Privately Evaluating Untrusted Black-Box Functions

Sofya Raskhodnikova

BOSTON UNIVERSITY

Joint work with:

Ephraim Linder
Boston University

Adam Smith
Boston University

Thomas Steinke
Google DeepMind

Goal

To provide tools for sharing sensitive data

in situations when

the data curator does not know in advance

what questions the (untrusted) analyst will ask about the data

Want:

- an automated way for the analyst to interact with the data

Instead of:

- putting the analyst through background checks and
- monitoring their access to data

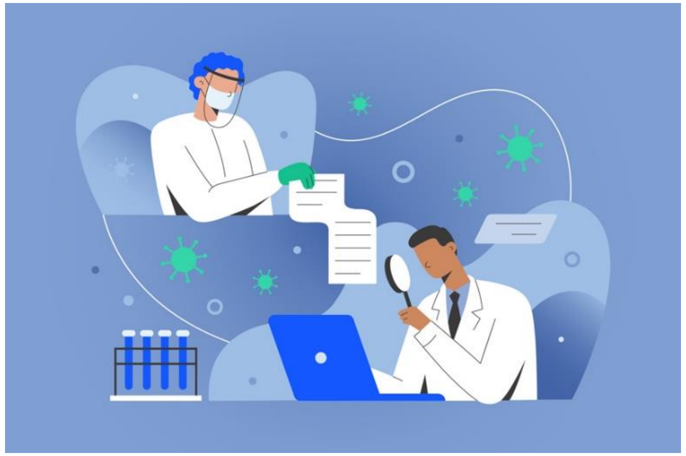
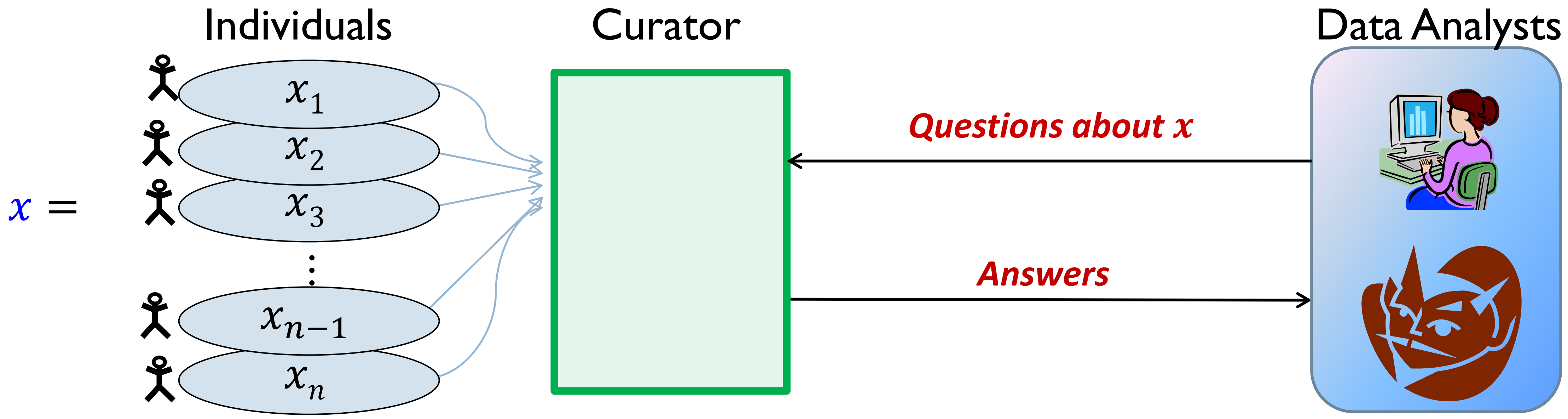


image sources: : https://sharemedix.com/

Private data analysis



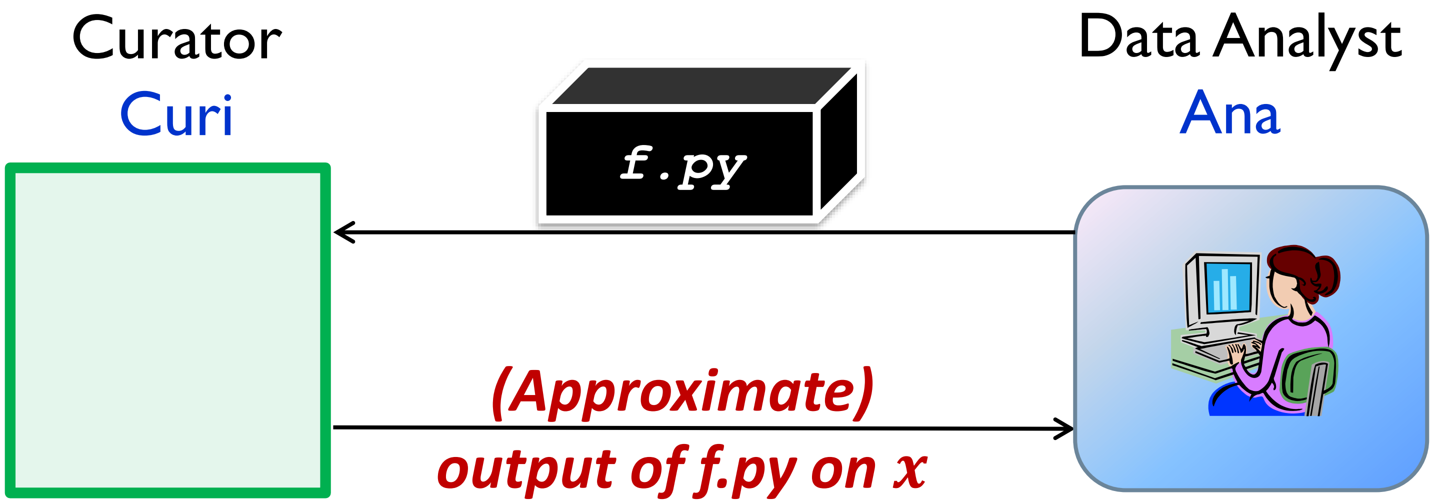
Typical examples: *census, medical studies, data collected by industry…*

Two conflicting goals

- *Protect privacy of individuals:* ***Differential privacy*** [Dwork McSherry Nissim Smith 06]
- *Provide accurate information*

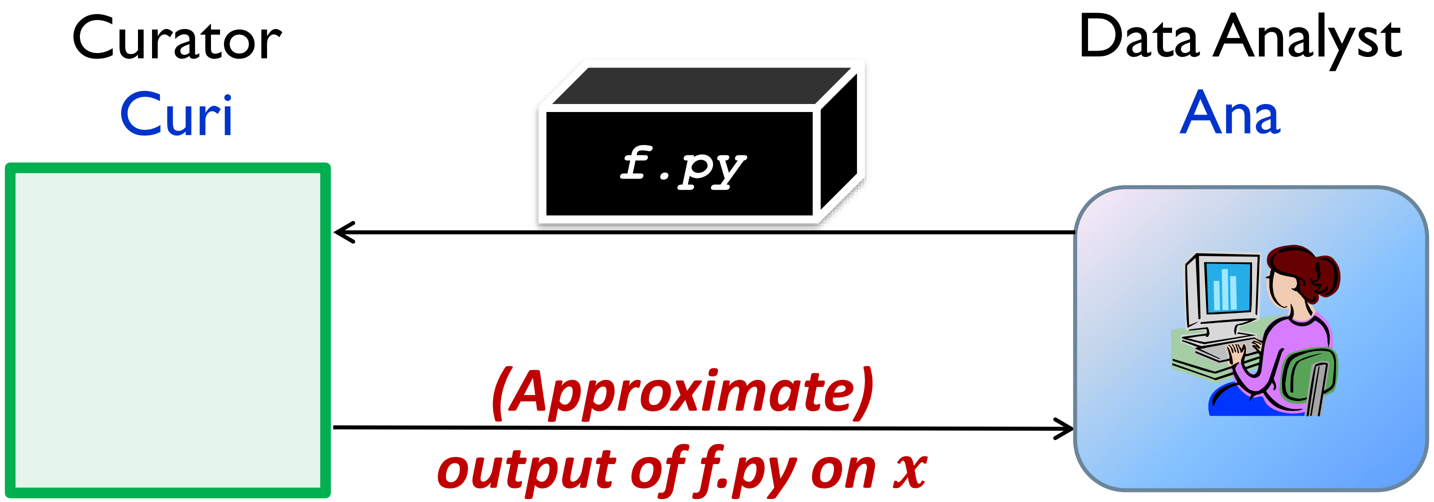
Many techniques developed for releasing *specific* functions of dataset x that are not too \`\`sensitive'' to individual inputs.

The black-box privacy problem [Jha Raskhodnikova 11]



- Ana asks Curi to evaluate her program on the dataset x and send back the output
- The overall algorithm Curi runs to produce the output must be differentially private
- What can Curi do?

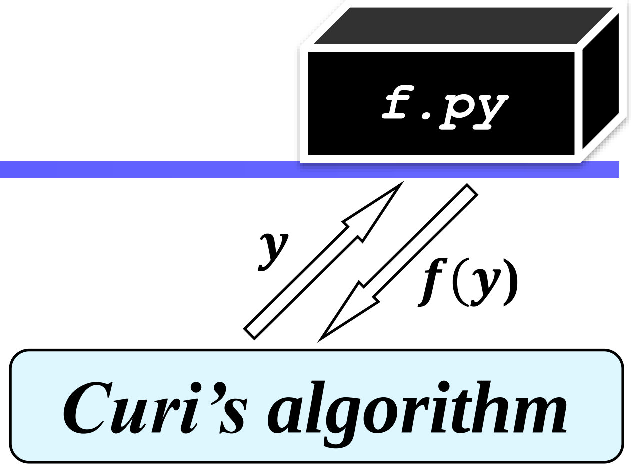
Embracing the black box



What can Curi do?

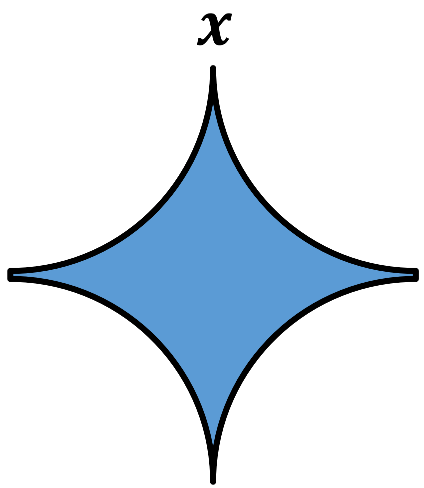
- Inspect the code of f?
    - Curi will run into computationally hard problems
- Ask Ana to implement f using a restricted language?
    - Limits the functions Ana can use
- Query the black box in a few places (e.g., to check sensitivity)?
    - Allows Ana to construct arbitrarily complicated programs
    - Enables Ana to obfuscate her programs

What queries can Curi use?



- Function f can be queried on **any dataset**
  [Jha Raskhodnikova 11, Awasthi Jha Raskhodnikova Molinaro 16, Lange Linder Raskhodnikova Vasilyan]

- Function f can be queried only on **dataset x and its subsets**
  [Kohli Laskowski 23, this work]



Consider actual data of people in x rather than adding hypothetical individuals' data

This restriction allows us to

✓ deal with large (or even infinite) universe for individual data entries

✓ give accurate answers for functions f that behave nicely on x and its subsets, but do strange things on outliers

✓ improve accuracy for functions f that are more ``sensitive'' to additions of data entries than to removals

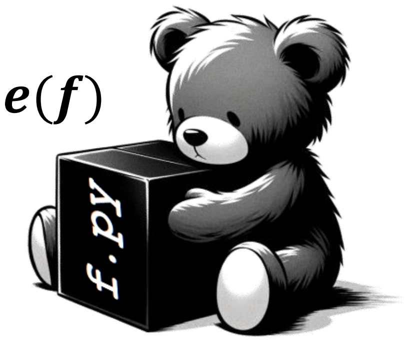
Example: $\max(x_1, \dots, x_n)$ can *increase* arbitrarily under an *addition of* $x_{n+1} \in \mathbb{R}$, but can *decrease* by at most the gap between the largest and the second largest element under a *removal of an entry* x_i.

Information provided by the analyst

Automated sensitivity detection setting [this work]

- The analyst supplies
  - the black-box function f
  - the intended range of f

$\boldsymbol{range}(\boldsymbol{f})$

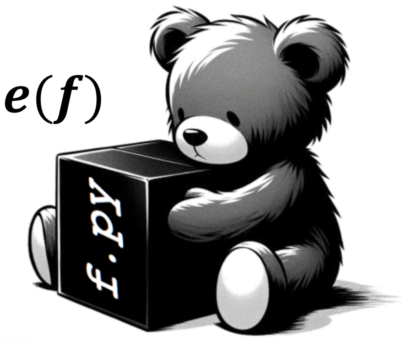


Claimed sensitivity bound setting [Jha Raskhodnikova 11, Awasthi Jha Molinaro Raskhodnikova 16, Kohli Laskowski 23, Lange Linder Raskhodnikova Vasilyan, this work]

- The analyst supplies (in addition to the above)
  - parameters that describe the sensitivity of f

$\boldsymbol{range}(\boldsymbol{f})$

sensitivity parameters



Privacy is guaranteed even if the parameters supplied by the analyst are incorrect

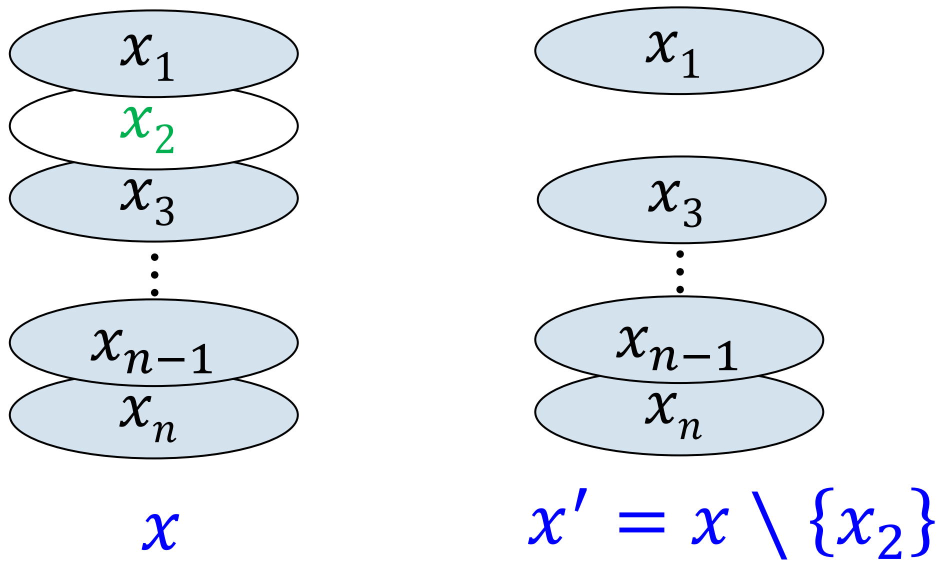
Correct setting of parameters ensures better accuracy

Notions of sensitivity: preliminary definitions

We consider functions $f: \mathcal{U}^* \to \mathbb{R}$, where

- $\mathcal{U}$ is (finite or infinite) universe, where data items come from
- Each dataset is a (multi)-set of items $x_1, \ldots, x_n \in \mathcal{U}$ for some $n \in \mathbb{N}$
- $\mathcal{U}^*$ represents the set of all datasets

Two datasets are ***neighbors*** if one can be obtained from the other by deleting one data item



Notions of sensitivity

We consider functions $f: \mathcal{U}^* \to \mathbb{R}$, where $\mathcal{U}^*$ represents the set of all datasets

Two datasets are ***neighbors*** if one can be obtained from the other by deleting one data item

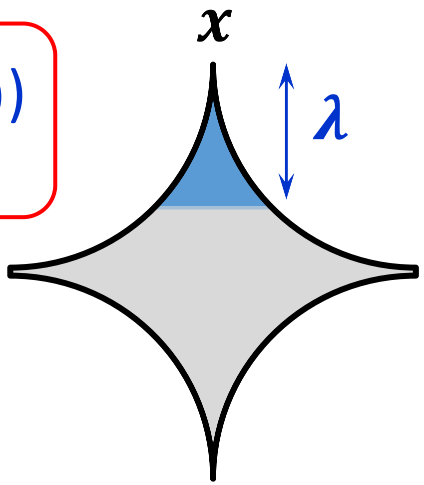
- The global sensitivity of f (denoted GS^f) is

$$\max_{x,x' \text{ neighbors}} |f(x) - f(x')|$$

If $GS^f = c$, then f is called c-Lipschitz.

Example: f is $\max(x_1, \dots, x_n)$
If the universe $\mathcal{U} = [r]$, then $GS^f = r - 1$
If the universe $\mathcal{U} = \mathbb{N}$, then $GS^f = \infty$

For depth $\lambda \in \mathbb{N}$, the λ-down-neighborhood of dataset x (denoted $\mathcal{N}_\lambda^\downarrow(x)$) is the set of all subsets of x of size at least $|x| - \lambda$.



- The down sensitivity of f at depth λ on dataset x (denoted $\mathrm{DS}_\lambda^f(x)$) is

$$\max_{z \in \mathcal{N}_\lambda^\downarrow(x)} |f(x) - f(z)|$$

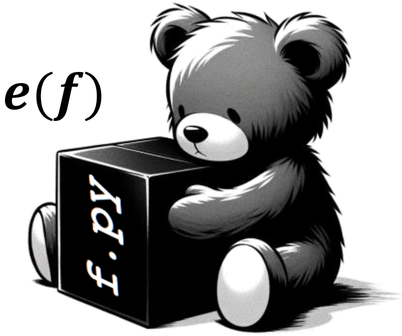
How much can the value of f change if at most λ people are removed from x?

Example: f is $\max(x_1, \dots, x_n)$, $x = \{0,1,1,1,2,2,2,3\}$
Then $DS_3^f(x) = 1$

Our contributions

Automated sensitivity detection setting

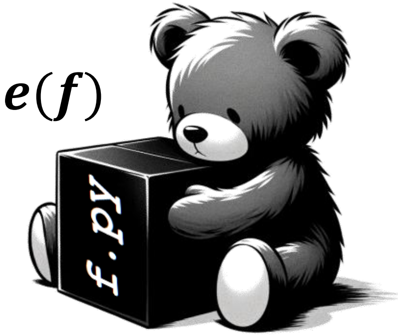
- Introduce the setting
- Give a privacy mechanism and a tight lower bound for $f\colon \mathcal{U}^* \to \mathbb{R}$



on query complexity and accuracy

Claimed sensitivity bound setting

- First guarantees in terms of down sensitivity
- First accuracy guarantees with **no** dependence on the universe size
- Tight upper and lower bounds for $f\colon \mathcal{U}^* \to \mathbb{R}$
- Reinterpretation & analysis of other constructions in our framework



[Jha Raskhodnikova 11, Lange Linder Raskhodnikova Vasilyan]
- gave guarantees in terms of global sensitivity and have dependence on $|\mathcal{U}|$
- Used different techniques, based on local Lipschitz filters [Saks Seshadhri 10]

[Kohli Laskowski 23] designed the first black-box private algorithm with queries in $\mathcal{N}_\lambda^{\downarrow}(x)$ and analyzed its privacy (but not accuracy)

Plan

- Background on differential privacy and definition of privacy wrappers
- Quantitative statement of results
- Privacy wrapper for the automated sensitivity detection setting
- Extension to graphs and other types datasets

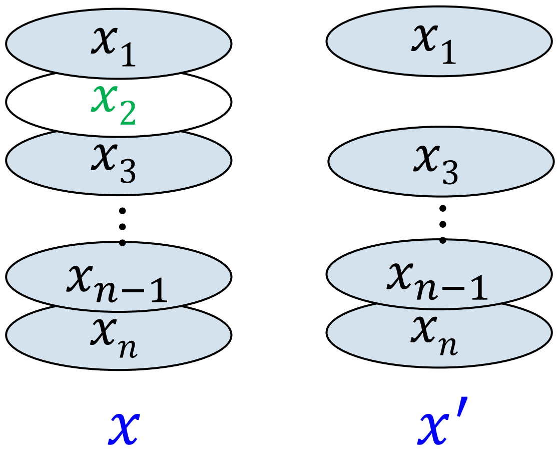
Differential privacy [Dwork McSherry Nissim Smith 06]

Intuition: An algorithm is **differentially private (DP)** if its output distribution is roughly the same for all pairs of ***neighbor datasets.***

Think: The output distribution is roughly the same whether or not your data is in the dataset.

An algorithm $\mathcal{A}$ is **(ϵ, δ)-differentially private** if
for all pairs of ***neighbors*** $\boldsymbol{x}, \boldsymbol{x}'$ and all possible sets of outputs S:

$$\mathbf{Pr}[\mathcal{A}(\boldsymbol{x}) \in \boldsymbol{S}] \leq \boldsymbol{e}^{\epsilon}\, \mathbf{Pr}[\mathcal{A}(\boldsymbol{x}') \in \boldsymbol{S}] + \boldsymbol{\delta}$$



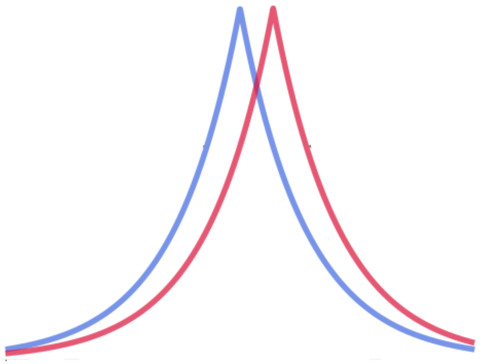
If $\delta = 0$, we say $\mathcal{A}$ is purely DP

Basic $(\epsilon, 0)$-differentially private mechanisms

- Laplace Mechanism (for approximating $f: \mathcal{U}^* \to \mathbb{R}$)

Given x, return $f(x) + Z$ for $Z \sim Laplace(\sigma)$ where $\sigma = O\left(\frac{GS^f}{\varepsilon}\right)$

Previous work on the black-box DP problem tries to emulate this mechanism (for the case when the claimed GS^f is correct)



- Exponential Mechanism (for approximating $f: \mathcal{U}^* \to \mathcal{Y}$)

Define a score function $score_x(y)$ for all $y \in \mathcal{Y}$, and let Δ be its sensitivity:

$$|score_x(y) - score_{x'}(y)| \le \Delta \text{ for all } y \in \mathcal{Y} \text{ and all neighbor datasets } x, x'$$

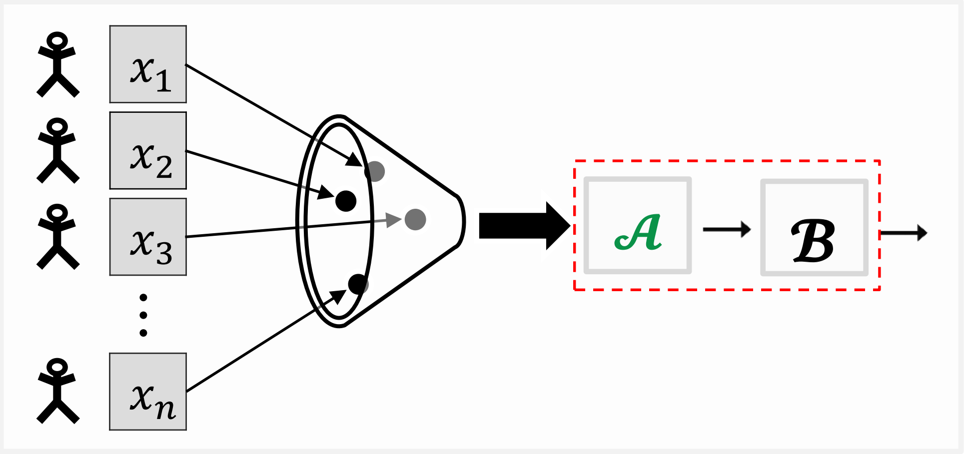
Given x, return each $y \in \mathcal{Y}$ with probability proportional to $\exp\left(\frac{\epsilon \cdot score_x(y)}{2\Delta}\right)$

Utility: ExponentialMechanism(x) returns $\hat{y}$ satisfying: for all $\beta \in (0,1)$,

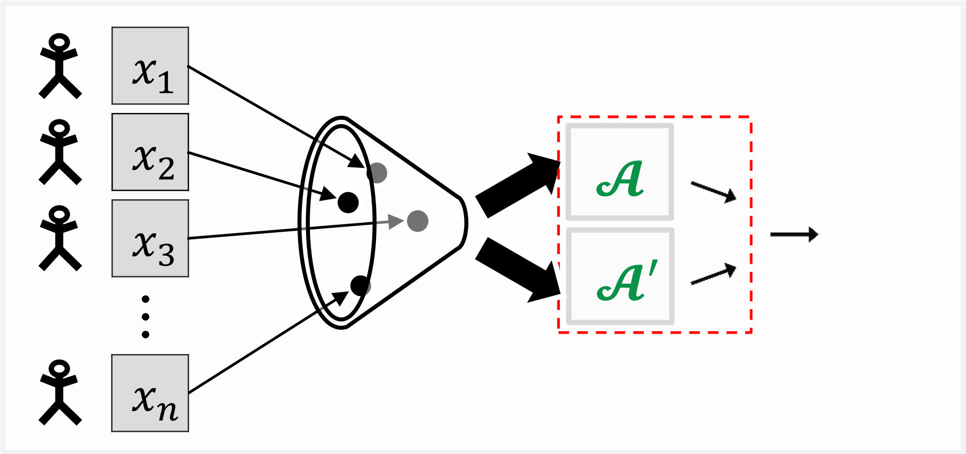
$$score_x(\hat{y}) \le \min_y score_x(y) + \frac{2\Delta}{\epsilon} \ln \frac{|\mathcal{Y}|}{\beta} \quad \text{with probability} \ge 1 - \beta$$

Properties of Differential Privacy (DP)

- Post-processing: **If** algorithm $\mathcal{A}$ is (ϵ, δ)-DP and $\mathcal{B}$ is any randomized algorithm **then** $\mathcal{B}\big(\mathcal{A}(x)\big)$ is (ϵ, δ)-DP



- Composition: **If** algorithms $\mathcal{A}$ and $\mathcal{A}'$ are (ϵ, δ)-DP **then** the algorithm that outputs $\big(\mathcal{A}(x), \mathcal{A}'(x)\big)$ is $(2\epsilon, 2\delta)$-DP



Privacy wrapper

[Kohli Laskowski 23]

An algorithm $\mathcal{W}$ that

- gets and input $x \in \mathcal{U}^*$ and query access to a function f on $\mathcal{U}^*$;
- produces an output in range$(f) \cup \{\perp\}$

is an (ϵ, δ)-privacy wrapper if $\mathcal{W}^f$ is (ϵ, δ)-DP for every function f

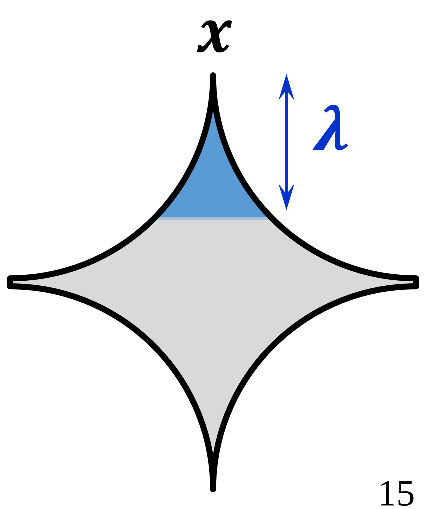
and potentially some additional parameters

and all settings of the parameters

- Algorithm $\mathcal{W}$ is λ-down local if for all functions f and datasets x, the queries of $\mathcal{W}^f$ on input x are contained in $\mathcal{N}_\lambda^{\downarrow}(x)$

makes $O(|x|^\lambda)$ queries

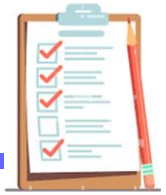
- Algorithm $\mathcal{W}$ is (α, β)-accurate for a function f and a dataset x if $\Pr\left[\left|\mathcal{W}^f(x) - f(x)\right| \geq \alpha\right] \leq \beta$



Example: For each $\beta \in (0,1]$, Laplace mechanism is (α, β)-accurate for all functions with $GS^f = 1$ and all datasets with $\alpha = O\left(\frac{\ln 1/\beta}{\varepsilon}\right)$

- It is not a privacy wrapper, since it is not private when the parameter GS^f is not set correctly

Plan



- ✓ Background on differential privacy and definition of privacy wrappers

- Quantitative statement of results

- Privacy wrapper for the automated sensitivity detection setting

- Extension to graphs and other types datasets

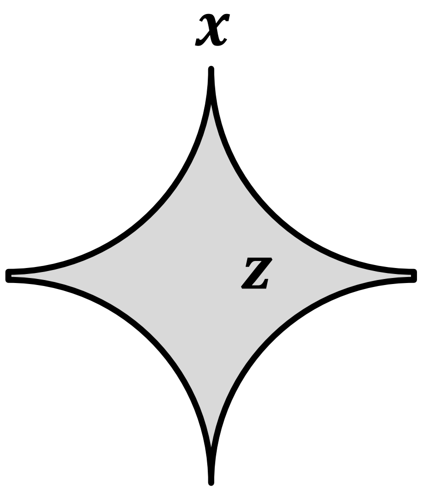
Our results for the automated sensitivity detection setting

The starting point for our algorithm is the Shifted Inverse (ShI) mechanism [Fang Dong Yi 22]

- It is an $(\epsilon, 0)$-DP algorithm for releasing a value of a monotone function

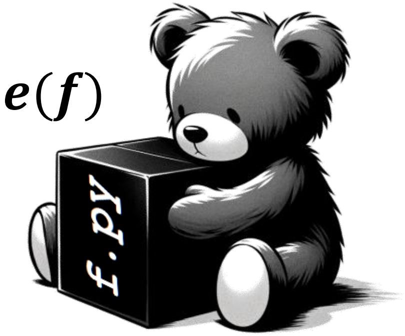
A function $f: \mathcal{U}^* \to \mathbb{R}$ is monotone if

$f(x) \geq f(z)$ for all $x, z \in \mathcal{U}^*$ such that $z \subset x$



$f(x) \geq f(z)$

$\boldsymbol{range(f)}$



Our results for the automated sensitivity detection setting

The starting point for our algorithm is the Shifted Inverse (ShI) mechanism [Fang Dong Yi 22]

- It is **not** a privacy wrapper, because it is private only for monotone functions.
- It works for real-valued functions with a **finite** range $\mathcal{Y} \subset \mathbb{R}$.

Algorithm	**Privacy**		**Accuracy α**	**Down locality λ**
[Fang Dong Yi 22]	$(\epsilon, 0)$-DP	only for monotone functions	down sensitivity at depth λ, $\mathrm{DS}_\lambda^f(x)$	$\lambda_{(\epsilon,0)} := O\left(\frac{1}{\epsilon}\log\frac{\lvert\mathcal{Y}\rvert}{\beta}\right)$
Modified ShI	(ϵ, δ)-DP			$\lambda_{(\epsilon,\delta)} := \frac{1}{\epsilon}\log\frac{1}{\delta} \cdot 2^{O(\log^* \lvert\mathcal{Y}\rvert)}$
Generalized ShI	(ϵ, δ)-DP	all functions	$\mathrm{DS}_\lambda^f(x)$	$\min\left(\lambda_{(\epsilon,0)}, \lambda_{(\epsilon,\delta)}\right)$
Lower bound	(ϵ, δ)-DP	all functions	$\mathrm{DS}_\lambda^f(x)$ $\Rightarrow$	$\Omega\left(\frac{1}{\epsilon}\log\min\left(\frac{\lvert\mathcal{Y}\rvert}{\beta}, \frac{1}{\delta}\right)\right)$

Our results for the claimed sensitivity bound setting

sensitivity
c

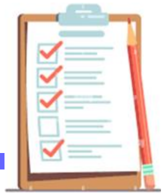
Reinterpretation & analysis of other constructions:

- We reinterpret the Lipschitz extension of [Cummings Durfee 20] as a privacy wrapper
- We analyze the accuracy of TAHOE by [Kohli Laskowski 23]

Algorithm	**Privacy**	**Accuracy assumption**	**Accuracy α**	**Down locality λ**
[Cummings Durfee 20]	$(\epsilon, 0)$-DP	c-Lipschitz on $\mathcal{N}_\lambda^{\downarrow}(x)$	$\frac{c}{\epsilon}$	$\lvert x \rvert$
TAHOE	(ϵ, δ)-DP		$O\left(\frac{c}{\epsilon^2}\log\frac{1}{\delta}\right)$	$O\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right)$
Subset Extension	(ϵ, δ)-DP	as above	$O\left(\frac{c}{\epsilon}\right)$	$O\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right)$
Lower bounds	(ϵ, δ)-DP	as above	[Ghosh Roughgarden Sundararajan 09] c/ϵ, e.g., for $f(x) = \lvert x \rvert$	$\Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right)$

The two lower bounds hold separately.

Plan



✓ Background on differential privacy and definition of privacy wrappers

✓ Quantitative statement of results

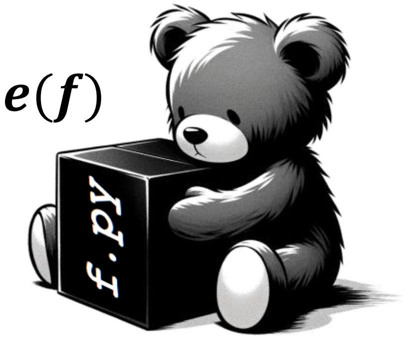
- Privacy wrapper for the automated sensitivity detection setting

    1. ShI mechanism [Fang Dong Yi 22] for monotone functions

    2. Modified ShI (with better dependence on r, the size of the range)

    3. From monotone to general functions

$\boldsymbol{range(f)}$



- Extension to graphs and other types datasets

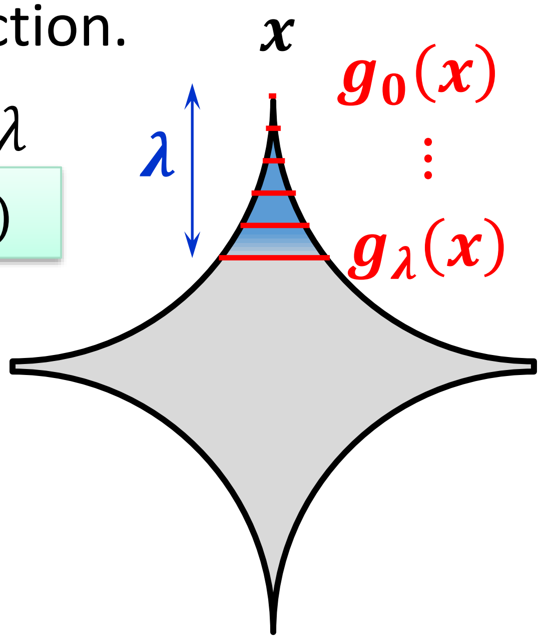
ShI mechanism [Fang Dong Yi 22]

- Let $g: \mathcal{U}^* \to \mathcal{Y}$, where $\mathcal{Y}$ is a finite subset of $\mathbb{R}$, be a monotone function.
- Define $g_j(x) = \min\{g(z): z \in \mathcal{N}_\lambda^{\downarrow}(x)\}$ for each depth $j = 0,1, \dots, \lambda$ and a sequence $\vec{g}(x) = (g_0(x), g_1(x), \dots, g_\lambda(x))$

$\boldsymbol{g_0(x) = g(x)}$

- For each answer $y \in \mathcal{Y}$, define $score_x(y)$ = the smallest number of $g_j(x)$ values that must be changed in $\vec{g}(x)$ to make y the median of the resulting sequence

Used in the exponential mechanism for the median

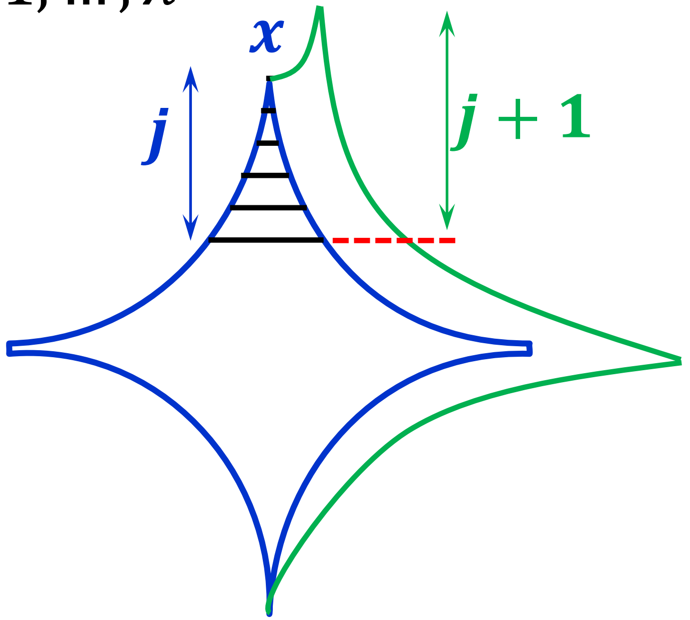


ShI (Input: dataset x, privacy parameter $\epsilon > 0$, failure probability β, finite range $\mathcal{Y}$; query access to a *monotone* function $g: \mathcal{U}^* \to \mathcal{Y}$)

1. Set $\lambda = \Theta\left(\frac{1}{\epsilon} \log \frac{|\mathcal{Y}|}{\beta}\right)$ and compute $\vec{g}(x)$
2. Compute the scores $score_x(y)$ for all $y \in \mathcal{Y}$
3. Return: the output of the exponential mechanism run with these scores

ShI mechanism: analysis [Fang Dong Yi 22]

- Let $g: \mathcal{U}^* \to \mathcal{Y}$, where $\mathcal{Y}$ is a finite subset of $\mathbb{R}$, be a monotone function.
- Define $g_j(x) = \min\{g(z): z \in \mathcal{N}_\lambda^{\downarrow}(x)\}$ for each depth $j = 0,1,\ldots,\lambda$ and a sequence $\vec{g}(x) = (g_0(x), g_1(x), \ldots, g_\lambda(x))$



The interleaving property

If g is monotone and datasets $x \subset x'$ are neighbors, then $\vec{g}(x)$ and $\vec{g}(x')$ are interleaved:

$$g_\lambda(x) \le g_\lambda(x') \le \cdots \le g_1(x) \le g_1(x') \le g_0(x) \le g_0(x')$$

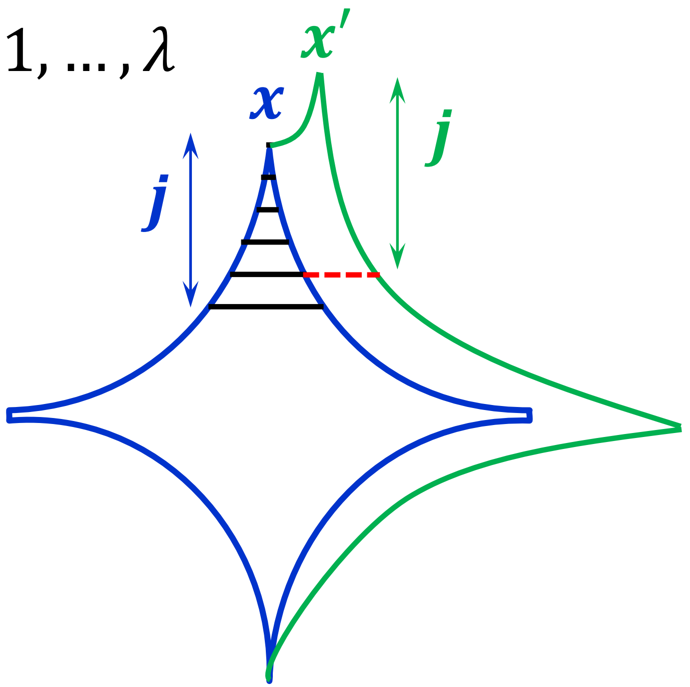
Proof: (1) $g_{j+1}(x') \le g_j(x)$ for all j

$$\mathcal{N}_{j+1}^{\downarrow}(x') \supset \mathcal{N}_j^{\downarrow}(x)$$

We are taking the minimum over the corresponding down−neighborhoods.

ShI mechanism: analysis [Fang Dong Yi 22]

- Let $g: \mathcal{U}^* \to \mathcal{Y}$, where $\mathcal{Y}$ is a finite subset of $\mathbb{R}$, be a monotone function.
- Define $g_j(x) = \min\{g(z): z \in \mathcal{N}_\lambda^{\downarrow}(x)\}$ for each depth $j = 0,1,\dots,\lambda$ and a sequence $\vec{g}(x) = \left(g_0(x), g_1(x), \dots, g_\lambda(x)\right)$



The interleaving property

If g is monotone and datasets $x \subset x'$ are neighbors, then $\vec{g}(x)$ and $\vec{g}(x')$ are interleaved:

$$g_\lambda(x) \le g_\lambda(x') \le \cdots \le g_1(x) \le g_1(x') \le g_0(x) \le g_0(x')$$

Proof: (2) $g_j(x) \le g_j(x')$ for all j

Suppose $x' = x \cup \{k\}$ and let $z' = argmin\,\{g(z): z \in \mathcal{N}_j^{\downarrow}(x')\}$, i.e., $g_j(x') = g(z')$

Then either $z' \in \mathcal{N}_j^{\downarrow}(x)$ $\Rightarrow$ $\boldsymbol{g_j(x) \le g(z') = g_j(x')}$

or $z' = z \cup \{k\}$ $\Rightarrow$ $\boldsymbol{g_j(x) \le g(z) \le g(z') = g_j(x')}$ by monotonicity of g

ShI mechanism: analysis [Fang Dong Yi 22]

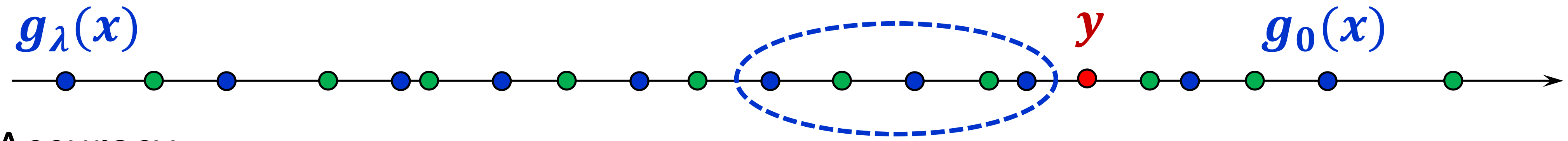
The interleaving property

If g is monotone and datasets $x \subset x'$ are neighbors, then $\vec{g}(x)$ and $\vec{g}(x')$ are interleaved:

$$g_\lambda(x) \le g_\lambda(x') \le \cdots \le g_1(x) \le g_1(x') \le g_0(x) \le g_0\,(x')$$

- Interleaving ⇒ privacy

  $|score_x(y) - score_{x'}(y)| \le 1$

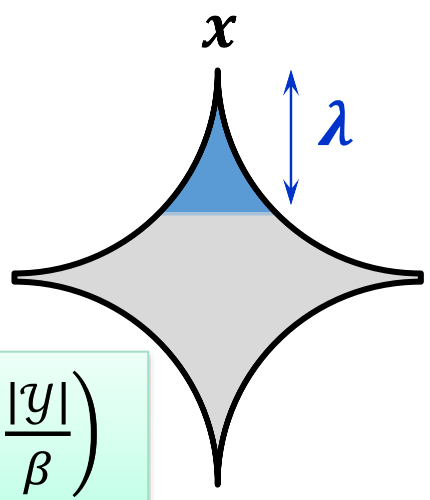


- Accuracy

  With probability at least $1-\beta$, ShI outputs

$$y \text{ with } score_x(y) = O\left(\frac{1}{\epsilon}\log\frac{|\mathcal{Y}|}{\beta}\right)$$

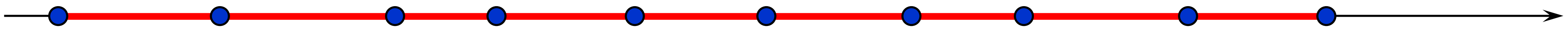
$$y \in [g(x) - DS_\lambda^f(x), g(x)] \text{ for sufficiently large } \lambda \in \Theta\left(\frac{1}{\epsilon}\log\frac{|\mathcal{Y}|}{\beta}\right)$$



Modified ShI [this work]: *abstracting ShI*

Idea: Abstract the original version of ShI
as a reduction to the Generalized Interior Point problem

Given a sequence $\vec{a}$ of numbers, define $hull(\vec{a}) = [\min(\vec{a}), \max(\vec{a})]$



- Interior Point Problem: Given a dataset x, return $y \in hull(x)$
(with the usual definition of DP)
- Generalized Interior Point Problem [Bun Dwork Rothblum Steinke 18, Cohen Lyu Nelson Sarlós Stemmer 23]: Given a dataset x, construct $\vec{a}$ and return $y \in hull(a)$
(DP if for all neighbors x and x',
the corresponding sequences $\vec{a}$ and $\overrightarrow{a'}$ are interleaved)

Modified ShI [this work]

- Generalized Interior Point Problem [Bun Dwork Rothblum Steinke 18, Cohen Lyu Nelson Sarlós Stemmer 23]: Given a dataset x, construct $\vec{a}$ and return $y \in hull(a)$
  (DP if for all neighbors x and x', the corresponding sequences $\vec{a}$ and $\overrightarrow{a'}$ are interleaved)

- Modified ShI: Instead of using the exponential mechanism for the median on $\vec{g}(x)$, we use the state-of-the-art (ϵ, δ)-DP algorithms for Generalized Interior Point.

Accuracy of these algorithms translates into locality λ for Modified ShI

With probability at least $1 - \beta$, it outputs $y \in [\min_{z \in \mathcal{N}_\lambda^{\downarrow}(x)} g(z), g(x)]$

This improves the dependence on $r = |\mathcal{Y}|$ in locality λ from $\log r$ to $\log\frac{1}{\delta} \cdot 2^{O(\log^* r)}$ at the price of having an (ϵ, δ)-DP with positive δ instead of $(\epsilon, 0)$-DP

Modified ShI runs the best of the two algorithms for a given parameter setting.

What's missing for a black-box wrapper?

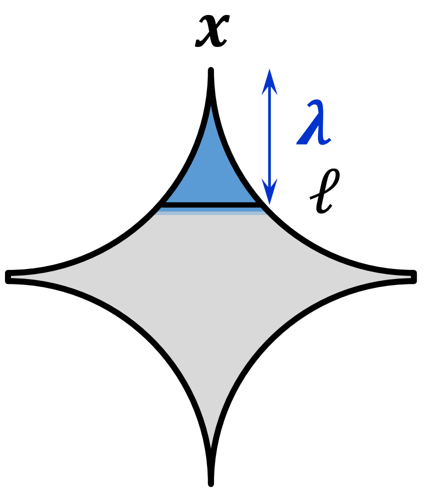
Issue

- Privacy guarantees of ShI [Fang Dong Yi 22] and our Modified ShI [this work] require monotonicity everywhere
- But Curi gets a black box that computes f

Solution

- Locally transform to f get monotonicity

Enforcing monotonicity locally



Monotonization operator M_ℓ

For each $\ell \in \mathbb{Z}$, the level-ℓ monotonization of a function $f : \mathcal{U}^* \to \mathcal{Y}$ is a function $M_\ell[f]$ defined by

$$M_\ell[f](x) = \max(\{f(z) \colon \underbrace{z \subseteq x, |z| \geq \ell}_{\mathcal{N}^{\downarrow}_{|x|-\ell}(x)}\} \cup \{\inf \mathcal{Y}\})$$

Properties of monotonization (for all ℓ and f)

1. Function $M_\ell[f](x)$ is monotone
2. If f is monotone, then $M_\ell[f] = f$.
3. The value $M_\ell[f](x)$ can be computed by querying f on all subsets of x of size at least ℓ.

Idea: Pick ℓ randomly, aiming to get $\ell \approx |x| - \lambda$

Privacy wrapper with automated sensitivity detection (GenShI)

GenShI (Input: dataset x, privacy parameters ϵ, δ, failure probability β, finite range $\mathcal{Y} \subset \mathbb{R}$;
query access to a function $f: \mathcal{U}^* \to \mathcal{Y}$)

1. Set λ to twice the depth needed to run Modified ShI with parameters $\frac{\epsilon}{2}, \delta, \frac{\beta}{2}$ and $\mathcal{Y}$
2. Release $\ell \leftarrow \left\lfloor |x| - \frac{3}{4}\lambda + Z \right\rfloor$ where $Z \sim Laplace\left(\frac{2}{\epsilon}\right)$
3. Run Modified ShI with parameters $\frac{\epsilon}{2}, \delta, \frac{\beta}{2}$ and $\mathcal{Y}$
   and query access to the monotonization $M_\ell[f]$ and return its answer.

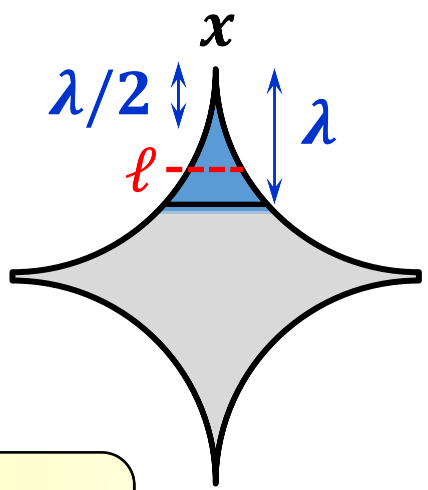
Privacy analysis:

- Step 2 runs the Laplace mechanism with parameter $\frac{\epsilon}{2}$ to release a function with GS=1
- Step 3 runs an $\left(\frac{\epsilon}{2}, \delta\right)$-DP mechanism
- By composition, GenShI is (ϵ, δ)-DP

Privacy wrapper with automated sensitivity detection (GenShI)

$\mathcal{W}$: GenShI (Input: dataset x, privacy parameters ϵ, δ, failure probability β, finite range $\mathcal{Y} \subset \mathbb{R}$;
query access to a function $f: \mathcal{U}^* \to \mathcal{Y}$)

1. Set λ to twice the depth needed to run Modified ShI with parameters $\frac{\epsilon}{2}, \delta, \frac{\beta}{2}$ and $\mathcal{Y}$
2. Release $\ell \leftarrow \left\lfloor |x| - \frac{3}{4}\lambda + Z \right\rfloor$ where $Z \sim Laplace\left(\frac{2}{\epsilon}\right)$
3. Run Modified ShI with parameters $\frac{\epsilon}{2}, \delta, \frac{\beta}{2}$ and $\mathcal{Y}$
and query access to the monotonization $M_\ell[f]$ and return its answer.



Accuracy claim for GenShI privacy wrapper

With probability at least $1 - \beta$, GenShI outputs $y \in hull\{f(z):\ z \in \mathcal{N}_\lambda^\downarrow(x)\}$

Proof: Bad events: (1) noise magnitude $|Z|$ is large; (2) Modified ShI fails

- Condition on bad events not occurring. Then $|x| - \lambda \leq \ell \leq |x| - \lambda/2$ and

$$\min_{z \in \mathcal{N}_{\lambda/2}^\downarrow(x)} M_\ell[f](z) \leq \mathcal{W}^f(x) = \mathrm{ShI}^{M_\ell[f]}(x) \leq M_\ell[f](x)$$

Upper bound: $\mathcal{W}^f(x) \leq M_\ell[f](x)$

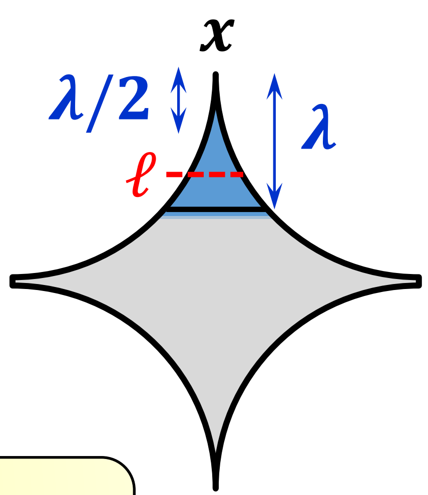
$= \max\{f(z):\ z \subseteq x, |z| \geq \ell\}$ by definition of monotonization

$\leq \max\{f(z):\ z \in \mathcal{N}_\lambda^\downarrow(x)\}$ since $\ell \geq |x| - \lambda$

Privacy wrapper with automated sensitivity detection (GenShI)

$\mathcal{W}$: GenShI (Input: dataset x, privacy parameters ϵ, δ, failure probability β, finite range $\mathcal{Y} \subset \mathbb{R}$; query access to a function $f: \mathcal{U}^* \to \mathcal{Y}$)

1. Set λ to twice the depth needed to run Modified ShI with parameters $\frac{\epsilon}{2}, \delta, \frac{\beta}{2}$ and $\mathcal{Y}$
2. Release $\ell \leftarrow \left\lfloor |x| - \frac{3}{4}\lambda + Z \right\rfloor$ where $Z \sim Laplace\left(\frac{2}{\epsilon}\right)$
3. Run Modified ShI with parameters $\frac{\epsilon}{2}, \delta, \frac{\beta}{2}$ and $\mathcal{Y}$ and query access to the monotonization $M_\ell[f]$ and return its answer.



Accuracy claim for GenShI privacy wrapper

With probability at least $1-\beta$, GenShI outputs $y \in hull\{f(z) \colon z \in \mathcal{N}_\lambda^{\downarrow}(x)\}$

Proof (continued): Condition on bad events not occurring. Then

$$\min_{z \in \mathcal{N}_{\lambda/2}^{\downarrow}(x)} M_\ell[f](z) \le \mathcal{W}^f(x) = \mathrm{ShI}^{M_\ell[f]}(x) \le M_\ell[f](x)$$

Lower bound: $\mathcal{W}^f(x) \ge \min_{z \in \mathcal{N}_{\lambda/2}^{\downarrow}(x)} M_\ell[f](z) \ge \min_{z' \in \mathcal{N}_{\lambda}^{\downarrow}(x)} f(z')$

Monotonization $M_\ell[f](z) = f(z')$ for some $z' \subset z, |z'| \ge \ell$
Since $\ell \ge |x| - \lambda$, this $z' \in \mathcal{N}_\lambda^{\downarrow}(x)$

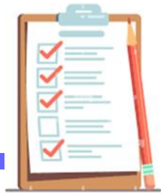
Our results for the automated sensitivity detection setting

The starting point for our algorithm is the Shifted Inverse (ShI) mechanism [Fang Dong Yi 22]

- It is **not** a privacy wrapper, because it is private only for monotone functions.
- It works for real-valued functions with a **finite** range $\mathcal{Y} \subset \mathbb{R}$.

Algorithm	**Privacy**		**Accuracy α**	**Down locality λ**
[Fang Dong Yi 22]	$(\epsilon, 0)$-DP	only for monotone functions	down sensitivity at depth λ, $\mathrm{DS}_\lambda^f(x)$	$\lambda_{(\epsilon,0)} := O\left(\frac{1}{\epsilon}\log\frac{\lvert\mathcal{Y}\rvert}{\beta}\right)$
Modified ShI	(ϵ, δ)-DP			$\lambda_{(\epsilon,\delta)} := \frac{1}{\epsilon}\log\frac{1}{\delta} \cdot 2^{O(\log^* \lvert\mathcal{Y}\rvert)}$
Generalized ShI	(ϵ, δ)-DP	all functions	$\mathrm{DS}_\lambda^f(x)$	$\min(\lambda_{(\epsilon,0)}, \lambda_{(\epsilon,\delta)})$
Lower bound	(ϵ, δ)-DP	all functions	$\mathrm{DS}_\lambda^f(x)$ $\Rightarrow$	$\Omega\left(\frac{1}{\epsilon}\log\min\left(\frac{\lvert\mathcal{Y}\rvert}{\beta}, \frac{1}{\delta}\right)\right)$

Plan



✓ Background on differential privacy and definition of privacy wrappers

✓ Quantitative statement of results

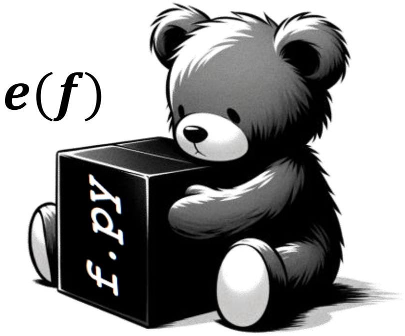
✓ Privacy wrapper for the automated sensitivity detection setting

1. ShI mechanism [Fang Dong Yi 22] for monotone functions

2. Modified ShI (with better dependence on r, the size of the range)

3. From monotone to general functions

$\boldsymbol{range(f)}$



- Extension to graphs and other types datasets

General domains

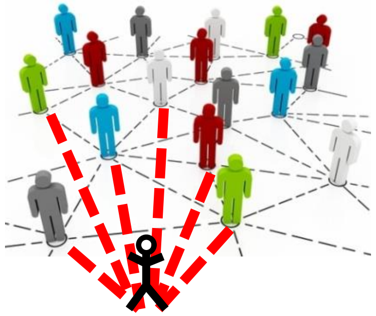
Our privacy wrappers can be implemented
for any partially ordered domain of datasets $(D, \preccurlyeq)$ that satisfies:

- There exists a unique minimum element in D denoted $\emptyset$.
- There is a function $size: D \to \mathbb{Z}_{\geq 0}$ such that,
  for all $u \in \mathrm{D}$, the partial order on the down neighborhood of u
  is isomorphic to a hypercube $\{0,1\}^{size(u)}$.
- There exists a neighbor relation $\sim$ such that
  if $u, v \in D$; $v \preccurlyeq u$; and $size(v) = size(u) - 1$ then $u \sim v$.

Example: Datasets can be graphs (or hypergraphs) with ``node-neighbor'' relationship

G:

G':



Summary of our contributions

- Formulated the automated sensitivity detection setting in the context of black-box privacy.

- Formalized notions of accuracy in both automated sensitivity detection and claimed sensitivity bound settings, appropriate for dealing with large/infinite universe.

- Reinterpreted and analyzed existing constructions, fitting them in the black-box privacy setting.

- Gave nearly optimal privacy wrappers and lower bounds for both settings for black-box functions with real range.

Open questions

- Can the dependence on the size of the range be avoided in the automated sensitivity detection setting?

- Can we design privacy wrappers for functions with more complicated outputs (e.g., vector outputs)?

- Our accuracy guarantees are instance-based. Potentially one can consider different notions of sensitivity/accuracy. Which notions are the best?

- Query complexity $|x|^{\lambda} \approx n^{\mathrm{O}(\log n/\varepsilon)}$ is too large for practice. Are there practical alternatives (e.g., for important function classes, or in combination with formal methods)?