i\neq j} y_i\right) && (7)\\
&= \left[\frac{1}{n}\sum_{i\neq j} y_i^2 - \left(\frac{1}{n}\sum_{i\neq j} y_i\right)^2\right] + \frac{n-1}{n^2}\left(y_j^2 - 2y_j\bar{y}_{-j}\right) && (8)
\end{aligned}
$$

where $\bar{y}_{-j}$ is the mean after removing the jth observation:

$$
\bar{y}_{-j} = \frac{1}{n-1}\sum_{i\neq j} y_i \tag{9}
$$

The first term is not a function of y_j (and resembles the variance of the dataset after omitting jth observation, except for the incorrect scaling by n). This gives us partials with respect to the jth observation of:

$$\frac{\delta s^2}{\delta y_j} = \frac{2(n-1)}{n^2}\left(y_j - \bar{y}_{-j}\right) \tag{10}$$

$$\frac{\delta^2 s^2}{\delta y_j^2} = \frac{2(n-1)}{n^2} > 0 \tag{11}$$

Therefore:

$$\arg\min_{y_j} s^2 = \bar{y}_{-j} \tag{12}$$

$$\arg\max_{y_j} s^2 = \begin{cases} y_{\text{min}}, & \text{if } |y_{\text{min}} - \bar{y}_{-j}| \geq |y_{\text{max}} - \bar{y}_{-j}| \\ y_{\text{max}}, & \text{if } |y_{\text{min}} - \bar{y}_{-j}| < |y_{\text{max}} - \bar{y}_{-j}| \end{cases} \tag{13}$$

Thus across all possible values of y_j, the variance is minimized when y_j is at the mean of the rest of the data, and the variance is maximized when y_j is at the bound farthest from the mean.
This gives us the difference in the variance from changing the j-th observation from y_j to y_j' as:

$$s^2(y_j) - s^2(y_j') = \frac{n-1}{n^2}\left(y_j^2 - y_j'^2 - 2(y_j - y_j')\bar{y}_{-j}\right) \tag{14}$$

And thus sensitivity of the variance as:

$$\Delta s^2 = \max_{y_j, y_j'} \left[s^2(y_j) - s^2(y_j') \right] = \max_{y_j, \bar{y}_{-j}} \frac{n-1}{n^2} \left(y_j^2 - \bar{y}_{-j}^2 - 2(y_j - \bar{y}_{-j})\bar{y}_{-j} \right) \tag{15}$$

$$= \max_{y_j, \bar{y}_{-j}} \frac{n-1}{n^2} \left(y_j^2 + \bar{y}_{-j}^2 - 2y_j\bar{y}_{-j} \right) \tag{16}$$

$$= \max_{y_j, \bar{y}_{-j}} \frac{n-1}{n^2} \left(y_j - \bar{y}_{-j} \right)^2 \tag{17}$$

$$= \frac{n-1}{n^2} \left(y_{\max} - y_{\min} \right)^2. \quad \blacksquare \tag{18}$$

It is not generally the case that $\Delta\sqrt{f} = \sqrt{\Delta f}$ for the same reasons that it is not generally true that $\sqrt{a^2 - b^2} = a - b$. Thus sensitivity of the standard deviation need not follow directly as the square-root of the sensitivity of the variance. However, in this special case it does:

Lemma

The sensitivity of the sample standard deviation, s, is:

$$\Delta s = \frac{\sqrt{n-1}}{n} \left(y_{max} - y_{min} \right) \tag{19}$$

Proof.

Since $s \geq 0, \ \forall Y$ then equations 12 and 13 also hold for s. As $\min_{Y_j, Y_{-j}}[s] = 0$ then $\Delta s^2 = \max_{y_j, y_j'} \left[s^2(y_j) - s^2(y_j') \right] = \max \left[s^2(y_j) - 0 \right] \Rightarrow \Delta s = \max \left[s(y_j) - 0 \right] = \sqrt{\Delta s^2}$. $\square$

Covert Channels/Side Channels

As we've seen:

1. there are necessary and more complex DP mechanisms beyond noise addition

Covert Channels/Side Channels

As we've seen:

1. there are necessary and more complex DP mechanisms beyond noise addition
2. calculating or tracking sensitivity for complex combinations of transformations/chaining of functions is difficult.

Covert Channels/Side Channels

As we've seen:

1. there are necessary and more complex DP mechanisms beyond noise addition
2. calculating or tracking sensitivity for complex combinations of transformations/chaining of functions is difficult.

Moreover, programming languages have implementation mismatches to the theorems we prove:

- Timing/Faults
- Floating Point
- Source of Randomization
- Memory Races
- Globals

OpenDP

OpenDP

A **community effort** to build a **trustworthy** and **open-source** suite of differential privacy tools that can be **easily adopted** by custodians of sensitive data to make it available for **research and exploration** in the public interest.

Why?

- Channel our collective advances on science & practice of DP
- Enable wider adoption of DP
- Address high-demand, compelling use cases
- Provide a starting point for custom DP solutions
- Identify important research directions for the field

Project site: `http://opendp.org`

DP Programming Frameworks

Goal: make it easier for a data custodian or analyst to write programs that are DP, and be confident that they actually are DP.

Common approach (starting with PinQ [McSherry `09]):

- (Small) set of trusted DP subroutines: (Lap, Geo, ExpMech, …) only channel for info to flow from dataset to rest of program.
- Track privacy budget consumption: using composition of DP, with either a runtime monitor or static analysis.
- Allow "stable" data transformations: (recursively) track impact on privacy consumption.

Library Architecture

OpenDP

A Programming Framework that:

1. allows arbitrary distance relations to define privacy-loss across a library of data transforms and privacy measures, and concretely reason about and compose total privacy loss,
2. centrally enables interactive queries such as sparse vector and matrix mechanism,
3. can integrate with trusted compute resources (eg. SQL, Spark) for performance at scale.

Library Architecture

OpenDP

Transformation	Stability
Select(T, *maper*)	(1)
Where(T, *predicate*)	(1)
GroupBy(T_1, *keyselector*)	(2)
Join*(T_1,T_2, n, m, *keyselector*$_1$, *keyselector*$_2$)	(n,m)
Intersect(T_1,T_2)	(1,1)
Union(T_1,T_2)	(1,1)
Partition(T, *keyselector*, *keysList*)	(1)

Table 1. Transformation stability

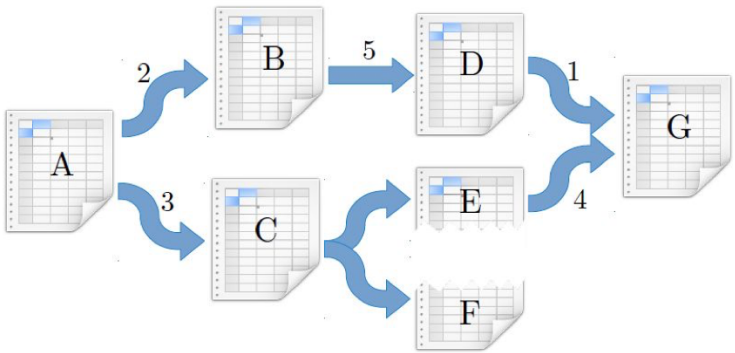


Fig. 2. Transformations

	s	Calculation
A	1	Input table
B	2	$s(A) \times 2$
C	1	$s(A) \times 3$
D	10	$s(B) \times 5$
E	3	$s(C)$
F	3	$s(C)$
G	22	$s(D) \times 1 + s(E) \times 4$

Fig. 3. Scaling factors (s)

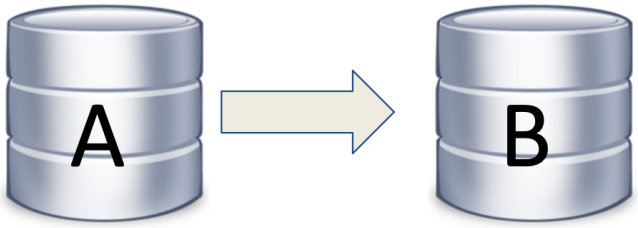
[from Ebadi & Sands, “Featherweight PinQ”, 2017]

Library Architecture

OpenDP

Transformations:

Function from data(sets) to data(sets).

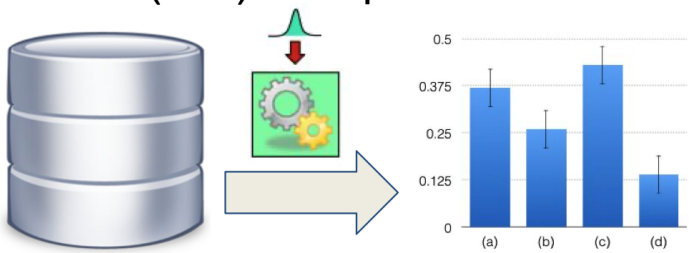


Transformation Attributes

- Input domain
- Input metric
- Output domain
- Output metric
- Function
- Stability relation

Measurements:

Randomized functions from data(sets) to outputs.

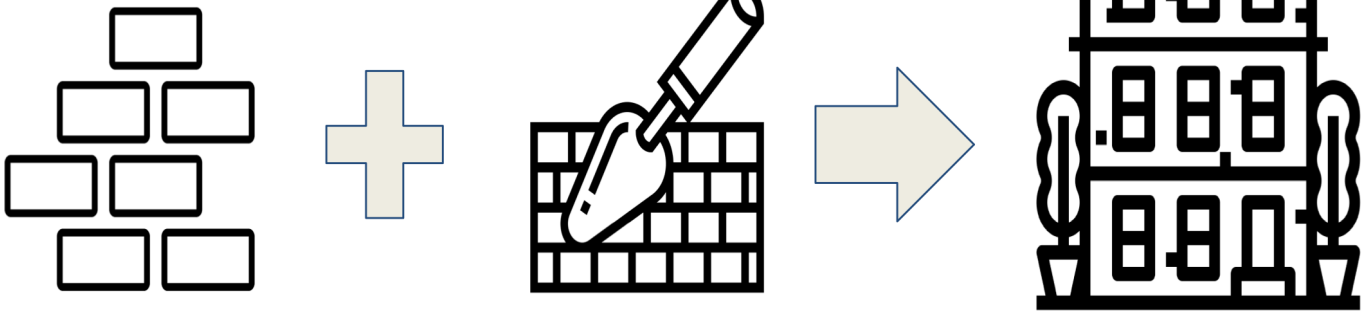


Measurement Attributes

- Input domain
- Input metric
- Output measure
- Function
- Privacy relation

Library Architecture

OpenDP



Measurements & Transformations

Combinators, e.g. Chaining, Composition, Post-processing

Complex DP programs

Library Language Choice

OpenDP

Library implemented in Rust:

- Rust is a fast-growing, imperative functional language.
- high performance, reliable, good memory safety and type safety.
- structures, generics, higher-order functions, and values are immutable by default.
- It has a strong ownership model that gives memory safety.

Collectively, this makes Rust code easier to reason about concretely.

Contrib, Vetting and Proofs

A requirement of the vetting process is having the code supported by a proof document. The library is designed to make this as easy as possible, because it consists of modular building blocks (Transformations and Measurements) for which encapsulated proofs may be written.

Each Transformation or Measurement proof must show the following:

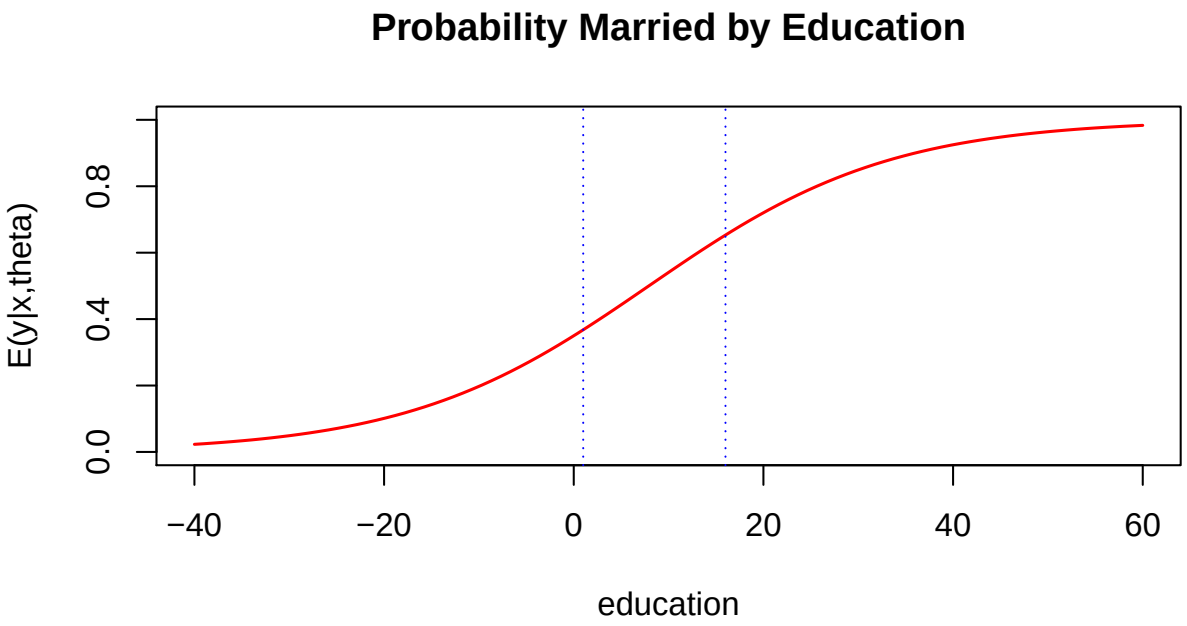
1. That the function, when evaluated on any element in the input domain, emits a value in the output domain.
2. That the relation always returns false if the function is not (d_in, d_out)-close for all d_in and d_out.
3. That your choices of metrics/measures are compatible with your domains.

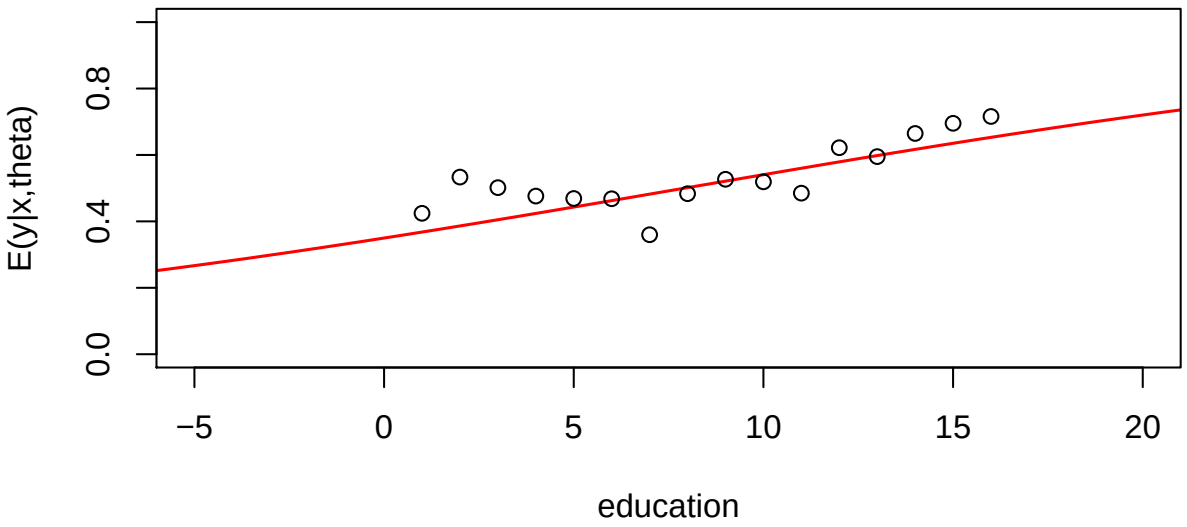
Optimization of Complex Models
DP-SGD and Opacus

Logit Model

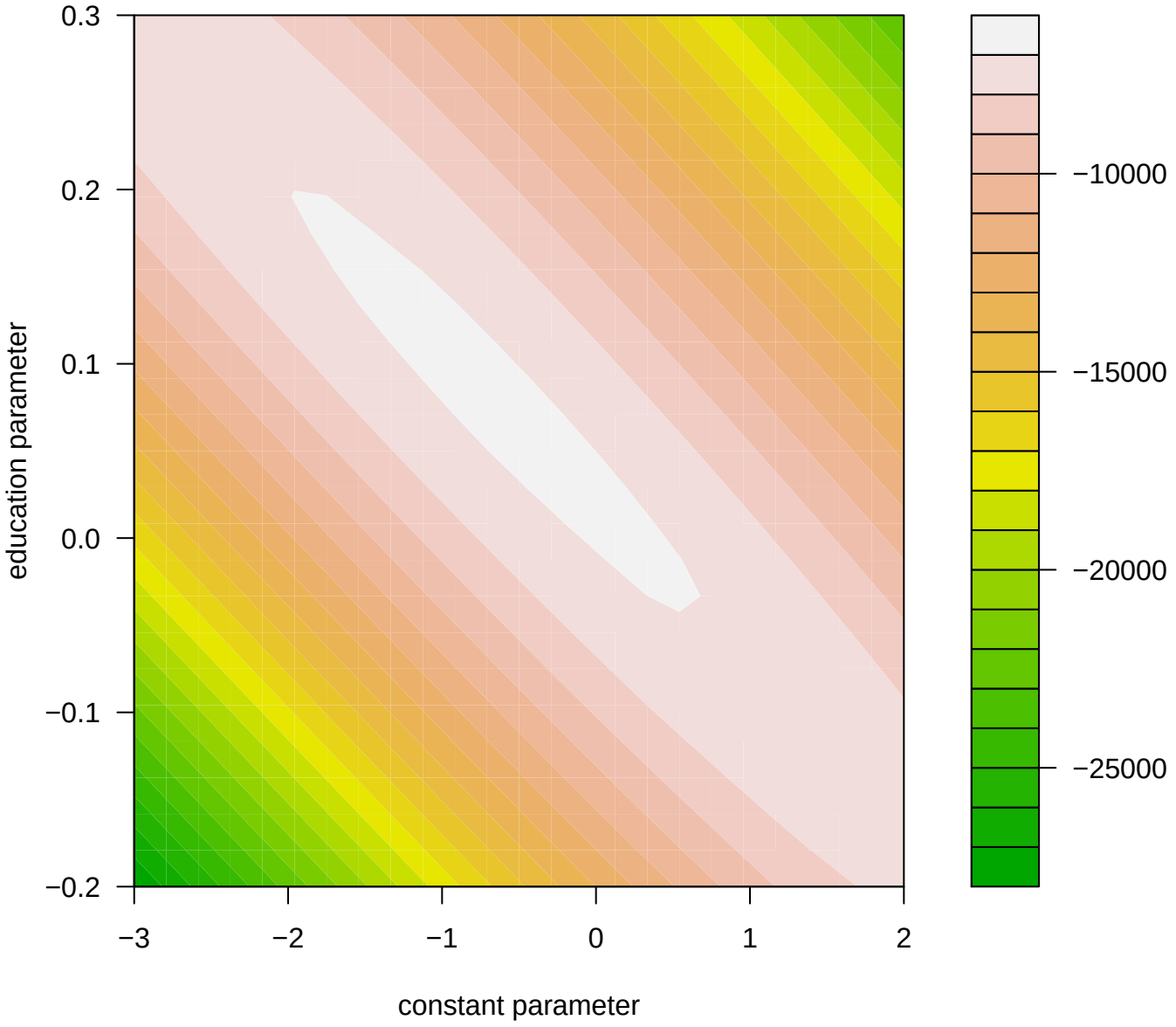
$$logL(y|x,\theta) = \sum_{i=1}^{N} y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i),$$

$$\pi_i = \frac{1}{1+e^{-\beta_0-\beta_1 x_i}}.$$


Probability Married by Education
E(y|x,theta)
0.0
0.4
0.8
−40
−20
0
20
40
60
education


E(y|x,theta)
0.0
0.4
0.8
−5
0
5
10
15
20
education

logLikelihood surface



Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i)/\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

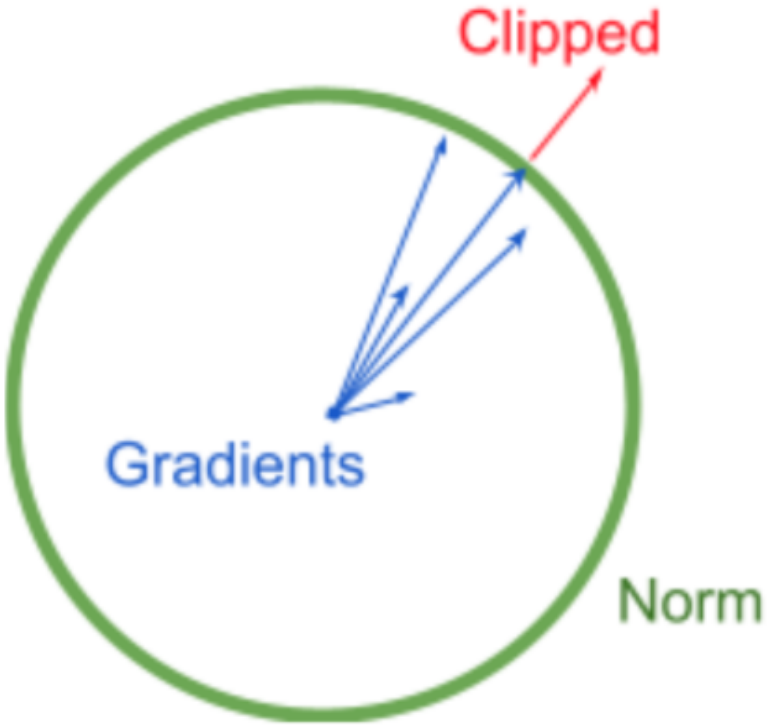
Add noise

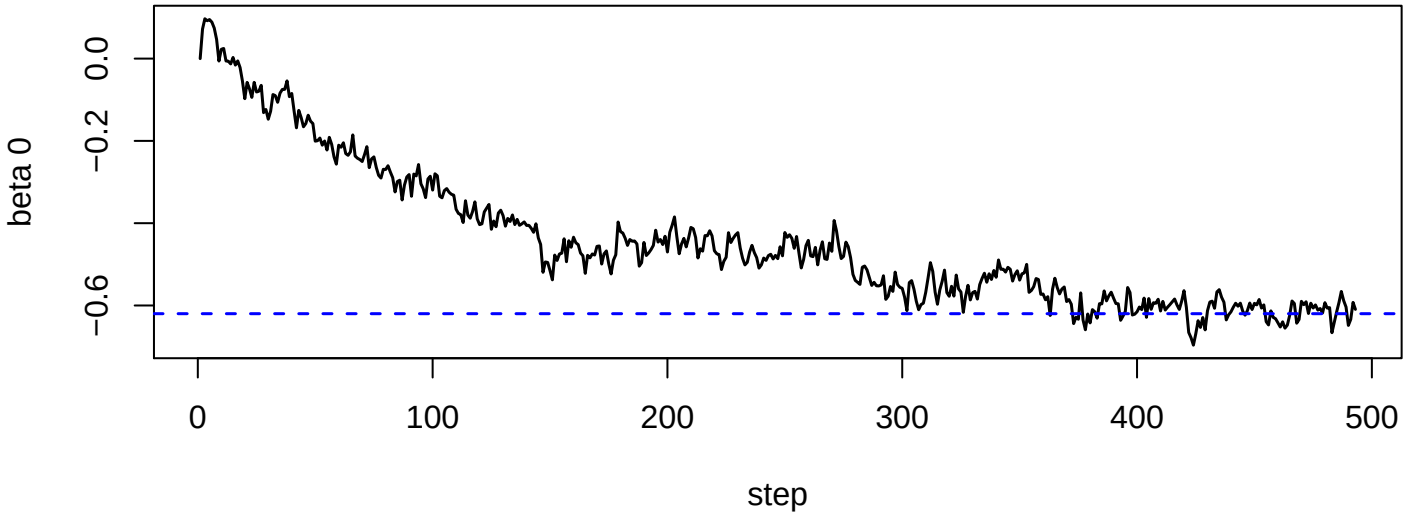
$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

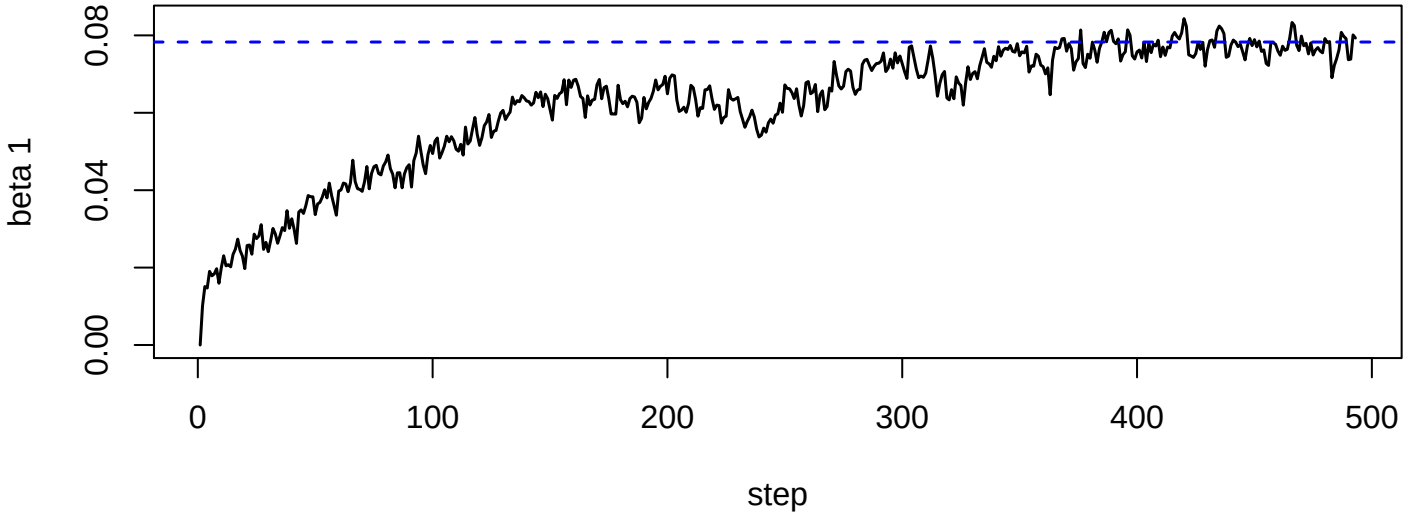
Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.


Clipped
Gradients
Norm


beta 0
0.0
−0.2
−0.6
0
100
200
300
400
500
step


beta 1
0.08
0.04
0.00
0
100
200
300
400
500
step

Parameter Tuning

Learning Rate	η_t
Clipping Norm	C
Batch Size	L

Approaches to parameter tuning:

Parameter Tuning

Learning Rate	η_t
Clipping Norm	C
Batch Size	L

Approaches to parameter tuning:

- Public data and transfer parameters
  (*Deep Learning with Differential Privacy* [Abadi *et al.* 2016])
  - Find similar styled public data, tune parameters there, transfer.

Parameter Tuning

Learning Rate	η_t
Clipping Norm	C
Batch Size	L

Approaches to parameter tuning:

- Exponential mechanism over private models (*Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism* [Raskhodnikova & Smith 2016])
  - Requires score function that has low sensitivity
  - Use (generalized) exponential mechanism over models

Parameter Tuning

Learning Rate	η_t
Clipping Norm	C
Batch Size	L

Approaches to parameter tuning:

- Private selection
  (*Private Selection from Private Candidates* [Liu & Talwar 2019])
  - Requires DP score function
  - Randomized stopping algorithm tunes parameters an indefinite period of time
  - however, lower expected computation and lower privacy consumption.

Broader Choices

- Instance level gradients
- Mechanisms
- Batch Sampler (Tensorflow Chunking, Opacus Uniform with replacement across batches)
- Composition
- DP definition

Opacus

Train PyTorch models with Differential Privacy

INTRODUCTION | GET STARTED | TUTORIALS

Key Features

Scalable

Vectorized per-sample gradient computation that is 10x faster than microbatching

Built on PyTorch

Supports most types of PyTorch models and can be used with minimal modification to the original neural network.

Extensible

Open source, modular API for differential privacy research. Everyone is welcome to contribute.

https://opacus.ai

Opacus for PyTorch

Write out a standard PyTorch model:

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class ExampleLogisticModule(nn.Module):
    def __init__(self, input_size):
        super().__init__()
        self.linear = nn.Linear(input_size, 1)

    def forward(self, x):
        x = self.linear(x)
        x = torch.sigmoid(x)
        return x[:,0]
```

Opacus for PyTorch

Write out a standard PyTorch model:

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class ExampleLogisticModule(nn.Module):
    def __init__(self, input_size):
        super().__init__()
        self.linear = nn.Linear(input_size, 1)

    def forward(self, x):
        x = self.linear(x)
        x = torch.sigmoid(x)
        return x[:,0]
```

Swap out the optimizer for DP:

```
from opacus import PrivacyEngine

privacy_engine = PrivacyEngine()
model, optimizer, data_loader = privacy_engine.make_private(
    module=model,
    optimizer=optimizer,
    data_loader=data_loader,
    noise_multiplier=1.0,
    max_grad_norm=0.5,
)
```