Part II

Last time

- Failure of ad-hoc anonymization
  - data linkage and background knowledge
  - 33 bits of entropy
- Reconstruction attacks
  - fundamental law of information recovery
  - Boosting weak signals
  - Approximate inversion
- Randomized response
  - Simple noise addition method
  - How does it generalize?

Today

Differential privacy, a mathematical privacy notion aimed at privacy-preserving statistical data analysis

Main intuition that differential privacy formalizes

“Whether or not you’re in the dataset has little effect on the output of the analysis.”

Differential Privacy
[Dwork-McSherry-Nissim-Smith-06]

Two data sets *D,D'* are called *neighboring* if they differ in at most one data record.

Example: *D* = {GWAS test population}, *D'* = *D* – {Moritz's DNA}

Informal Definition (Differential Privacy):

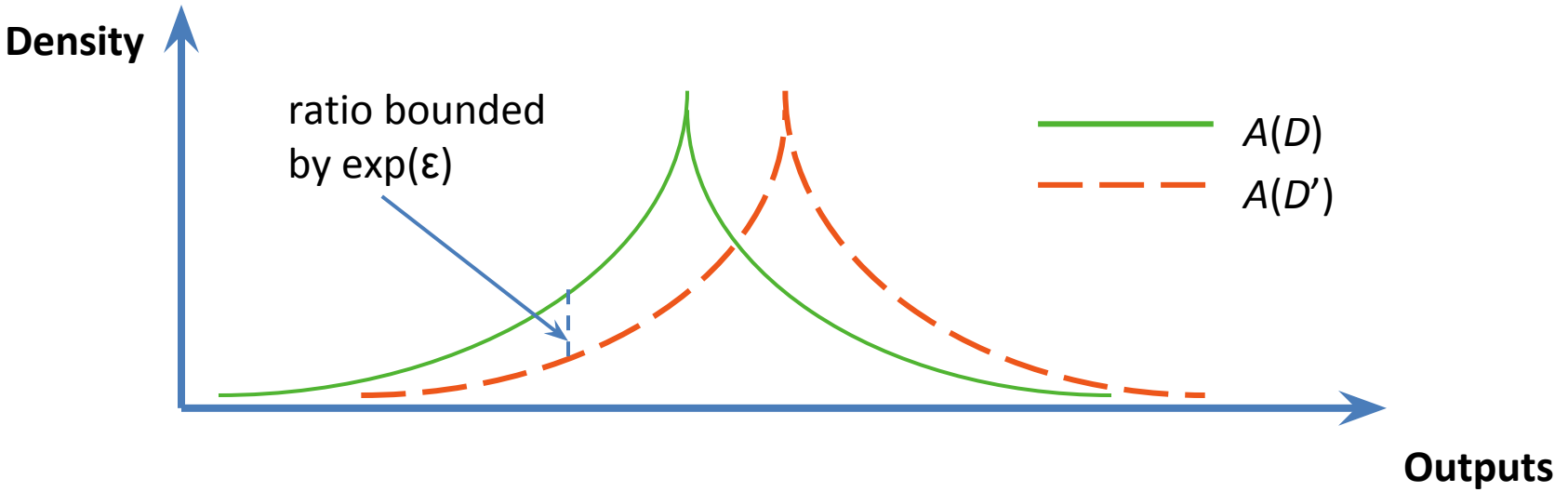
A randomized algorithm *A(D)* is differentially private if for all neighboring data sets *D,D'* and all events *S*:

$$\mathbb{P}\{A(D) \in S\} \approx \mathbb{P}\{A(D') \in S\}$$

Definition (Differential Privacy):

A randomized algorithm *A(D)* is ε-differentially private if for all neighboring data sets *D,D'* and all events *S*:

$$\mathbb{P}\{A(D) \in S\} \leq \exp(\epsilon) \cdot \mathbb{P}\{A(D') \in S\}$$



Interpretation

Differential privacy limits harm resulting from ***participation:***

> *Whether you're in or out is about the same.*

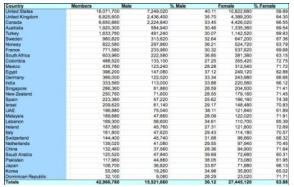
Answers counterfactual: What would've happened, had I not participated?

Population-level inferences still possible and perhaps harmful to you

> Data shows smoking causes cancer
>
> You smoke; your insurance premium goes up

Query model

Trusted Curator

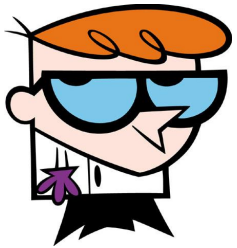


data set D

Query q

Output:

$q(D)$ + noise



Analyst

Requirement: Output satisfies differential privacy

Goal: “Minimize |noise|”

Statistical queries

Database D subset of some universe X

Example: $X = \{0,1\}^d$ (binary d-tuples)

Statistical query: predicate $q : X \to [0, 1]$

Answer $q(D) := \Sigma_{x \text{ in } D}\, q(x)$

between 0 and $n = |D|$

- Example: "How many people in D smoke and have cancer?"

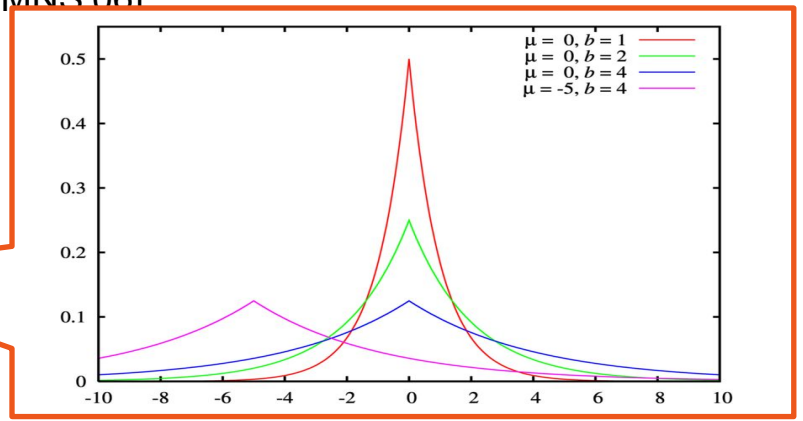
By definition: For neighboring D, D': $|q(D) - q(D')| \leq 1$

This is called the **query sensitivity**. Sensitivity determines noise level. Everything we do in this lecture assumes sensitivity 1.

Laplacian Mechanism [DMNS'06]

Given query q:

1. Compute $q(D)$
2. Output $q(D) + \text{Lap}(1/\varepsilon)$



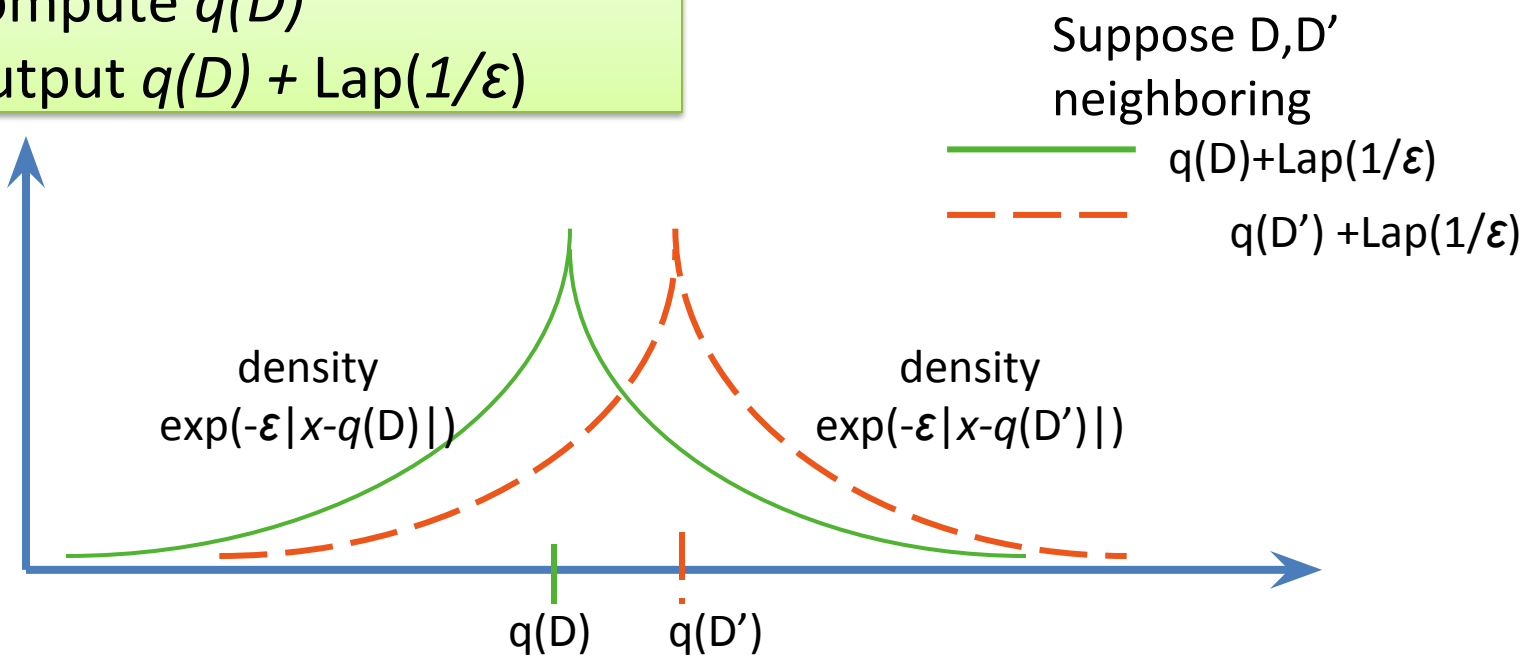
Density $\exp(-\varepsilon|x-q(D)|)$

Fact: Satisfies ε-differential privacy

Laplacian Mechanism [DMNS'06]

Given query q:

1. Compute $q(D)$
2. Output $q(D) + \text{Lap}(1/\varepsilon)$



Recall randomized response

Suppose n individuals have sensitive -1/1 bits $D = (b_1, b_2, \ldots b_n)$

Randomized response (RR):

1. Compute noisy bits $b_i' \sim$ **Bernoulli**$(\frac{1}{2} + \varepsilon b_i)$.
2. Release $\mathrm{RR}(D) = \sum_i b_i'$

Claim: For $\varepsilon < \frac{1}{4}$, RR satisfies 2ε-differential privacy.

Local differential privacy

Each individual computes the randomization “locally” before sending differentially private input to aggregation step

I.e., in randomized response noisy bit b_i' is already differentially private

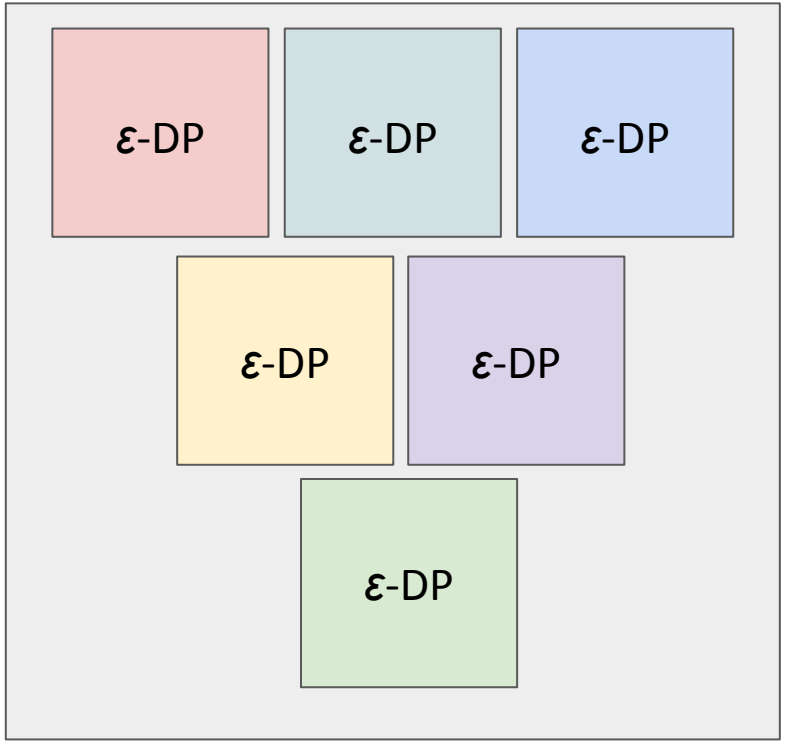
Contrast with “central differential privacy”: Compute $\sum_i b_i + \text{Lap}(2/\varepsilon)$

Central approach adds less noise

Local approach gives privacy even when data curator (aggregator) is untrusted

How do we get more?

Composition guarantees for differential privacy



Fact:
Arbitrary composition (sequential and/or parallel) of *k* differentially private algorithms is still differentially private.

Privacy guarantee:
kε-differential privacy

Example: Laplacian Mechanism for multiple queries

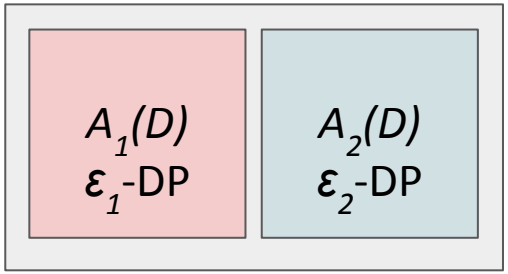
Given queries $q_1,\ldots q_k$:

1. Compute $q_i(D)$, $i=1\ldots k$
2. Output $q_i(D)$ + Lap(k/ε)

Answer k queries by adding Lap(k/ε) to each answer; Gives ε-DP *over all*

But: Becomes *useless* when $k > n = |D|$

Base case of composition

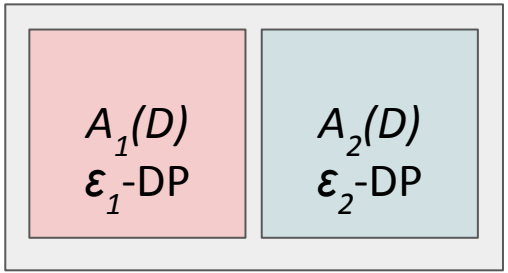


Claim:
The algorithm $A(D) = (A_1(D), A_2(D))$
is $(\varepsilon_1+\varepsilon_2)$-differentially private.

Proof *(for discrete probability distributions)*:
Fix neighboring D, D' and any two outputs r_1 in the range of A_1, and r_2 in the range of A_2.

$$\frac{\mathbb{P}\{A(D) = (r_1, r_2)\}}{\mathbb{P}\{A(D') = (r_1, r_2)\}}$$

Base case of composition

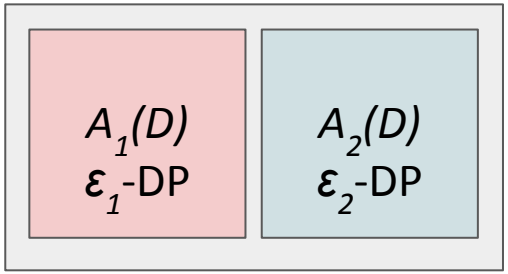


Claim:
The algorithm $A(D) = (A_1(D), A_2(D))$
is $(\varepsilon_1+\varepsilon_2)$-differentially private.

Proof *(for discrete probability distributions)*:
Fix neighboring D, D' and any two outputs r_1 in the range of A_1, and r_2 in the range of A_2.

$$\frac{\mathbb{P}\{A(D) = (r_1, r_2)\}}{\mathbb{P}\{A(D') = (r_1, r_2)\}} = \frac{\mathbb{P}\{A_1(D) = r_1\}\mathbb{P}\{A_2(D) = r_2\}}{\mathbb{P}\{A_1(D') = r_1\}\mathbb{P}\{A_2(D') = r_2\}}$$

Base case of composition



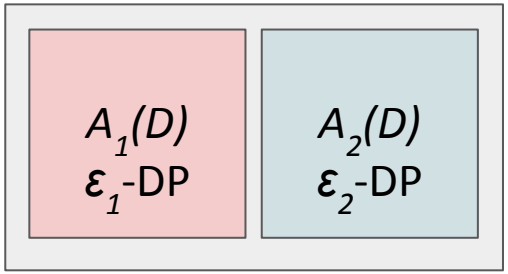
Claim:
The algorithm $A(D) = (A_1(D), A_2(D))$
is $(\varepsilon_1+\varepsilon_2)$-differentially private.

Proof *(for discrete probability distributions)*:
Fix neighboring D, D' and any two outputs r_1 in the range of A_1, and r_2 in the range of A_2.

$$\frac{\mathbb{P}\{A(D) = (r_1, r_2)\}}{\mathbb{P}\{A(D') = (r_1, r_2)\}} = \frac{\mathbb{P}\{A_1(D) = r_1\}\mathbb{P}\{A_2(D) = r_2\}}{\mathbb{P}\{A_1(D') = r_1\}\mathbb{P}\{A_2(D') = r_2\}}$$

$$= \left(\frac{\mathbb{P}\{A_1(D) = r_1\}\}}{\mathbb{P}\{A_1(D') = r_1\}\}}\right)\left(\frac{\mathbb{P}\{A_2(D) = r_2\}}{\mathbb{P}\{A_2(D') = r_2\}}\right)$$

Base case of composition

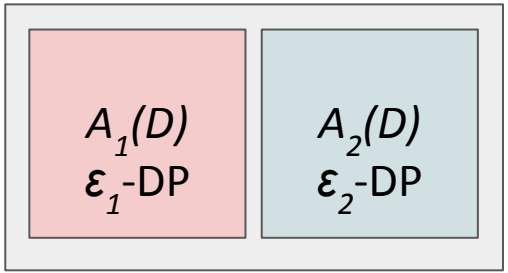


Claim:
The algorithm $A(D) = (A_1(D), A_2(D))$
is $(\varepsilon_1+\varepsilon_2)$-differentially private.

Proof *(for discrete probability distributions)*:
Fix neighboring D, D' and any two outputs r_1 in the range of A_1, and r_2 in the range of A_2.

$$\frac{\mathbb{P}\{A(D) = (r_1, r_2)\}}{\mathbb{P}\{A(D') = (r_1, r_2)\}} = \frac{\mathbb{P}\{A_1(D) = r_1\}\mathbb{P}\{A_2(D) = r_2\}}{\mathbb{P}\{A_1(D') = r_1\}\mathbb{P}\{A_2(D') = r_2\}}$$
$$= \left(\frac{\mathbb{P}\{A_1(D) = r_1\}}{\mathbb{P}\{A_1(D') = r_1\}}\right)\left(\frac{\mathbb{P}\{A_2(D) = r_2\}}{\mathbb{P}\{A_2(D') = r_2\}}\right)$$
$$\leq \exp(\epsilon_1)\exp(\epsilon_2)$$

Base case of composition

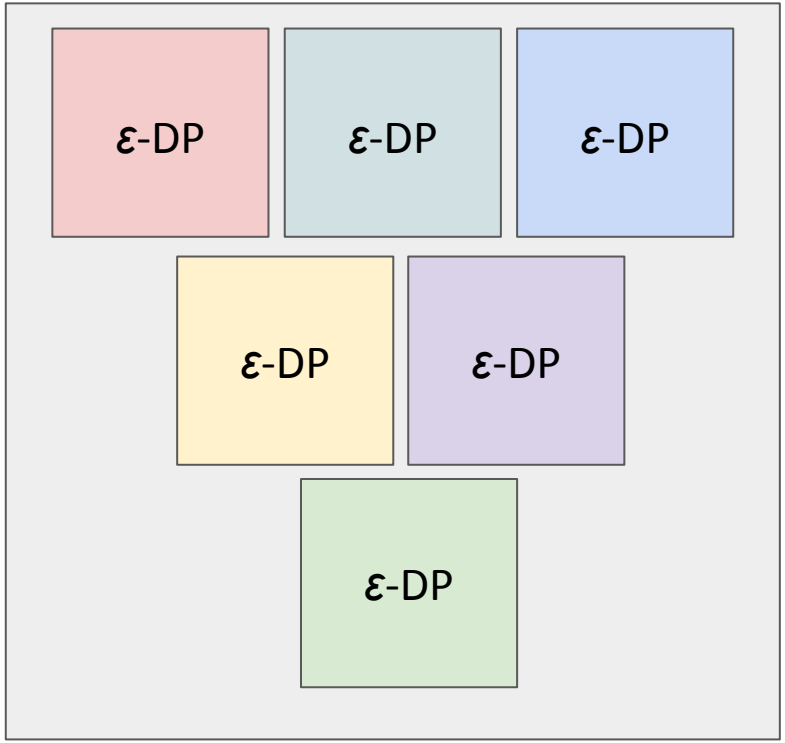


Claim:
The algorithm $A(D) = (A_1(D), A_2(D))$
is $(\varepsilon_1+\varepsilon_2)$-differentially private.

Proof *(for discrete probability distributions)*:
Fix neighboring D, D' and any two outputs r_1 in the range of A_1, and r_2 in the range of A_2.

$$\begin{aligned}
\frac{\mathbb{P}\{A(D) = (r_1, r_2)\}}{\mathbb{P}\{A(D') = (r_1, r_2)\}} &= \frac{\mathbb{P}\{A_1(D) = r_1\}\mathbb{P}\{A_2(D) = r_2\}}{\mathbb{P}\{A_1(D') = r_1\}\mathbb{P}\{A_2(D') = r_2\}} \\
&= \left(\frac{\mathbb{P}\{A_1(D) = r_1\}\}}{\mathbb{P}\{A_1(D') = r_1\}\}}\right)\left(\frac{\mathbb{P}\{A_2(D) = r_2\}}{\mathbb{P}\{A_2(D') = r_2\}}\right) \\
&\leq \exp(\epsilon_1)\exp(\epsilon_2) \\
&= \exp(\epsilon_1 + \epsilon_2)
\end{aligned}$$

Composition guarantees for differential privacy



General statement follows from base case by induction over acyclic computation graph

Need one additional fact:

Postprocessing property. Any function applied to the output of a differentially private algorithm is differentially private with the same privacy parameter.

How do we get even more?

Relaxation of differential privacy

Two data sets D,D' are called *neighboring* if they differ in at most one data record.

Approximate Differential Privacy:
A randomized algorithm $A(D)$ is (ε, δ)-differentially private if for all neighboring data sets D,D' and all events S:

$$\mathbb{P}\{A(D) \in S\} \leq \exp(\epsilon)\mathbb{P}\{A(D') \in S\} + \delta$$

Think: $\varepsilon = 0.01$ and $\delta = o(1/|D|)$

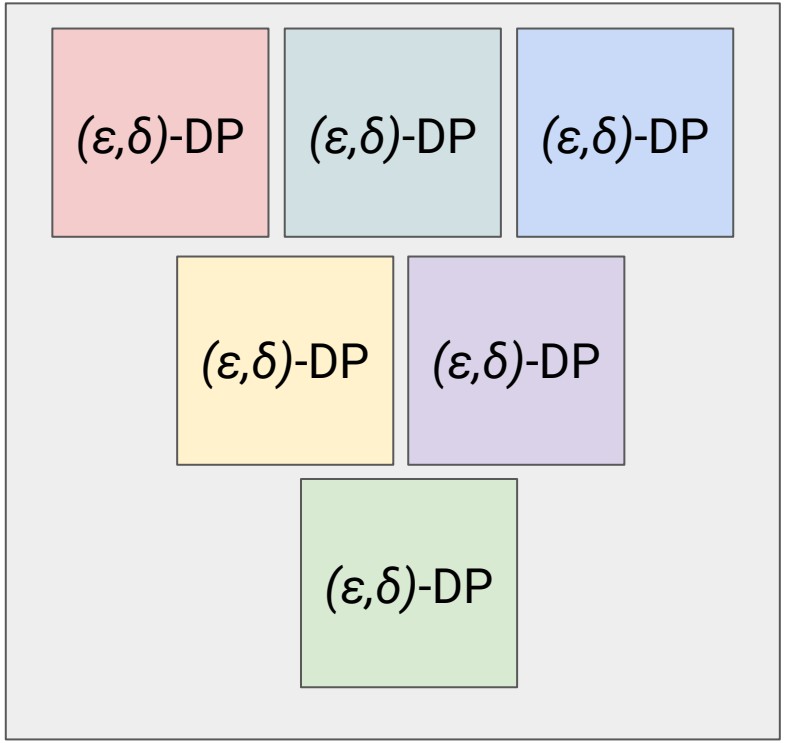
Notes

Like differential privacy but with *δ chance of failure*

Need δ to be less than 1/|D|. Why?

> Consider A(D) that randomly selects single *x in D* and outputs *x*
>
> This algorithm satisfies (0, 1/|*D*|)-differential privacy, but <u>always</u> compromises somebody's privacy.

Strong composition theorem



Strong composition theorem (informal):
Assume $\varepsilon << k^{-1/2}$ and δ negligible. Then, composition satisfies
$(k^{1/2}\varepsilon, k\delta)$-*differential privacy*

Main point:
Privacy loss factor $k^{1/2}$ instead of k

Can be shown to be best possible using signal boosting approach from last lecture.

Law of fundamental information recovery still kicks in!

Can we get even more in some cases?

Multiplicative Weights Approach

[*H*-Rothblum'10, Gupta-*H*-Roth-Ullman'11, *H*-McSherry-Ligett'12]

Handles huge query sets:

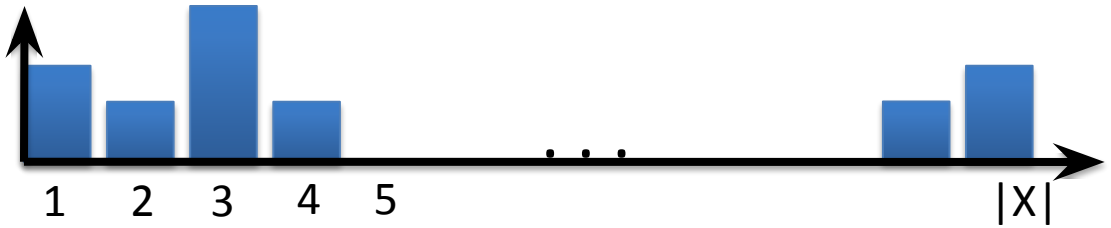
Small error for any $k < 2^{o(n)}$

Previous related work achieving similar results with different ideas:
Blum-Ligett-Roth 08, Dwork-Naor-Reingold-Rothblum-Vadhan 09,
Dwork-Rothblum-Vadhan 10, Roth-Roughgarden 10

Histogram View

Represent D as vector with |X| coordinates, one for each possible data point

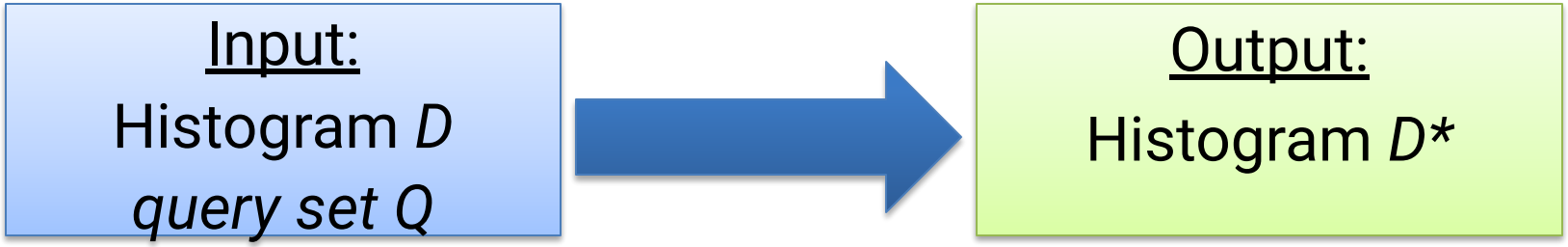
- coordinate *i* = count number of times *i* appears in *D*



Normalized histogram = distribution over X

Statistical query *q* becomes vector in histogram space

What we want to do



Requirements:

1. *D* satisfies differential privacy*
2. $|q(D)-q(D^*)|$ small for all *q* in *Q*

Basic Algorithm

Input: Data set D, query set Q

Let D_0 be uniform histogram

For $t=1$ until $t=T$:

1. Find "bad" query q where $|\, q(D) - q(D_{t-1}) \,|$ *too large*
2. Improve histogram using **multiplicative weights update rule**:

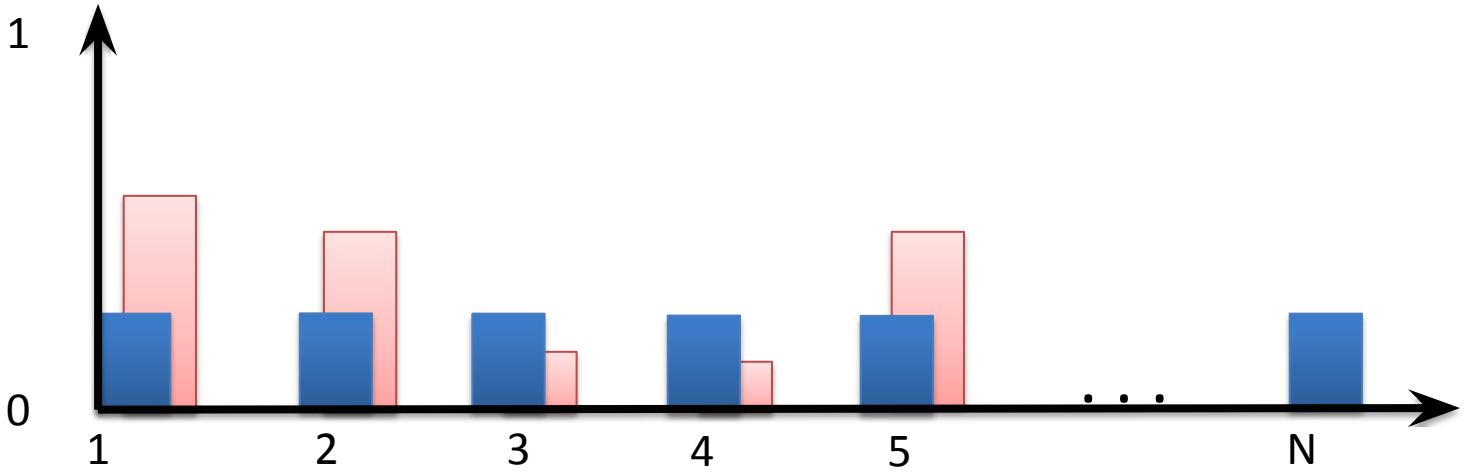
   $D_t = \text{MWUpdate}(D_{t-1}, q)$

Output: $D^* = D_T$

Each step satisfies differential privacy!

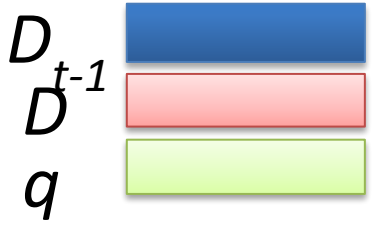
Analysis: By multiplicative weights magic, process runs out of bad queries quickly!

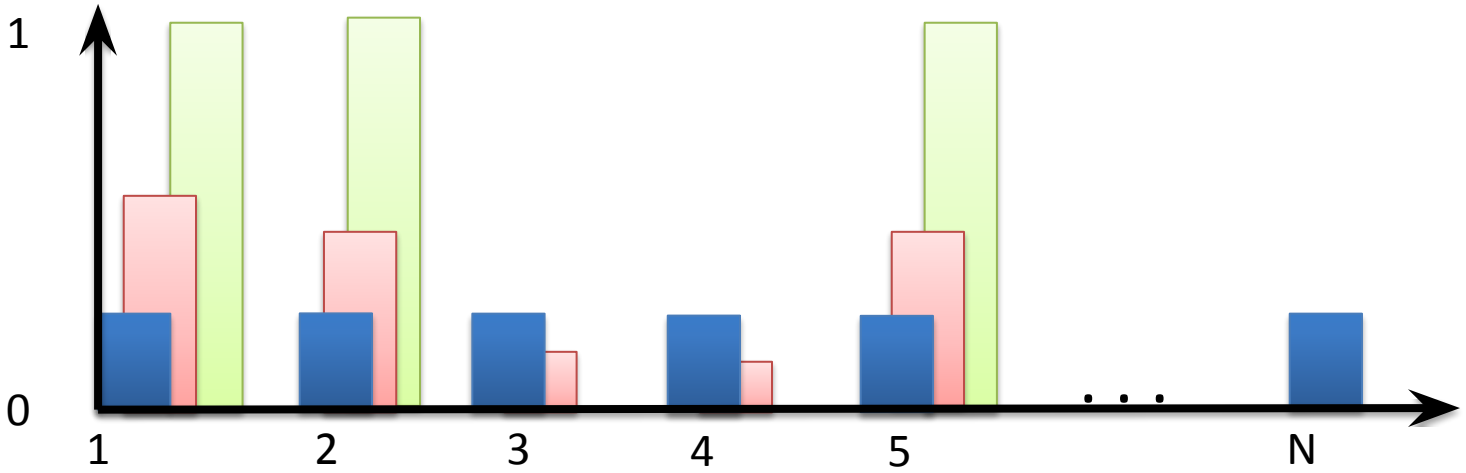
MWUpdate(D_{t-1}, q)



At step t

MWUpdate(D_{t-1}, q)

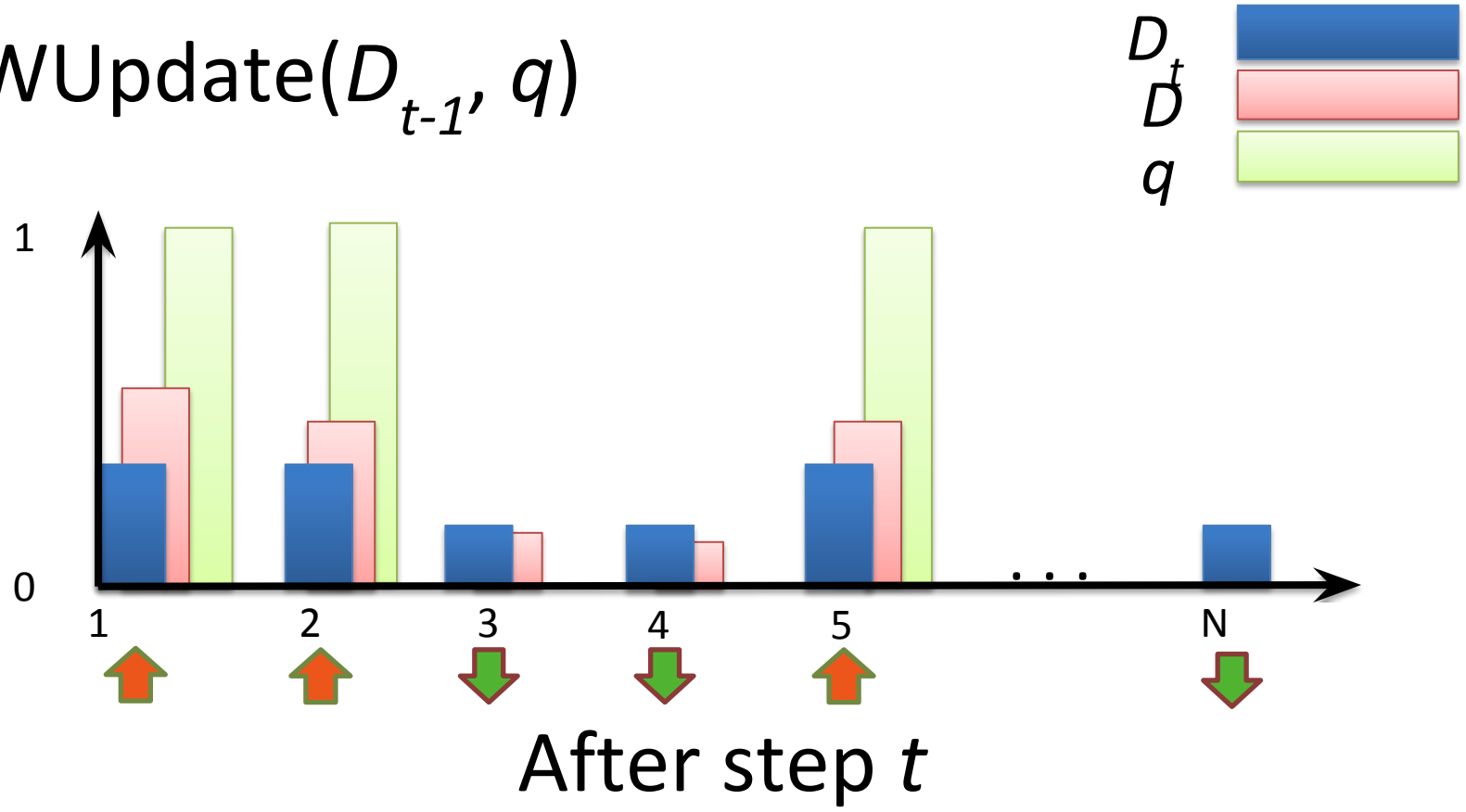




At step t

Suppose $q(D_{t-1}) << q(D)$

MWUpdate(D_{t-1}, q)



After step t

Implementation

- Computational bottleneck: Enumerating over all $|X|$ coordinates ($|X|$ can be exponential)
  - necessary in worst-case due to hardness result [DNRRV09,Ull13]
- Parallelizable, scalable implementation with heuristic tweaks [*H*-McSherry-Ligett, NIPS12]
  - https://github.com/mrtzh/PrivateMultiplicativeWeights.jl
- Recent work attempts to do similar things using generative adversarial networks (GANs)

Some applications of statistical queries

Recall: Statistical queries

Database D subset of some universe X

Example: $X = \{0,1\}^d$ (binary d-tuples)

Statistical query: predicate $q : X \to [0, 1]$

Answer $q(D) := \Sigma_{x \text{ in } D}\, q(u)$

between 0 and $n = |D|$

- Example: "How many people in D smoke and have cancer?"

Fact: For neighboring D, D': $|q(D) - q(D')| \leq 1$

Differentially private gradient descent

Suppose we want to train model on sensitive data using gradient descent

Update rule: $w_{t+1} = w_t - \alpha \nabla \text{loss}(w_t, D)$

Example squared loss: $\text{loss}(w_t, D) = \Sigma_{(x,y) \text{ in } D} (\langle w_t, x \rangle - y)^2$

Fact: Each coordinate of $\nabla \text{loss}(w_t, D)$ is a statistical query.

Hence, we can make gradient descent differentially private using what we have!

Differential privacy in the wild

Some context

Initially many thought differential privacy was a theoretical toy

> *“Adds too much noise”, “Nobody wants randomization”, “Will never catch on”, “Laplace, LOL”, “The lawyers won’t approve this.”, “Wake me up when Google uses it.”, “My statistics are already private.”*

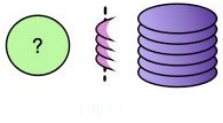
But over the last 15 years many worked hard to put differential privacy in practice

> There are many real and significant challenges in doing so

Privacy Integrated Queries (PINQ)

Established: June 22, 2009

Overview Publications Downloads



Privacy Integrated Queries is a LINQ-like API for computing on privacy-sensitive data sets, while providing guarantees of differential privacy for the underlying records. The research project is aimed at producing a simple, yet expressive language about which differential privacy properties can be efficiently reasoned and in which a rich collection of analyses can be programmed.

Substiantial progress has been recently made in the rigorous treatment of privacy-preserving data analysis, in the form of Differential Privacy: a formal and achievable requirement that a computation not reveal even the presence of any one individual in its input. As powerful as this privacy criterion is, its formal nature challenges data analysts and data providers to design new analyses and verify their privacy properties without the help of differential privacy experts.

https://www.microsoft.com/en-us/research/project/privacy-integrated-queries-pinq/

Early implementation of differential privacy spearheaded by Frank McSherry

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

https://github.com/google/rappor

Learning statistics with privacy, aided by the flip of a coin

October 30, 2014

Cross-posted on the Research Blog and the Chromium Blog

At Google, we are constantly trying to improve the techniques we use to protect our users' security and privacy. One such project, RAPPOR (Randomized Aggregatable Privacy-Preserving Ordinal Response), provides a new state-of-the-art, privacy-preserving way to learn software statistics that we can use to better safeguard our users' security, find bugs, and improve the overall user experience.

Building on the concept of randomized response, RAPPOR enables learning statistics about the behavior of users' software while guaranteeing client privacy. The guarantees of differential privacy, which are widely accepted as being the strongest form of privacy, have almost never been used in practice despite intense research in academia. RAPPOR introduces a practical method to achieve those guarantees.

Large-scale system implemented as part of Google Chrome.

First major industry product feature involving differential privacy

Google is open-sourcing a tool for data scientists to help protect private information

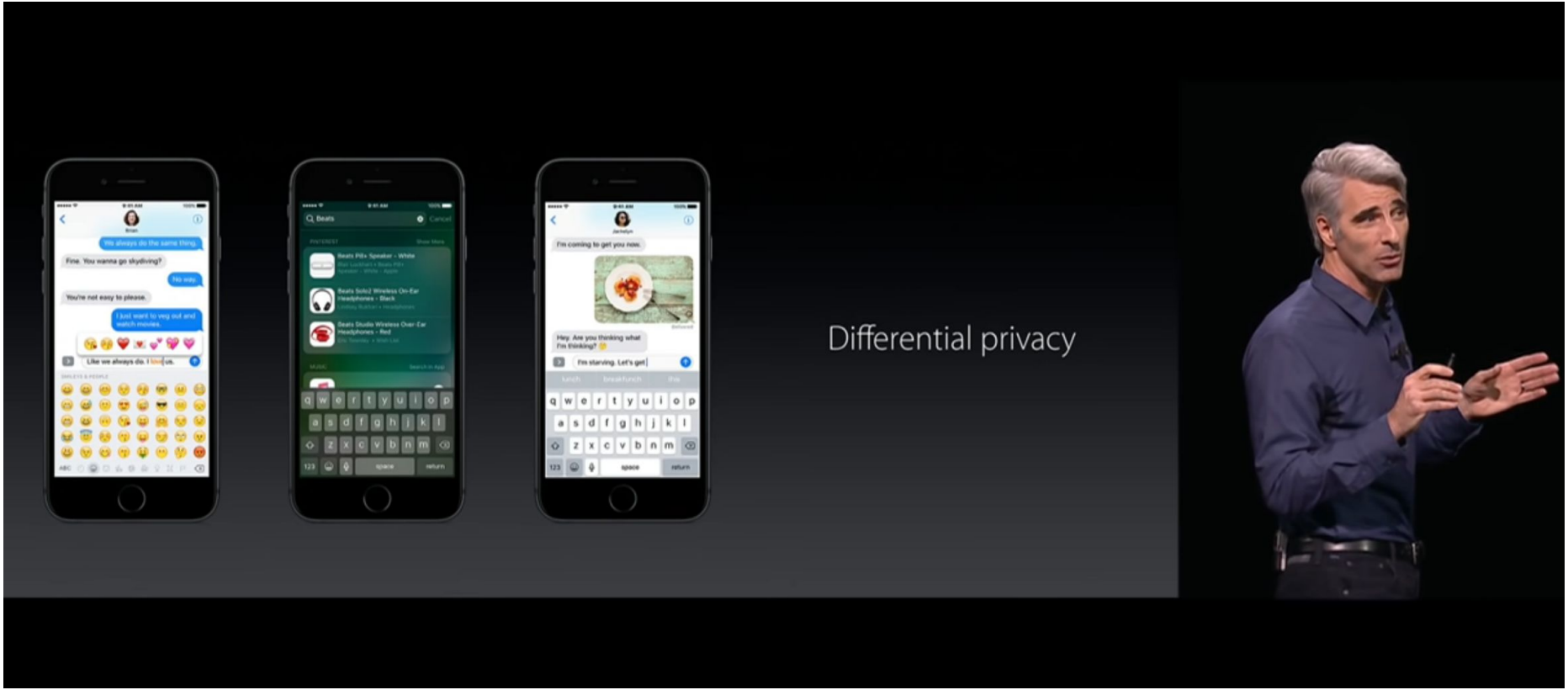
Google is making differential privacy available to anyone

By Nick Statt | @nickstatt | Sep 5, 2019, 6:00am EDT

SHARE

https://github.com/google/differential-privacy

Source:
https://www.theverge.com/2019/9/5/20850465/google-differential-privacy-open-source-tool-privacy-data-sharing



The Verge: “It was probably the most bewildering part of Apple’s [2016] WWDC Keynote: in the middle of a rundown of fancy new products arriving with iOS 10, Craig Federighi stopped to talk about abstract mathematics. He was touting differential privacy, a statistical method that’s become a valuable tool for protecting user data.” https://www.theverge.com/2016/6/17/11957782/apple-differential-privacy-ios-10-wwdc-2016

A privacy-preserving system

Apple has adopted and further developed a technique known in the academic world as *local differential privacy* to do something really exciting: gain insight into what many Apple users are doing, while helping to preserve the privacy of individual users. It is a technique that enables Apple to learn about the user community without learning about individuals in the community. Differential privacy transforms the information shared with Apple before it ever leaves the user's device such that Apple can never reproduce the true data.

From: https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12

Jun Tang
University of Southern California
juntang@usc.edu

Aleksandra Korolova
University of Southern California
korolova@usc.edu

Xiaolong Bai
Tsinghua University
bxl12@mails.tsinghua.edu.cn

Xueqiang Wang
Indiana University
xw48@indiana.edu

Xiaofeng Wang
Indiana University
xw7@indiana.edu

"In June 2016, Apple announced that it will deploy differential privacy for some user data collection in order to ensure privacy of user data, even from Apple. The details of Apple's approach remained sparse. Although several patents have since appeared hinting at the algorithms that may be used to achieve differential privacy, they did not include a precise explanation of the approach taken to privacy parameter choice. Such choice and the overall approach to privacy budget use and management are key questions for understanding the privacy protections provided by any deployment of differential privacy."
https://arxiv.org/abs/1709.02753

Differential Privacy in the Real World: The 2018 End-to-End Census Test

John M. Abowd
Chief Scientist and Associate Director for Research and Methodology
U.S. Census Bureau
American Association for the Advancement of Science
Annual Meeting Sunday, February 17, 2019 8:00-9:30

United States Census Bureau

U.S. Department of Commerce
Economics and Statistics Administration
U.S. CENSUS BUREAU
census.gov

The views expressed in this talk are my own and not those of the U.S. Census Bureau.

Formal Privacy:
Making an Impact at Large Organizations

Deploying Differential Privacy for the 2020 Census of Population and Housing

Simson L. Garfinkel
Senior Scientist, Confidentiality and Data Access
U.S. Census Bureau

July 31, 2019
JSM 2019

The views in this presentation are those of the author, and not those of the U.S. Census Bureau.

Stepping back

There are many challenges with putting differential privacy in practice

- Computational challenges
- Implementation pitfalls
- Political struggles
- Legal and policy hurdles

What we saw today

Differential privacy is a formal privacy notion

- Natural counterfactual interpretation
- Appealing properties (post-processing, composition)
- Rich theory (many beautiful results we didn’t discuss)
- Powerful methods to answer many queries with little noise

No panacea: Fundamental law of information recovery still relevant

Thank you.