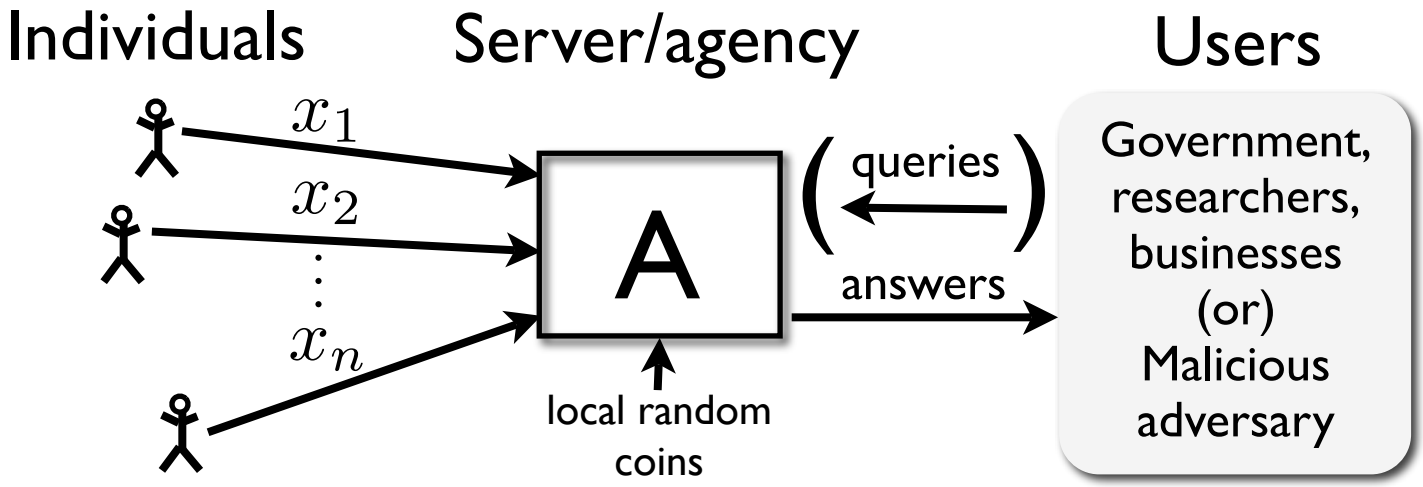
Pinning Down "Privacy" in Statistical Databases

Adam Smith
Computer Science & Engineering Department
Penn State

Privacy in Statistical Databases



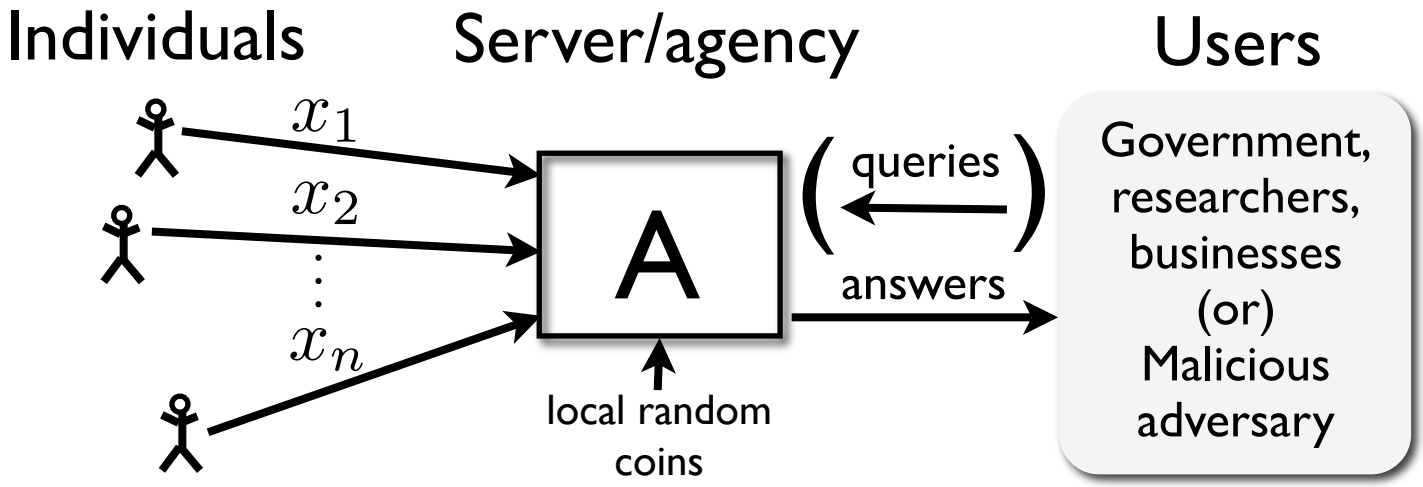
Large collections of personal information

- census data
- medical/public health data
- social networks
- recommendation systems
- trace data: search records, etc
- intrusion-detection systems

Recently:

- larger data sets
- more types of data

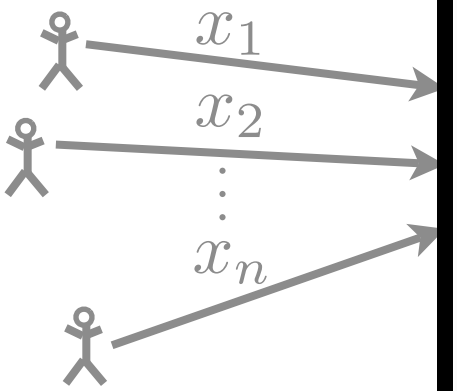
Privacy in Statistical Databases

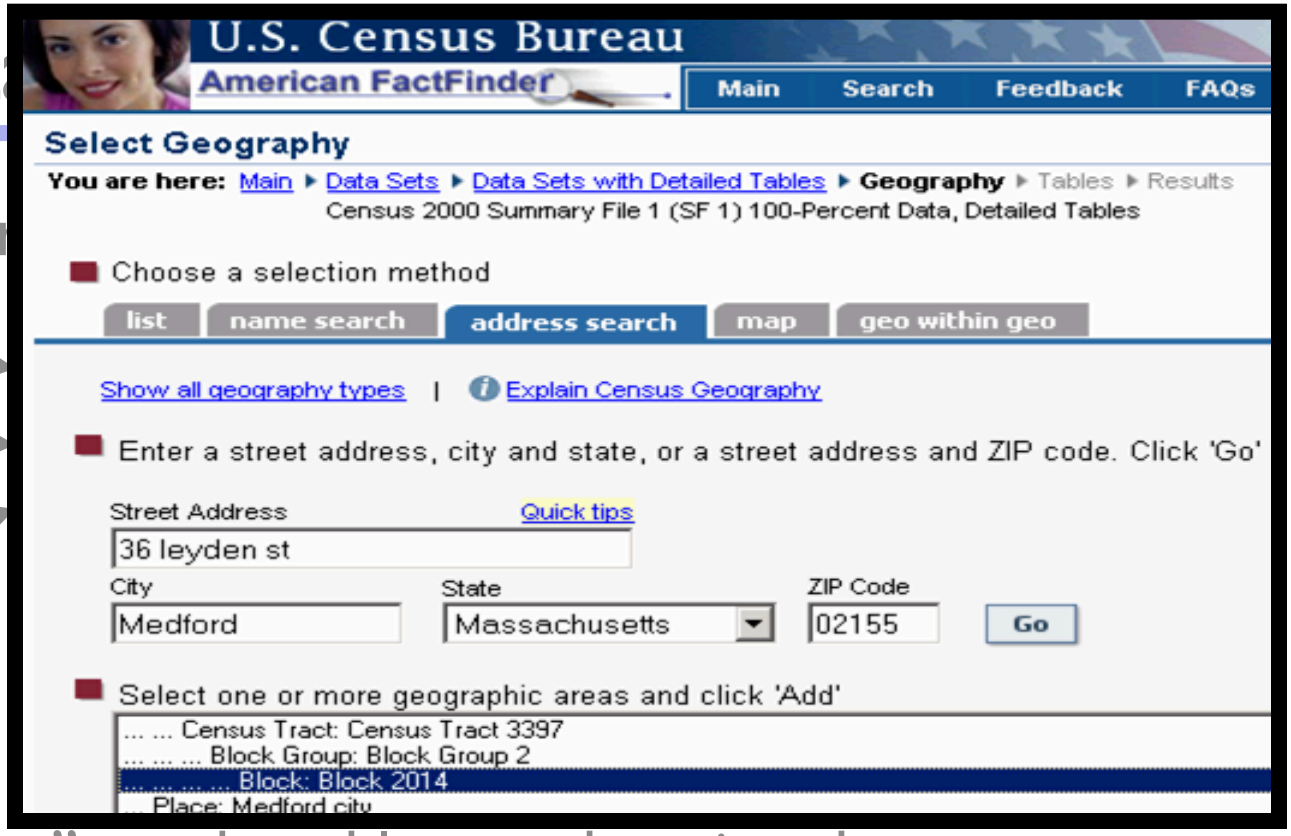


- Published "statistics" may be tables, graphs, microdata, decision trees, neural networks, confidence intervals...
- Data may be numbers, categories, tax forms, web searches...
- May be interactive

Privacy in St

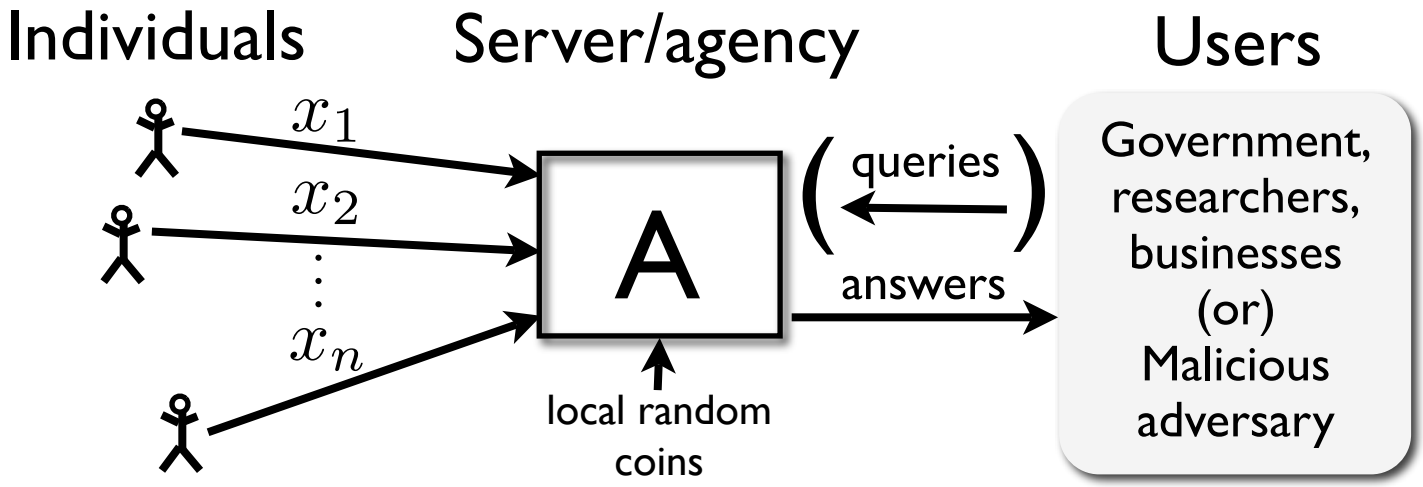
Individuals Ser





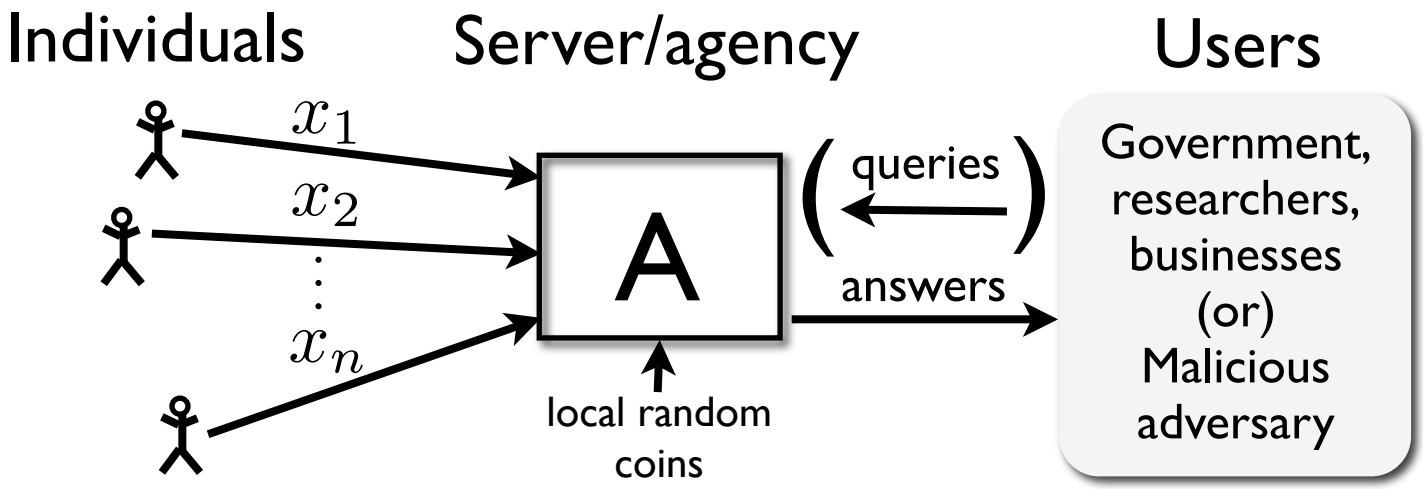
- Published "statistics" may be tables, graphs, microdata, decision trees, neural networks, confidence intervals...
- Data may be numbers, categories, tax forms, web searches...
- May be interactive

Privacy in Statistical Databases



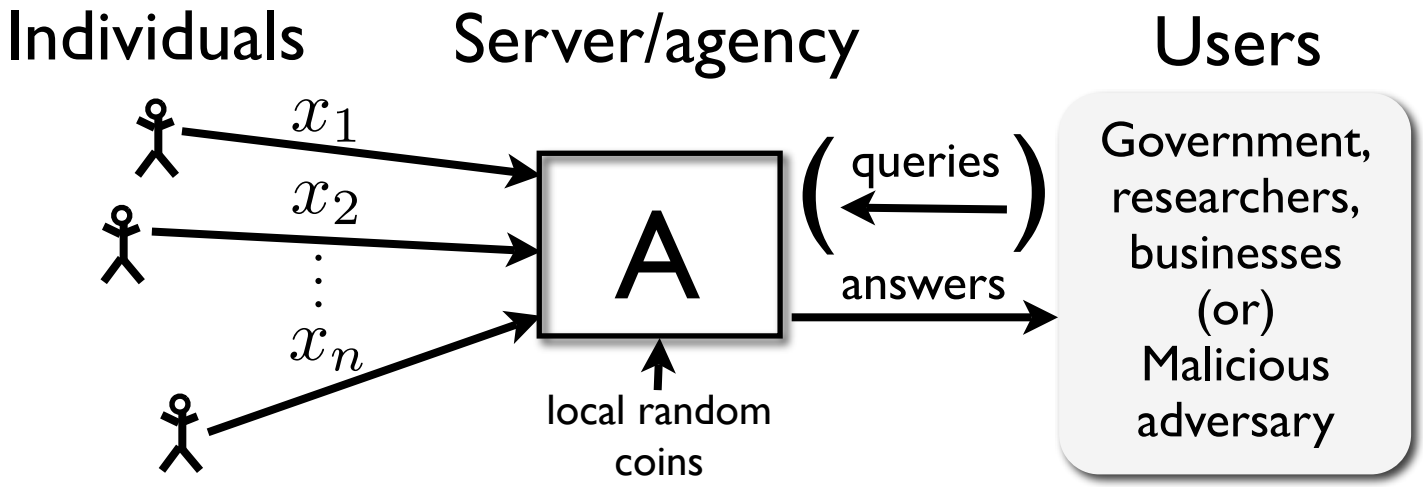
- What information can be released?
- Two conflicting goals
  - Utility: Users can extract "global" properties
  - Privacy ("confidentiality"): Individual information stays hidden
- How can these be formalized?

Privacy in Statistical Databases



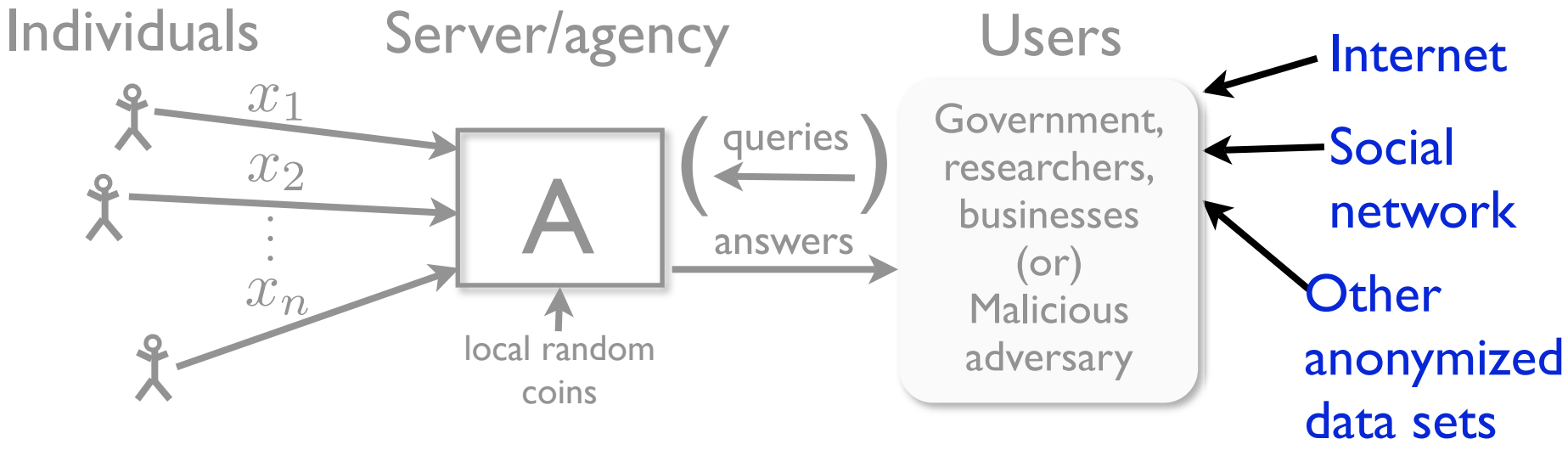
- Variations on model studied in
  - Statistics ("statistical disclosure control")
  - Data mining ("privacy-preserving data mining" *)
- No coherent theory
- Recently: crypto & theoretical CS
  - Focused on rigorous approach to privacy

How can we formalize "privacy"?

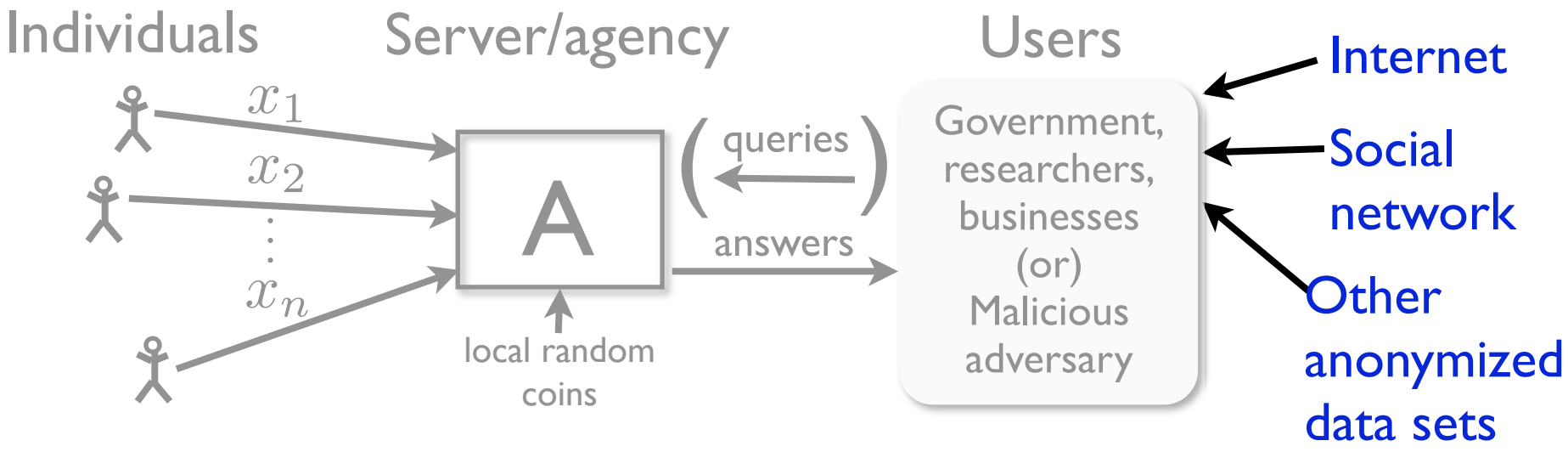


- "Privacy" is harder to reason about than "utility"
  - Utility is what we're used to
- Existing definitions problematic
  - Many are not specified precisely
  - Fail in the presence of external information

External Information

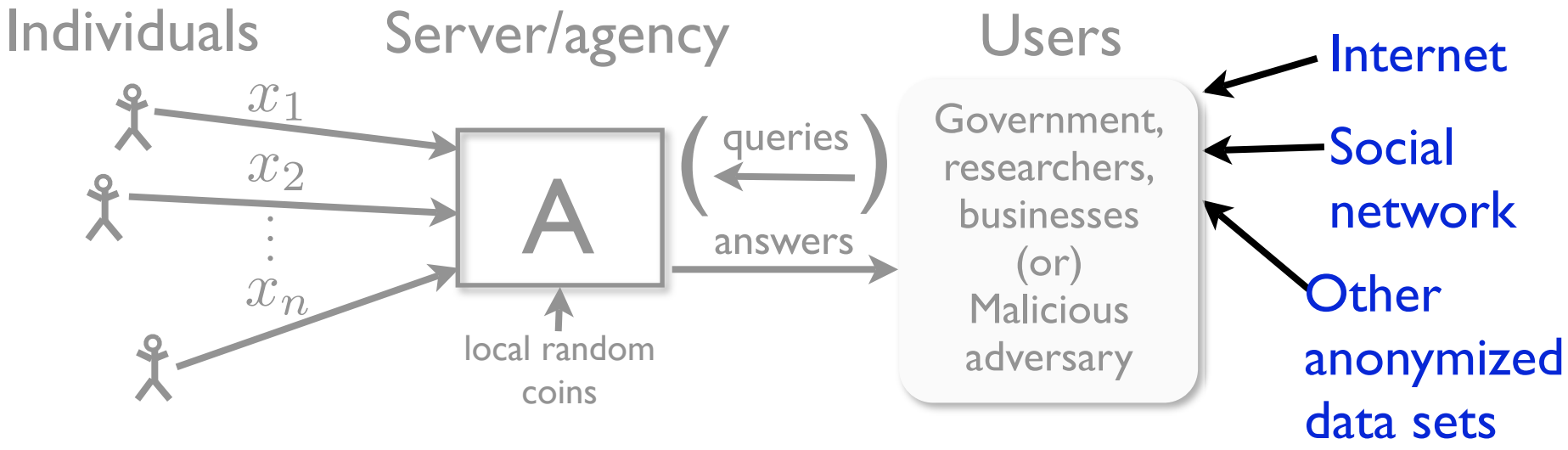


External Information



- Users have external information sources
  - Can't assume we know the sources
- Anonymization schemes regularly broken

External Information



- Users have external information sources
  - Can't assume we know the sources
- Anonymization schemes regularly broken
- Example: two hospitals independently release statistics about overlapping populations
  - Combining information "breaks" several current techniques [Ganta, S.]

How can we formalize “privacy”?

How can we formalize "privacy"?

- **Goal #1: Rigor**
  - Raise the bar for how we think about privacy
    - Especially external information
  - Make clear and refutable statements/conjectures

How can we formalize "privacy"?

- **Goal #1: Rigor**
  - Raise the bar for how we think about privacy
    - Especially external information
  - Make clear and refutable statements/conjectures
- **Goal #2: Interesting science**
  - (New) Computational phenomenon
  - Unify different approaches
  - Algorithmic, statistical, cryptographic challenges

This talk

- "Differential" privacy
  - Handles arbitrary external information
  - What can we compute privately?
- Example technique: Output perturbation
  - Calibrating noise to "sensitivity"
  - Sample-aggregate methodology

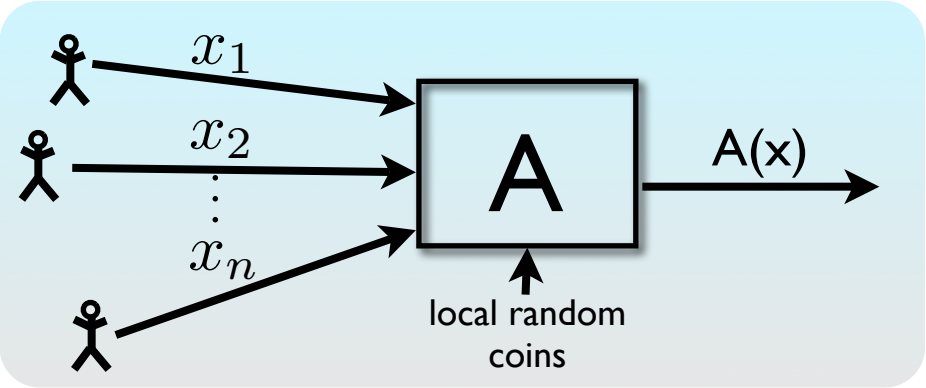
This talk

- "Differential" privacy
  - Handles arbitrary external information
  - What can we compute privately?
- Example technique: Output perturbation
  - Calibrating noise to "sensitivity"
  - Sample-aggregate methodology

Defining Privacy [DiNi,DwNi,BDMN,DMNS]

- Intuition:
  - Changes to my data not noticeable by users
  - Output is "independent" of my data

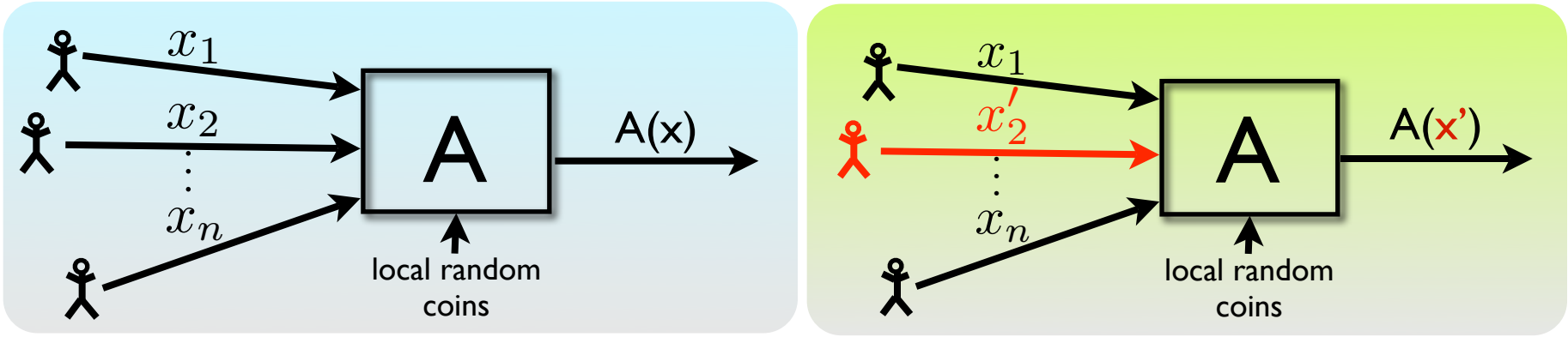
Defining Privacy [DiNi,DwNi,BDMN,DMNS]



- Data set $\mathbf{x} = (x_1, \ldots, x_n) \in D^n$
  - Domain D can be numbers, categories, tax forms
  - Think of x as **fixed** (not random)
- A = **randomized** procedure run by the agency
  - A(x) is a random variable distributed over possible outputs

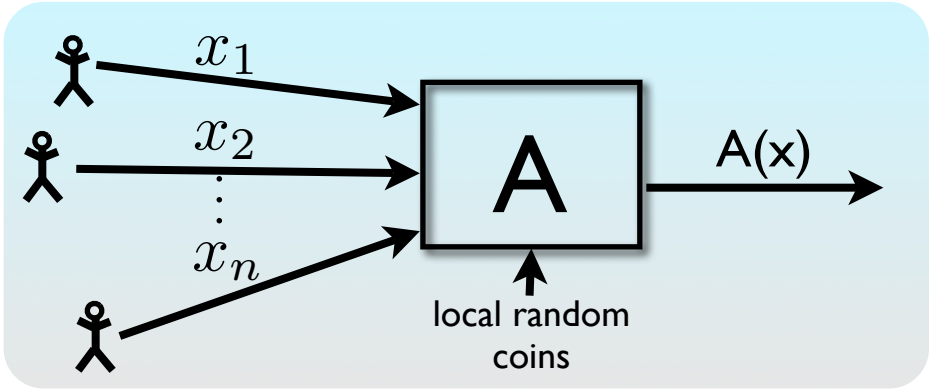
    Randomness might come from adding noise, resampling, etc.

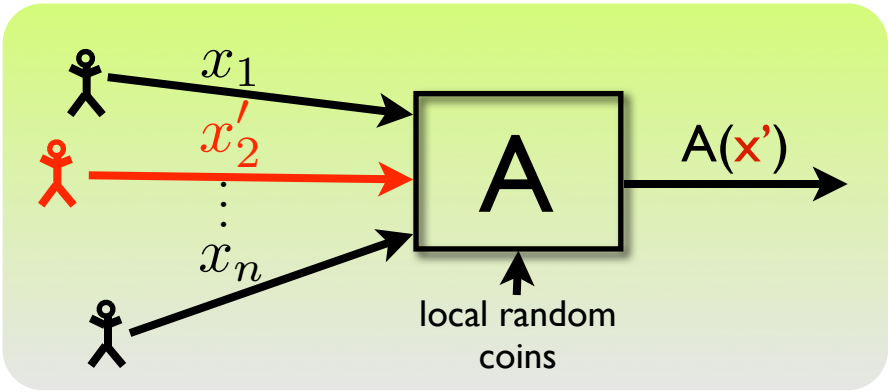
Defining Privacy [DiNi,DwNi,BDMN,DMNS]



x' is a neighbor of x
if they differ in one data point

Defining Privacy [DiNi,DwNi,BDMN,DMNS]

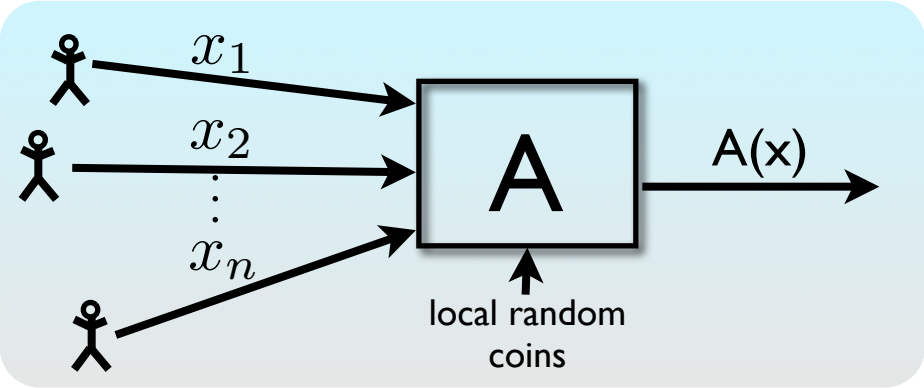


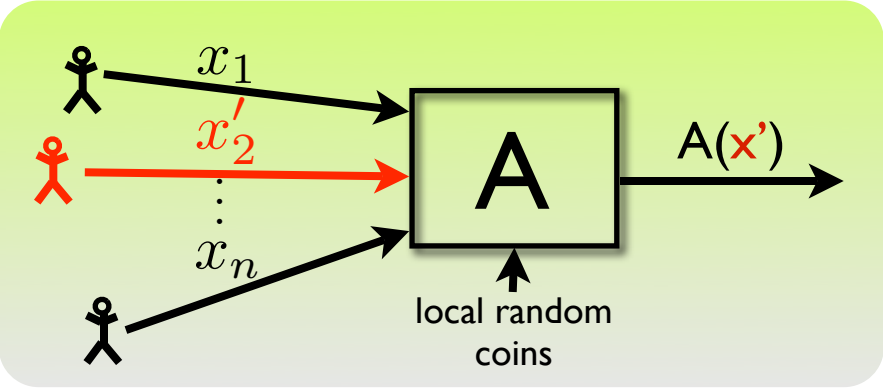


x' is a neighbor of x
if they differ in one data point

Neighboring databases induce **close** distributions on outputs

Defining Privacy [DiNi,DwNi,BDMN,DMNS]





x' is a neighbor of x
if they differ in one data point

Neighboring databases induce **close** distributions on outputs

Definition: A is ϵ-differentially private if,
for all neighbors x, x',
for all subsets S of outputs

$$\Pr(A(x) \in S) \leq e^{\epsilon} \cdot \Pr(A(x') \in S)$$

Defining Privacy [DiNi,DwNi,BDMN,DMNS]

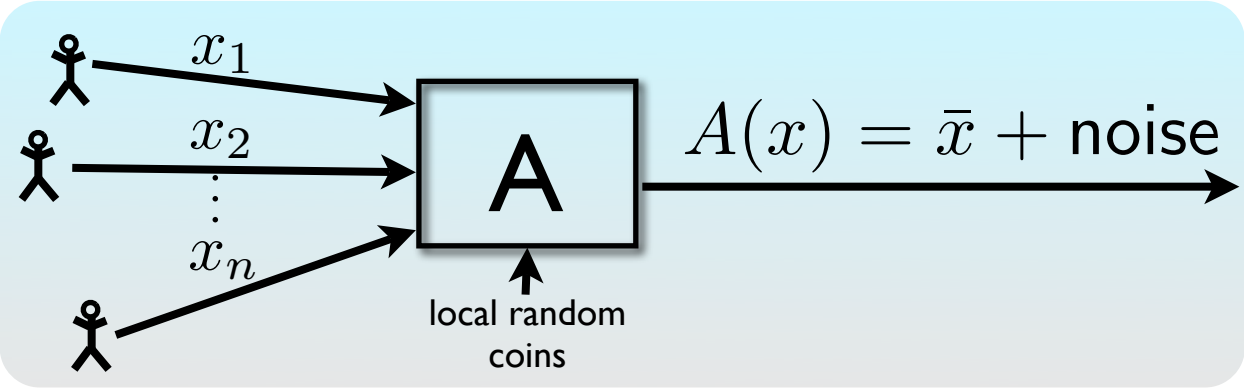
- ϵ cannot be too small (think $\frac{1}{10}$, not $\frac{1}{2^{50}}$)
- Distance measure on distributions matters
- This is a condition on the **algorithm** (process) A
  - Saying "this output is safe" doesn't take into account how it was computed
  - Common problem in the literature...

Neighboring databases induce **close** distributions on outputs

Definition: A is ϵ-differentially private if,
for all neighbors x, x',
for all subsets S of outputs

$$\Pr(A(x) \in S) \leq e^{\epsilon} \cdot \Pr(A(x') \in S)$$

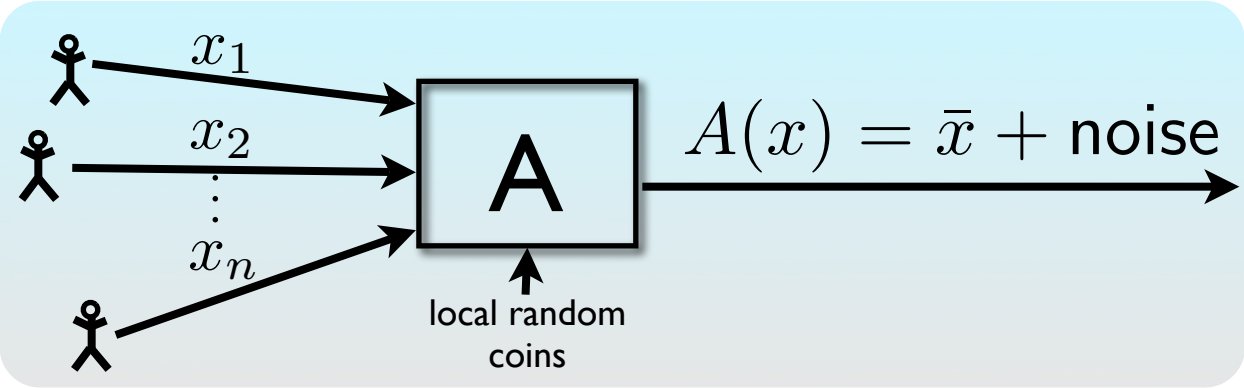
Example: Perturbing the Average



$$x_i \in \{0, 1\}$$

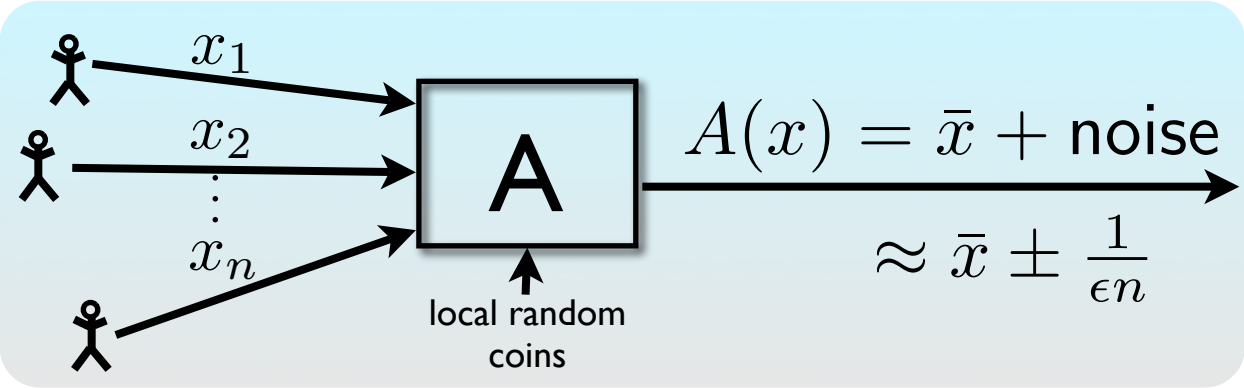
$$\bar{x} = \frac{1}{n} \sum_i x_i$$

Example: Perturbing the Average



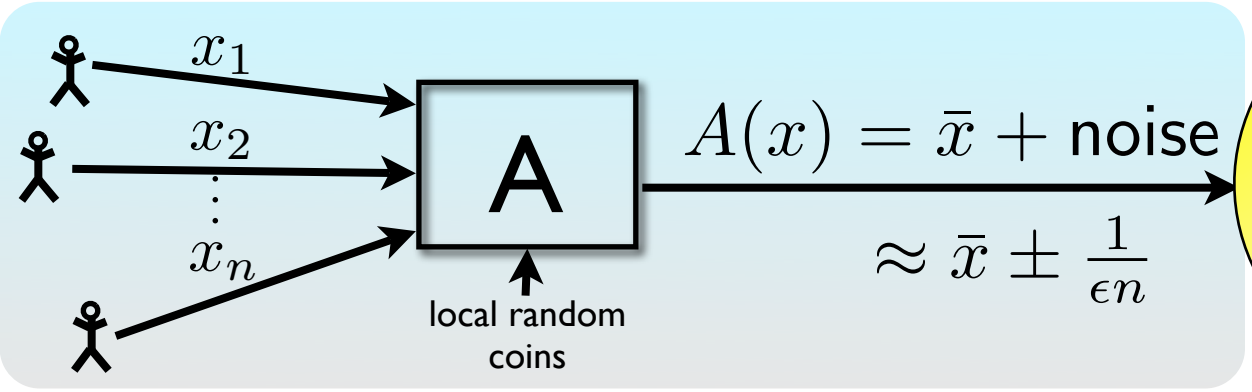
- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release average $\bar{x} = \frac{1}{n} \sum_i x_i$

Example: Perturbing the Average



- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release average $\bar{x} = \frac{1}{n} \sum_i x_i$
- **Claim**: If noise $\sim$ Lap$(\frac{1}{\epsilon n})$ then A is ϵ-differentially private

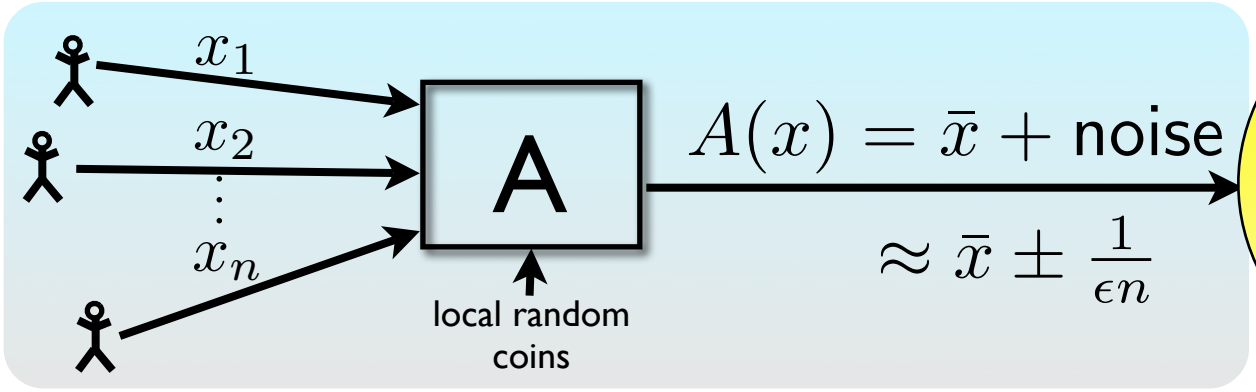
Example: Perturbing the Average



If x is a random sample from an underlying population, then get sampling noise $\approx \frac{1}{\sqrt{n}}$.

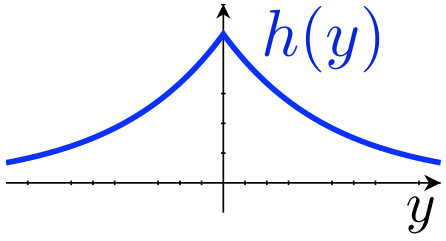
- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release average $\bar{x} = \frac{1}{n} \sum_i x_i$
- **Claim**: If noise $\sim$ Lap$(\frac{1}{\epsilon n})$ then A is ϵ-differentially private

Example: Perturbing the Average

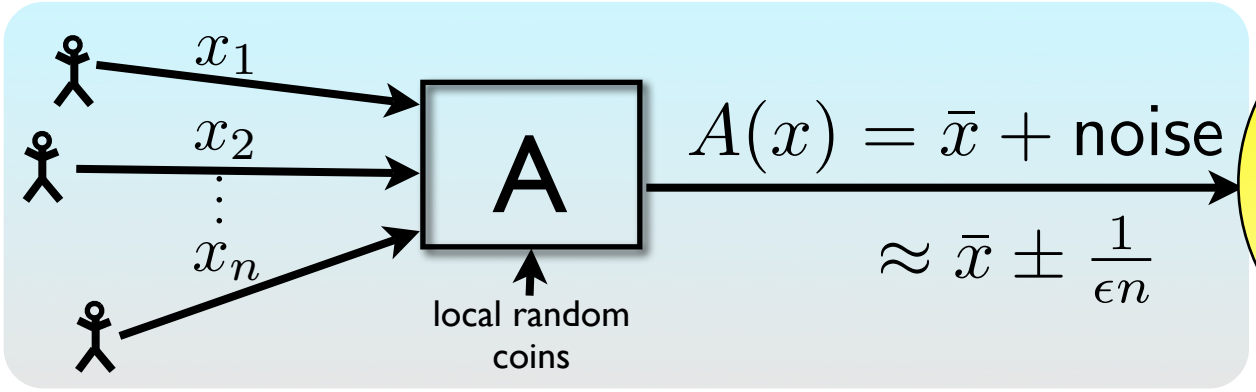


If x is a random sample from an underlying population, then get sampling noise $\approx \frac{1}{\sqrt{n}}$.

- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release average $\bar{x} = \frac{1}{n} \sum_i x_i$
- **Claim**: If noise $\sim$ Lap$(\frac{1}{\epsilon n})$ then A is ϵ-differentially private
  - Laplace distribution Lap(λ) has density $h(y) \propto e^{-|y|/\lambda}$
  - Sliding property: $\frac{h(y)}{h(y+\delta)} \leq e^{\delta/\lambda}$

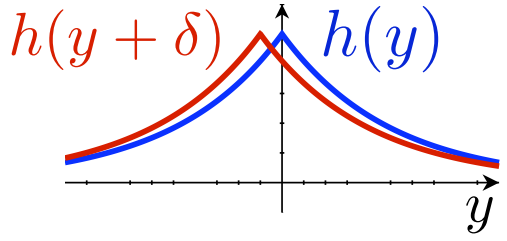


Example: Perturbing the Average

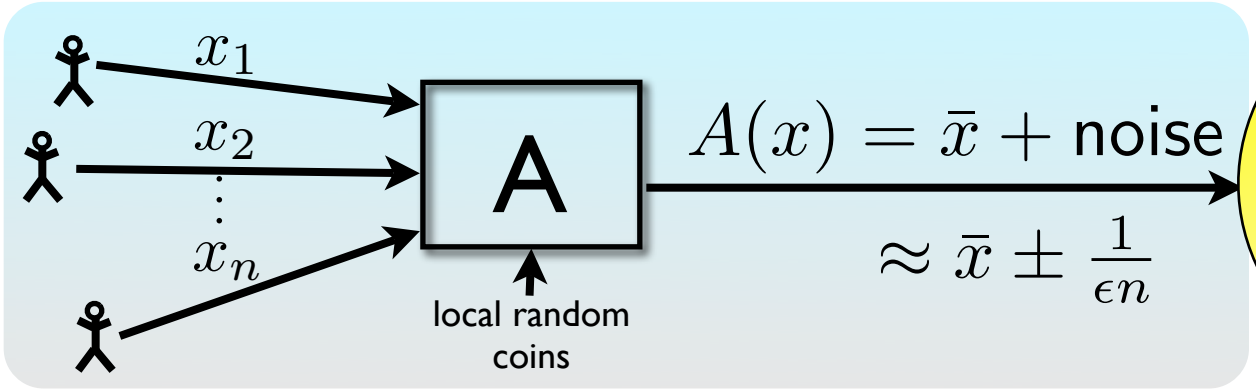


If x is a random sample from an underlying population, then get sampling noise $\approx \frac{1}{\sqrt{n}}$.

- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release average $\bar{x} = \frac{1}{n} \sum_i x_i$
- **Claim**: If noise $\sim$ Lap$(\frac{1}{\epsilon n})$ then A is ϵ-differentially private
  - Laplace distribution Lap(λ) has density $h(y) \propto e^{-|y|/\lambda}$
  - Sliding property: $\frac{h(y)}{h(y+\delta)} \leq e^{\delta/\lambda}$



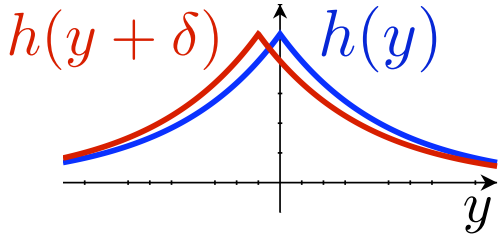
Example: Perturbing the Average



If x is a random sample from an underlying population, then get sampling noise $\approx \frac{1}{\sqrt{n}}$.

- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release average $\bar{x} = \frac{1}{n}\sum_i x_i$
- **Claim**: If noise $\sim$ Lap$(\frac{1}{\epsilon n})$ then A is ϵ-differentially private

  - Laplace distribution Lap(λ) has density $h(y) \propto e^{-|y|/\lambda}$
  - Sliding property: $\frac{h(y)}{h(y+\delta)} \leq e^{\delta/\lambda}$
  - A(x) = blue curve, A(x') = red curve
  - $\delta = |\bar{x} - \bar{x}'| \leq \frac{1}{n} \implies \frac{\text{blue curve}}{\text{red curve}} \leq e^{\epsilon}$



Why is this a good definition?

Definition: A is ϵ-differentially private if,
for all neighbors x, x',
for all subsets S of transcripts

$$\Pr(A(x) \in S) \leq e^{\epsilon} \cdot \Pr(A(x') \in S)$$

Neighboring databases induce **close** distributions on transcripts

Why is this a good definition?

- "Composition": **If** algorithms A_1 and A_2 are ϵ-differentially private **then** the outputting results of both algorithms $A_1(x), A_2(x)$ is 2ϵ-differentially private
- "Group privacy": $k\epsilon$-differential privacy for groups of size $\leq k$
- Meaningful in the presence of arbitrary external information

Definition: A is ϵ-differentially private if,
for all neighbors x, x',
for all subsets S of transcripts

$$\Pr(A(x) \in S) \leq e^{\epsilon} \cdot \Pr(A(x') \in S)$$

Neighboring databases induce **close** distributions on transcripts

Why is this a good definition?

Why is this a good definition?

- A naïve hope:

> Your beliefs about me are the same
> after you see the output as they were before

- Suppose you know I am the height of average Canadian
  - You could learn my height from database!
    But it didn't matter whether or not my data was part of it.
  - Has my privacy been compromised? No!
  - **Theorem** (Dwork-Naor): Learning things about individuals is unavoidable in the presence of external information

Why is this a good definition?

- A naïve hope:

  ~~Your beliefs about me are the same after you see the output as they were before~~

- Suppose you know I am the height of average Canadian
  - You could learn my height from database!
    But it didn't matter whether or not my data was part of it.
  - Has my privacy been compromised? No!
  - **Theorem** (Dwork-Naor): Learning things about individuals is unavoidable in the presence of external information

Why is this a good definition?

- A naïve hope:

~~Your beliefs about me are the same after you see the output as they were before~~

- Suppose you know I am the height of average Canadian
  - ➢ You could learn my height from database!
    But it didn't matter whether or not my data was part of it.
  - ➢ Has my privacy been compromised? No!
  - ➢ **Theorem** (Dwork-Naor): Learning things about individuals is unavoidable in the presence of external information
- [DM] Differential privacy implies:
  No matter what you know ahead of time,

You learn the same things about me
whether or not I am in the database

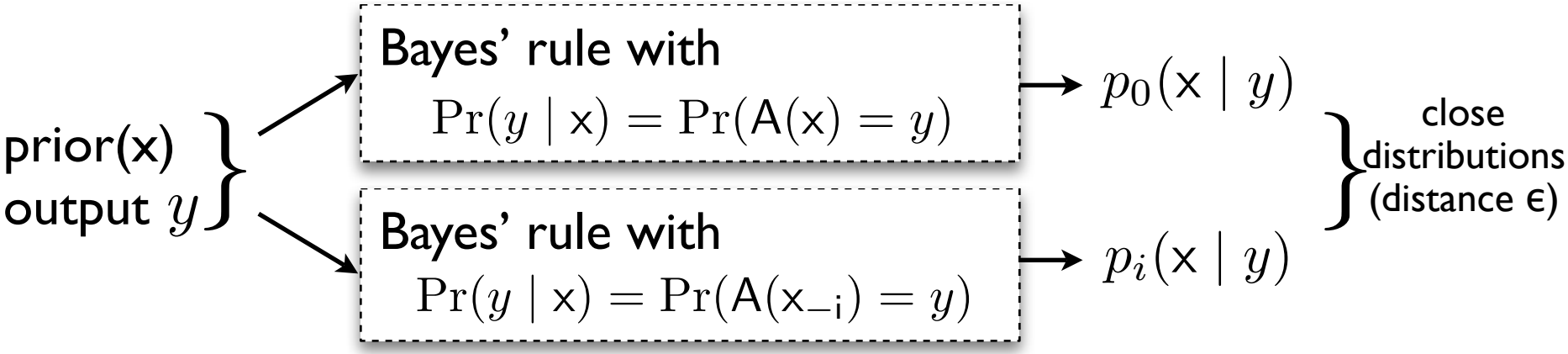
Why is this a good definition?

Why is this a good definition?

- Consider an intruder trying to infer personal information
  - "Background knowledge" = prior distribution on data x
  - "Conclusions you draw" = posterior $p(\cdot|output)$
  - Experiment 0: Run A(x)
  - Experiment *i*: Run $A(x_{-i})$ where $x_{-i} = (x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$

Why is this a good definition?

- Consider an intruder trying to infer personal information
  - "Background knowledge" = prior distribution on data x
  - "Conclusions you draw" = posterior $p(\cdot|output)$
  - Experiment 0: Run A(x)
  - Experiment *i*: Run $A(x_{-i})$ where $x_{-i} = (x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$
- **Lemma**: $\forall$ prior, $\forall$ output, $p_0(\cdot|output) \approx p_i(\cdot|output)$



Why is this a good definition?

- Consider an intruder trying to infer personal information
  - "Background knowledge" = prior distribution on data x
  - "Conclusions you draw" = posterior p(·|*output*)
  - Experiment 0: Run A(x)
  - Experiment *i*: Run $A(x_{-i})$ where $x_{-i} = (x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$
- **Lemma**: ∀ prior, ∀ output, $p_0(\cdot|output) \approx p_i(\cdot|output)$
- Proof:

$$p_0(x) = \frac{\Pr(A(x) = output) * prior(x)}{\int_t \Pr(A(t) = output) * prior(t)}$$

$$\approx \frac{\Pr(A(x_{-i}) = output) * prior(x)}{\int_t \Pr(A(t_{-i}) = output) * prior(t)} = p_1(x)$$

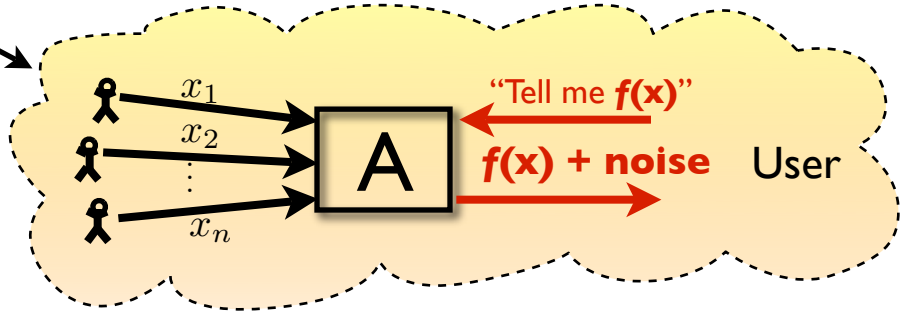
- Similar lemmas hold for relaxations of definition

What can we compute privately?

- "Privacy" = change in one input leads to small change in output distribution

  What computational tasks can we achieve privately?

- Research so far
  - Function approximation [DN, DN,BDMN,DMNS,NRS,BCDKMT,BLR]
  - Mechanism Design [MT]
  - Learning [BDMN,KLNRS]
  - Statistical estimation [S]
  - Synthetic Data [MKAGV]
  - Distributed protocols [DKMMN,BNO]
  - Impossibility results / lower bounds [DiNi,DMNS,DMT]



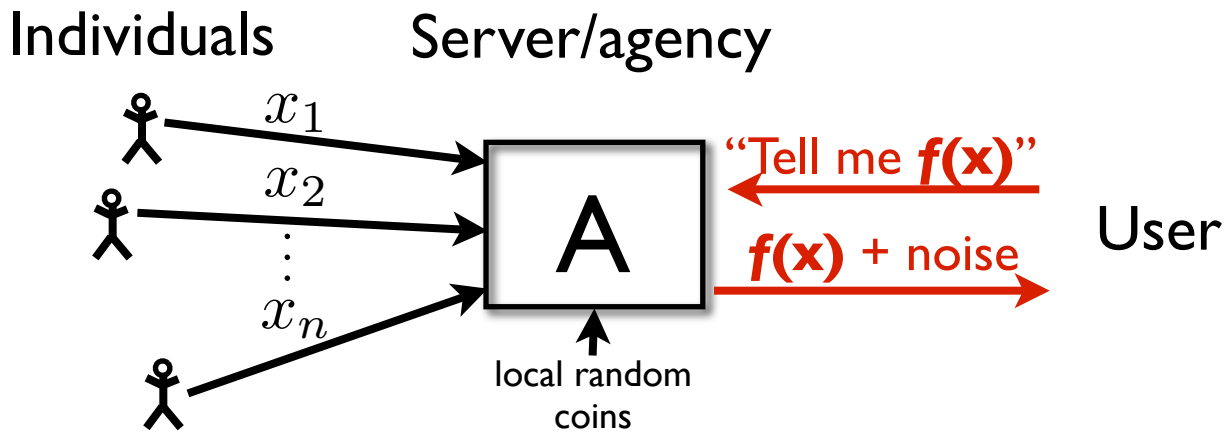
This talk

- "Differential" privacy
  - Handles arbitrary external information
  - What can we compute privately?
- Example technique: Output perturbation
  - Calibrating noise to "sensitivity"
  - Sample-aggregate methodology

This talk

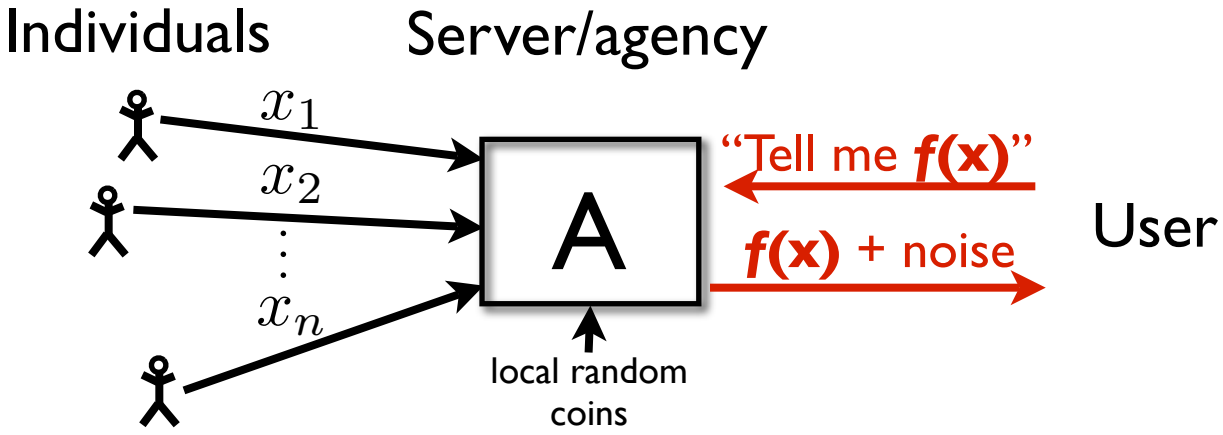
- "Differential" privacy
  - Handles arbitrary external information
  - What can we compute privately?
- Example technique: Output perturbation
  - Calibrating noise to "sensitivity"
  - Sample-aggregate methodology

Output Perturbation, more generally



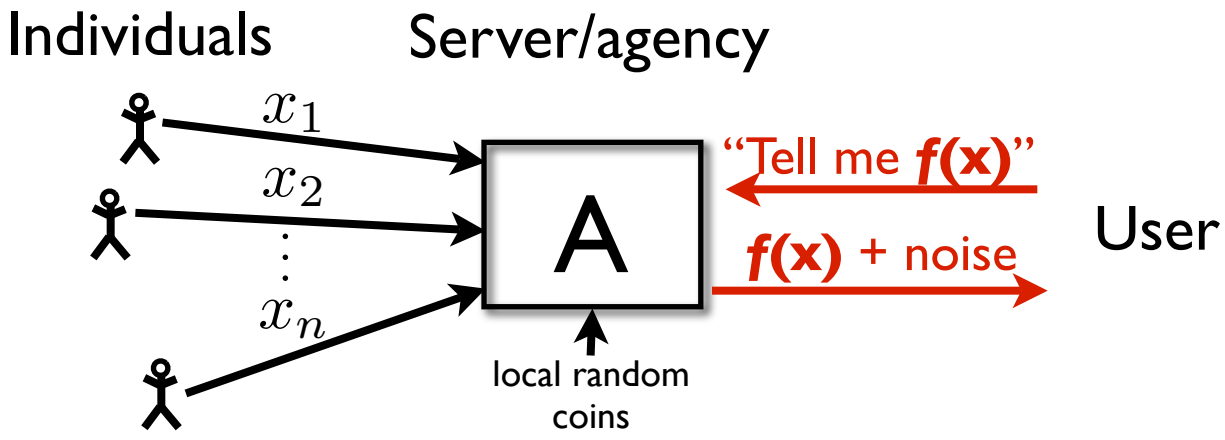
- May be interactive
  - Non-interactive: release pre-defined summary stats + noise
  - Interactive: respond to user requests
- May be repeated many times
  - Composition: q releases are jointly qϵ-differentially private
- How much noise is enough? (How much is too much?)

Global Sensitivity [DMNS06]



- Intuition: ***f*(x)** can be released accurately when ***f*** is insensitive to individual entries $x_1, x_2, \ldots, x_n$
- Global Sensitivity: $\mathrm{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
- Example: $\mathrm{GS}_{\text{average}} = \frac{1}{n}$

Global Sensitivity [DMNS06]



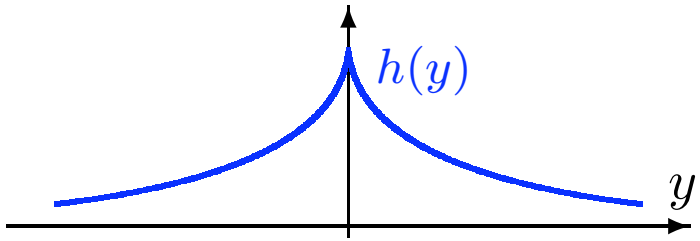
- Intuition: ***f*(x)** can be released accurately when ***f*** is insensitive to individual entries $x_1, x_2, \ldots, x_n$
- Global Sensitivity: $\mathrm{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
- Example: $\mathrm{GS}_{\text{average}} = \frac{1}{n}$

Theorem: If $\mathrm{A}(x) = f(x) + \mathrm{Lap}\left(\frac{\mathrm{GS}_f}{\epsilon}\right)$, then A is ϵ-differentially private.

Global Sensitivity [DMNS06]

Theorem: If $\mathsf{A}(\mathsf{x}) = f(\mathsf{x}) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\epsilon}\right)$, then A is ϵ-differentially private.

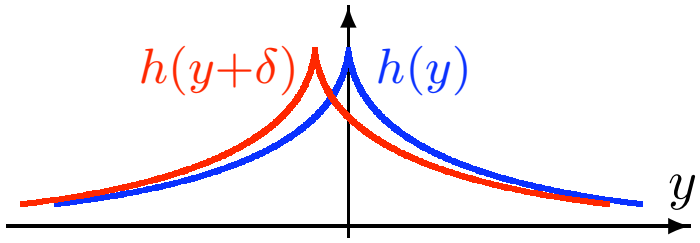
Laplace distribution $\mathsf{Lap}(\lambda)$ has density $h(y) \propto e^{-\frac{\|y\|_1}{\lambda}}$



Global Sensitivity [DMNS06]

Theorem: If $\mathsf{A}(\mathsf{x}) = f(\mathsf{x}) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\epsilon}\right)$, then A is ϵ-differentially private.

Laplace distribution $\mathsf{Lap}(\lambda)$ has density $h(y) \propto e^{-\frac{\|y\|_1}{\lambda}}$

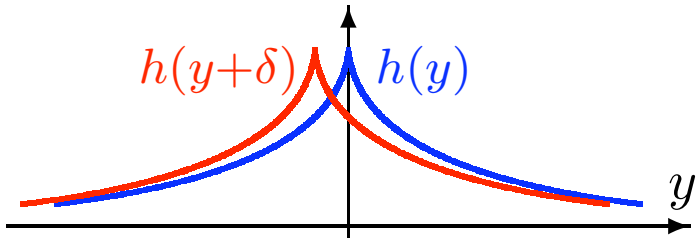


Sliding property of $\mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$: $\frac{h(y)}{h(y+\delta)} \leq e^{\varepsilon \cdot \frac{\|\delta\|}{\mathsf{GS}_f}}$ for all y, δ

Global Sensitivity [DMNS06]

Theorem: If $\mathsf{A}(\mathsf{x}) = f(\mathsf{x}) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\epsilon}\right)$, then A is ϵ-differentially private.

Laplace distribution $\mathsf{Lap}(\lambda)$ has density $h(y) \propto e^{-\frac{\|y\|_1}{\lambda}}$



Sliding property of $\mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$: $\frac{h(y)}{h(y+\delta)} \leq e^{\varepsilon \cdot \frac{\|\delta\|}{\mathsf{GS}_f}}$ for all y, δ

Proof idea:

$A(x)$: blue curve

$A(x')$: red curve

$\delta = f(x) - f(x') \leq \mathsf{GS}_f$

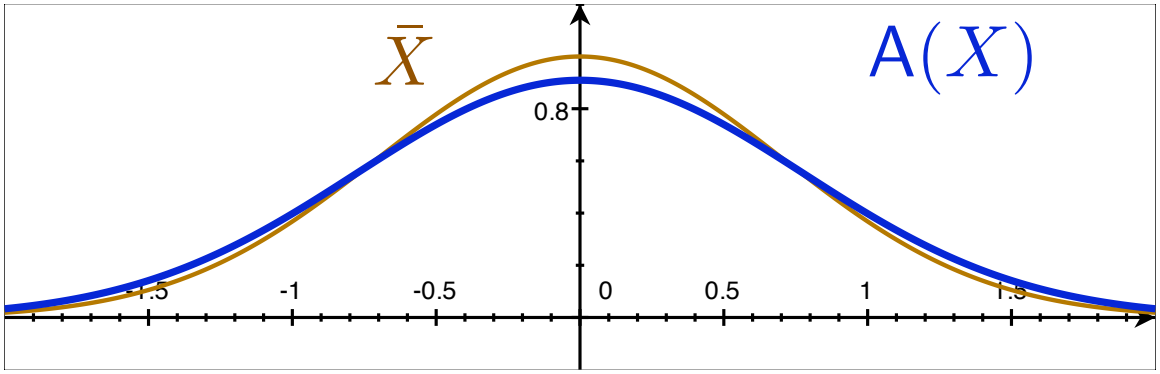
Examples of low global sensitivity

- Many natural functions have low GS, e.g.:
  - Sample mean
  - Histograms and contingency tables
  - Covariance matrix
  - Estimators with uniformly bounded sensitivity curve
  - Distance to a property
  - Functions that can be approximated from a random sample

- [BDMN] Many data-mining and statistical algorithms access the data via a sequence of low-sensitivity questions
  - e.g. perceptron, some EM algorithms, "SQ" learning algorithms

When Does Noise **Not** Matter?

- Average: $\mathrm{A}(x) = \bar{x} + \mathrm{Lap}(\frac{1}{\epsilon n})$
  - Suppose $X_1, X_2, X_3, \ldots, X_n$ are i.i.d. random variables
  - $\bar{X}$ is a random variable, and $\sqrt{n} \cdot (\bar{X} - \mu) \xrightarrow{\mathcal{D}} \mathrm{Normal}$
  - $\dfrac{\mathrm{A}(X) - \bar{X}}{\mathrm{StdDev}(\bar{X})} \xrightarrow{P} 0$ if $\epsilon\sqrt{n} \to \infty$ with n
  - No "cost" to privacy:
    - A(X) is "as good as" $\bar{X}$ for statistical inference*



When Does Noise **Not** Matter?

When Does Noise **Not** Matter?

- **Theorem:** For any exponential family, can release "approximately sufficient" statistics
  - Suff. stats T(X) are sums, add noise $\frac{d}{\epsilon n}$ for dimension d
  - $\frac{\mathrm{A}(X) - T(X)}{\mathrm{StdDev}(T(X))} \xrightarrow{P} 0$

When Does Noise **Not** Matter?

- **Theorem:** For any exponential family, can release "approximately sufficient" statistics
  - Suff. stats T(X) are sums, add noise $\frac{d}{\epsilon n}$ for dimension d
  - $\frac{\mathrm{A}(X) - T(X)}{\mathrm{StdDev}(T(X))} \xrightarrow{P} 0$
- **Theorem:** For any well-behaved parametric family, one can construct a private efficient estimator A, if $\epsilon \sqrt[4]{n} \to \infty$
  - A(X) converges to MLE
  - Requires additional techniques

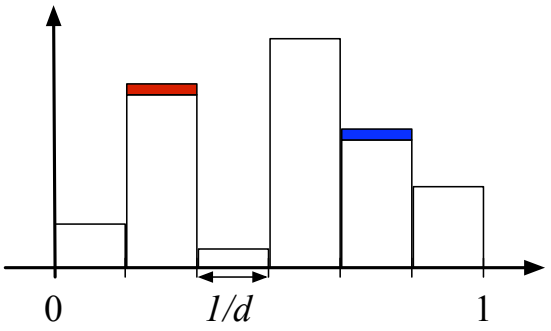
When Does Noise **Not** Matter?

- **Theorem:** For any exponential family, can release "approximately sufficient" statistics
  - Suff. stats T(X) are sums, add noise $\frac{d}{\epsilon n}$ for dimension d
  - $\frac{\mathrm{A}(X) - T(X)}{\mathrm{StdDev}(T(X))} \xrightarrow{P} 0$
- **Theorem:** For any well-behaved parametric family, one can construct a private efficient estimator A, if $\epsilon \sqrt[4]{n} \to \infty$
  - A(X) converges to MLE
  - Requires additional techniques
- Bounds gets worse as dimension increases
  - What is the "best" private estimator?

Example: Histograms

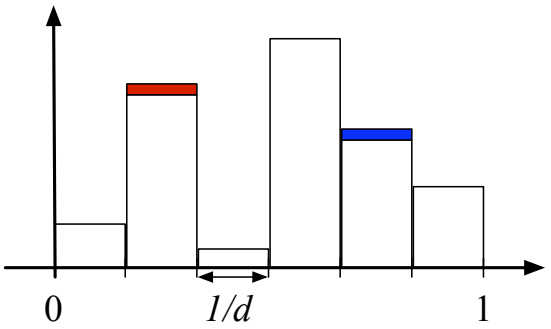
$f(x) = (n_1, n_2, ..., , n_d)$ where $n_j = \#\{i : x_i \text{ in } j\text{-th interval}\}$

$\text{Lap}(1/\epsilon)$



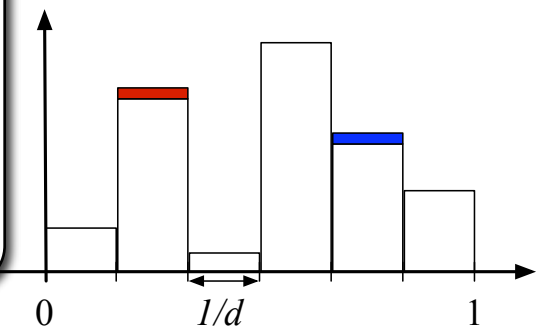
Example: Histograms

- Say $x_1, x_2, \ldots, x_n$ in [0,1]
  - Partition [0,1] into d intervals of equal size
  - $f(x) = (n_1, n_2, \ldots, , n_d)$ where $n_j = \#\{i : x_i \text{ in } j\text{-th interval}\}$
  - $GS_f = 2$
  - Sufficient to add noise $\text{Lap}(1/\epsilon)$ to each count
    - Independent of the dimension



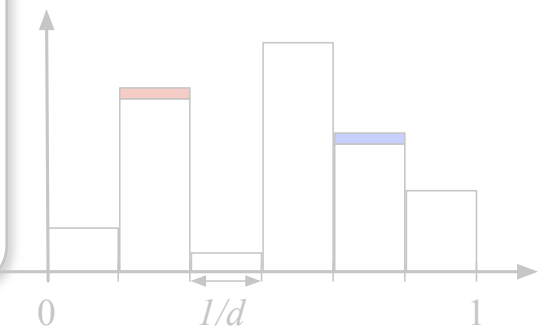
Example: Histograms

- Say $x_1, x_2, \ldots, x_n$ in [0,1]
  - Partition [0,1] into d intervals of equal size
  - $f(x) = (n_1, n_2, \ldots, , n_d)$ where $n_j = \#\{i : x_i \text{ in } j\text{-th interval}\}$
  - $GS_f = 2$
  - Sufficient to add noise $\mathrm{Lap}(1/\epsilon)$ to each count
    - Independent of the dimension
- For any smooth density h, if X_i i.i.d. ~ h, noisy histogram converges to h
  - Expected L_2 error $O(\frac{1}{\sqrt[3]{n}})$ if $\epsilon \geq \frac{1}{\sqrt[3]{n}}$
  - Same as non-private estimator



Example: Histograms

- Say $x_1, x_2, \ldots, x_n$ in ~~[0,1]~~ arbitrary domain D
  - Partition ~~[0,1] into d intervals of equal size~~ into d disjoint "bins"
  - $f(x) = (n_1, n_2, \ldots, , n_d)$ where $n_j = \#\{i : x_i$ in j-th ~~interval~~ bin
  - $GS_f = 2$
  - Sufficient to add noise $\mathrm{Lap}(1/\epsilon)$ to each count
    - Independent of the dimension
- For any smooth density h, if X_i i.i.d. ~ h, noisy histogram converges to h
  - Expected L_2 error $O(\frac{1}{\sqrt[3]{n}})$ if $\epsilon \geq \frac{1}{\sqrt[3]{n}}$
  - Same as non-private estimator



Contingency Tables

- Work horse of releases from US statistical agencies
  - Frequencies of combinations of set of categorical attributes
- Treat as a "histogram"
  - Eight bins (O+,O-,...,AB+,AB-)
  - Can add constant noise to counts
  - Change to proportions is $O(\frac{1}{n})$
  - Below sampling noise if *n* >> #bins
- Problem for practice:
  - Some entries may be negative. Multiple tables inconsistent.
  - [BCDKMT] Multiple noisy tables can be "rounded" to a consistent set of tables without increasing noise

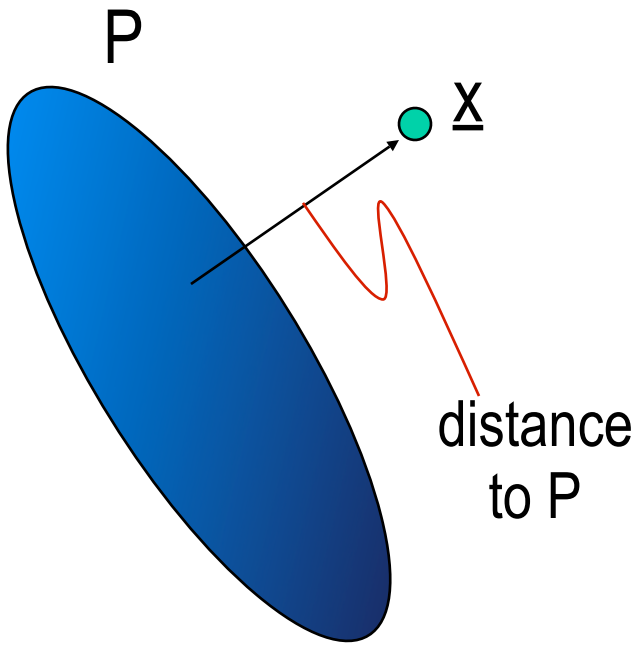
ABO and Rh Blood Type Frequencies in the United States

ABO Type	Rh Type	How Many Have It	
O	positive	38%	45%
O	negative	7%	
A	positive	34%	40%
A	negative	6%	
B	positive	9%	11%
B	negative	2%	
AB	positive	3%	4%
AB	negative	1%	

(Source: American Association of Blood Banks)

Example: Distance to a Property

- Say P = set of "good" databases
  - e.g. well-clustered databases
- Distance to P = # points in x that must be changed to make x in P
  - Always has GS = 1
- Examples:
  - Distance to good clustering
  - Weight of minimum cut in graph



Global Sensitivity Summary

- Simple framework for output perturbation with strong privacy guarantees
  - Noise levels small enough to allow meaningful analysis
- Improved in several respects
  - Worst case definition: even if f is sensitive on only one input, must add lots of noise
    - [NRS] Add less noise on "good" instances
  - One function at a time: To answer q queries, naive analysis suggests making noise increase linearly with q
    - [BLR] Simultaneously answer many "simple" questions
  - Focus on function approximation: many tasks not so simple
    - Auction design [MT], supervised learning [KLNRS]

This talk

- "Differential" privacy
  - Handles arbitrary external information
  - What can we compute privately?
- Example technique: Output perturbation
  - Calibrating noise to "sensitivity"
  - Sample-aggregate methodology

High Global Sensitivity: Median

Example 1: median of $x_1, \ldots, x_n \in [0, 1]$

$$x = \underbrace{0 \cdots 0}_{\frac{n-1}{2}} 0 \underbrace{1 \cdots 1}_{\frac{n-1}{2}} \qquad x' = \underbrace{0 \cdots 0}_{\frac{n-1}{2}} 1 \underbrace{1 \cdots 1}_{\frac{n-1}{2}}$$

$$\text{median}(x) = 0 \qquad \text{median}(x') = 1$$

$$\mathsf{GS}_{\text{median}} = 1$$

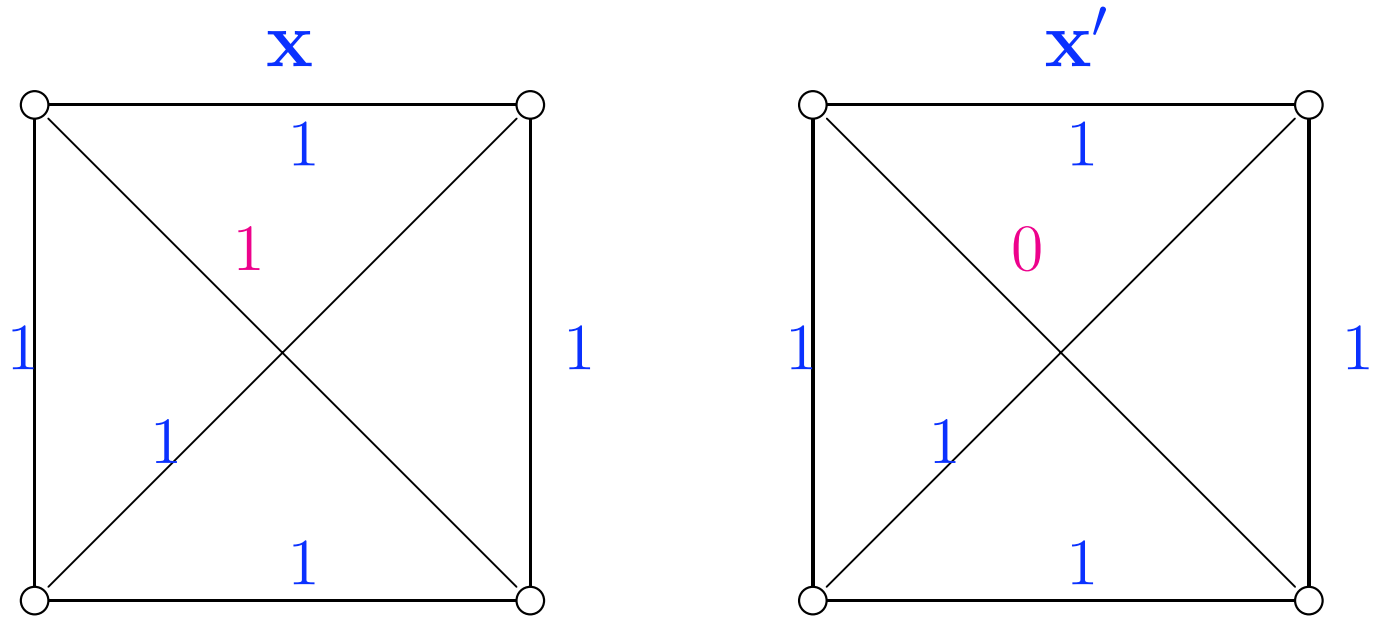
- Noise magnitude: $\frac{1}{\varepsilon}$. Too much noise!
- But for most neighbor databases x, x',

  $|\text{median}(x) - \text{median}(x')|$ is small.
- Can we add less noise on "good" instances?

High Global Sensitivity: MST Cost

Example 2: the weight of a minimum spanning tree

Database entries: edge weights in the range $[0, 1]$.

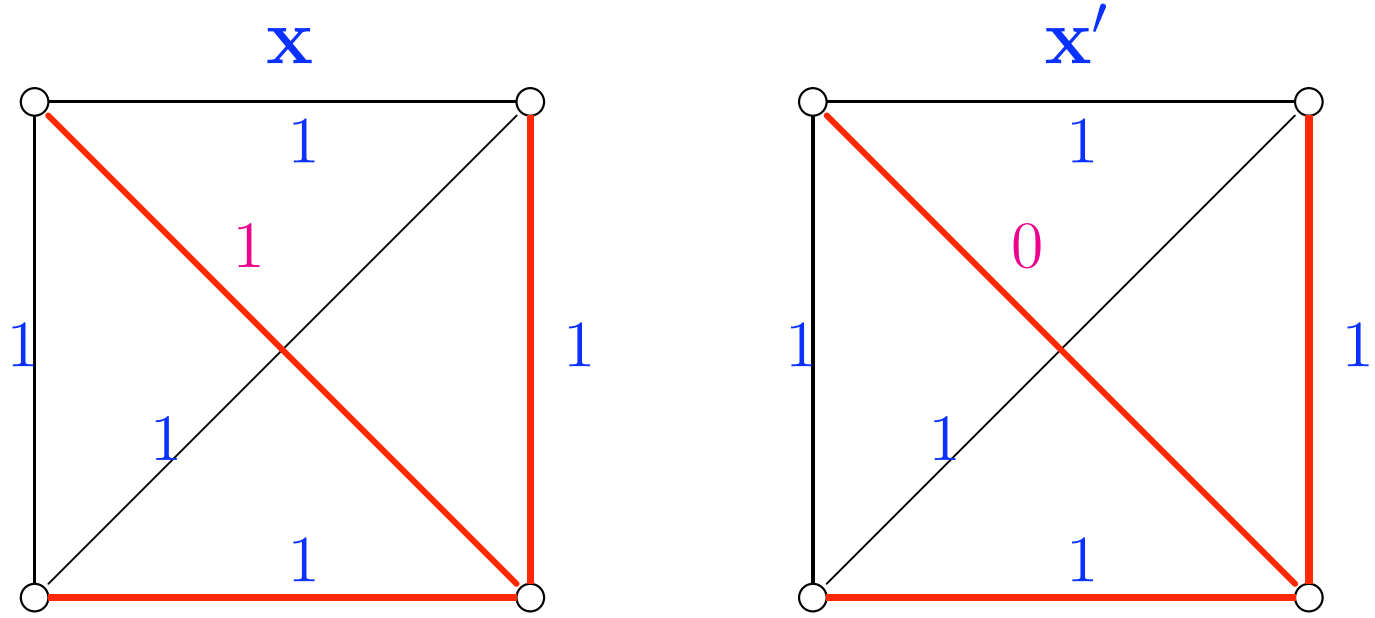


$$\mathsf{GS}_{\text{MST-weight}} = 1$$

High Global Sensitivity: MST Cost

Example 2: the weight of a minimum spanning tree

Database entries: edge weights in the range $[0, 1]$.

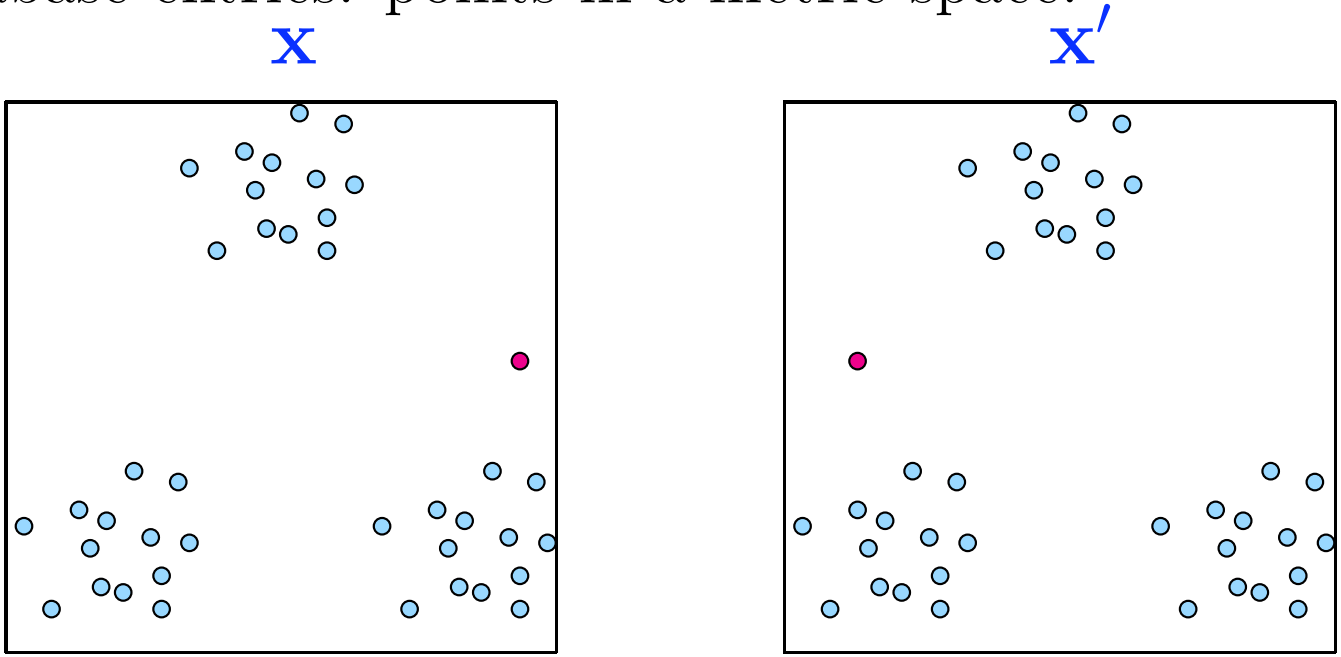


MST-weight$(x) = 3$ MST-weight$(x) = 2$

$$\mathsf{GS}_{\text{MST-weight}} = 1$$

High Global Sensitivity: Cluster centers

Database entries: points in a metric space.

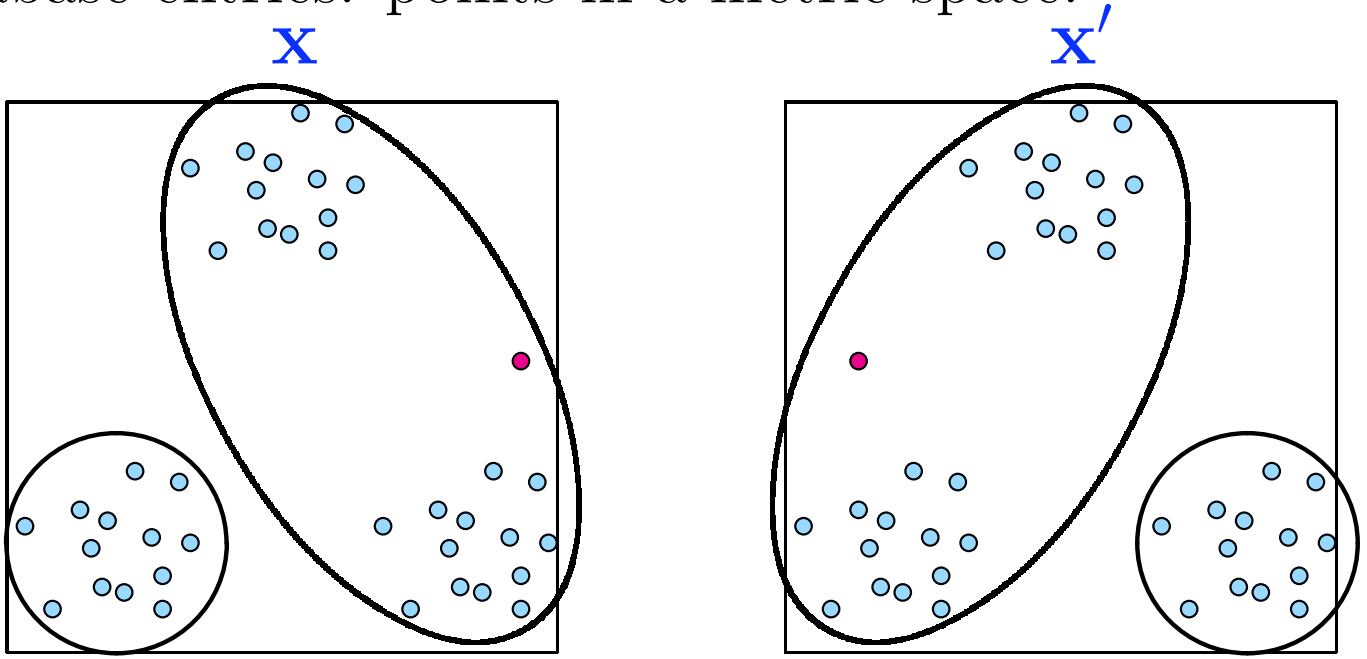


Global sensitivity of cluster centers is roughly the diameter of the space.

- But intuitively, if clustering is "good", cluster centers should be insensitive.

High Global Sensitivity: Cluster centers

Database entries: points in a metric space.

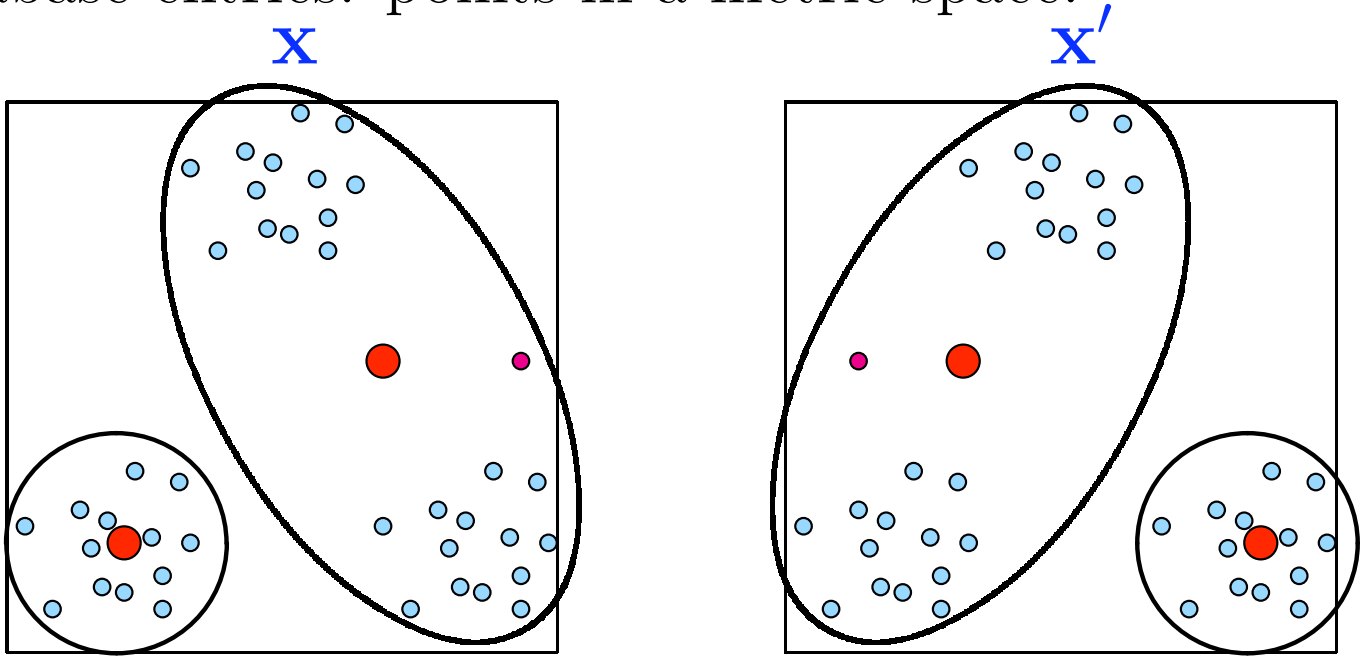


Global sensitivity of cluster centers is roughly the diameter of the space.

- But intuitively, if clustering is "good", cluster centers should be insensitive.

High Global Sensitivity: Cluster centers

Database entries: points in a metric space.



Global sensitivity of cluster centers is roughly the diameter of the space.

- But intuitively, if clustering is "good", cluster centers should be insensitive.

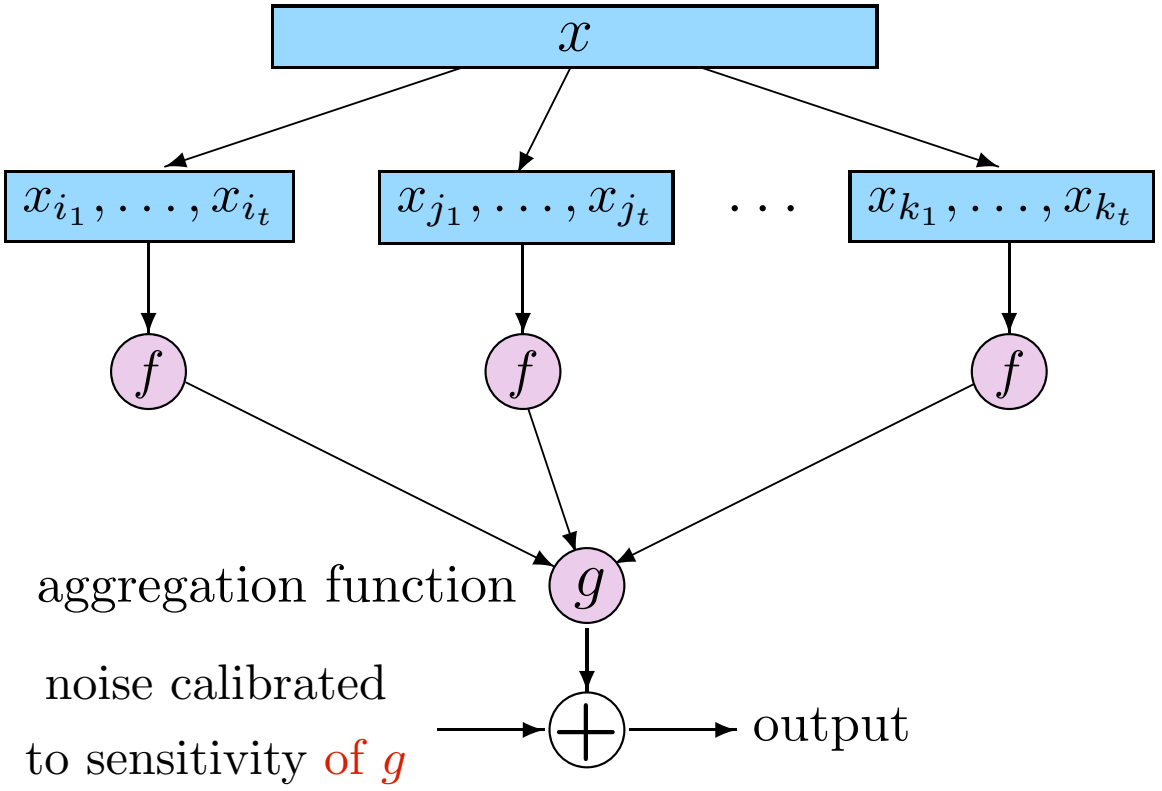
Getting Around Global Sensitivity

- Local sensitivity measures variability in neighborhood of specific data set [Nissim-Raskhodnikova-S, *STOC* 2007]
  - ➢ Connections to robust statistics
    - Bounded influence function implies expected local sensitivity is small
  - ➢ Local sensitivity needs to be smoothed
    - Interesting algorithmic/geometric problems
  - ➢ Not this talk
- Instead: Generic framework for smoothing functions so they have low sensitivity

Sample-and-Aggregate Methodology

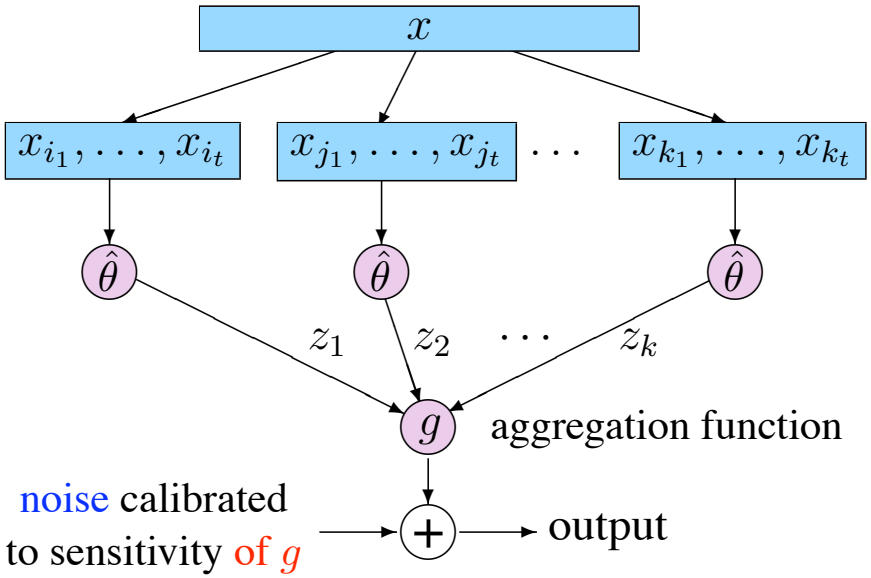
Intuition: Replace f with a less sensitive function $\tilde{f}$.

$$\tilde{f}(x) = g(f(sample_1), f(sample_2), \ldots, f(sample_s))$$



Example: Efficient Point Estimates

- Given a parametric model $\{f_\theta : \theta \in \Theta\}$
- MLE $= \text{argmax}_\theta(f_\theta(x))$
- Converges to Normal
  - Bias(MLE) = O(1/n)
  - Can be corrected so that bias($\hat{\theta}$) = O($n^{-3/2}$)



- **Theorem**: If model is well-behaved, then sample-aggregate using $\hat{\theta}$ gives efficient estimator if $\epsilon n^{1/4} \to \infty$

- Question: What is the best private estimator?
  - Error bounds degrade with dimension...

Sample-and-Aggregate Methodology

Theorem

If f can be approximated on x

from small samples

then f can be released with little noise

Sample-and-Aggregate Methodology

Theorem

If f can be approximated on x within distance r
from small samples of size $n^{1-\delta}$
then f can be released with little noise $\approx \frac{r}{\varepsilon} + negl(n)$

Sample-and-Aggregate Methodology

Theorem

If f can be approximated on x within distance r from small samples of size $n^{1-\delta}$ then f can be released with little noise $\approx \frac{r}{\varepsilon} + negl(n)$

- Works in several different metric spaces
- Example application: clustering
  - I.i.d. random inputs: parametric estimation of mixture models
  - Arbitrary inputs: approximate optimal k-means clustering if data is "separated" à la [OstrovksyRabaniSchulmanSwamy'06]

Conclusions

- Define privacy in terms of my effect on output
  - Meaningful despite arbitrary external information
  - I should participate if I get benefit
- What can we compute privately?
  - Lots of recent work
  - Existing techniques work best for highly structured computations. What about graph data, text, searches, ...?
- Data privacy is now (even) more challenging than in past
  - Data vastly more varied and valuable
  - External information more available
  - How should we think about data privacy? (This is one example.)