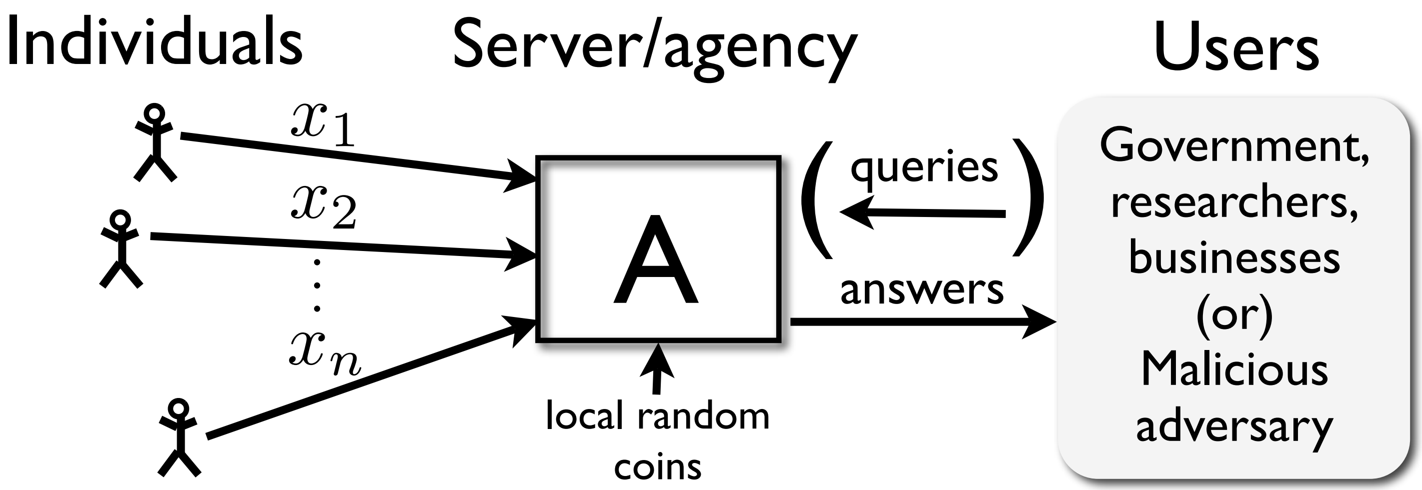
Integrating Differential Privacy with Statistical Theory

Adam Smith
Computer Science & Engineering Department
Penn State

NCHS/CDC Workshop on Data Confidentiality
May 1, 2008

Differential Privacy



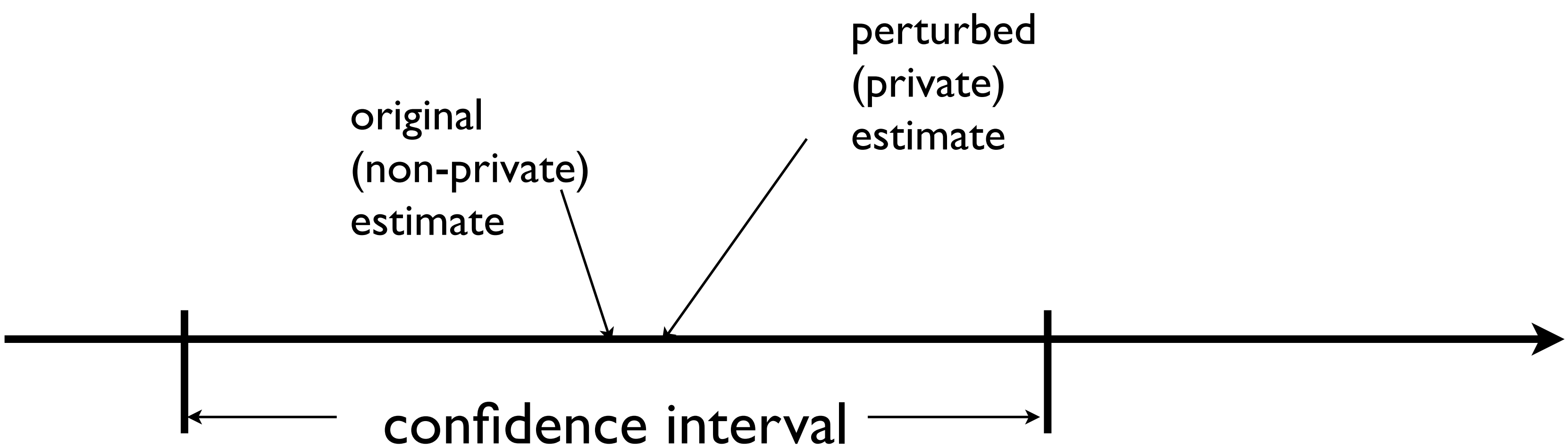
- Definition of privacy in statistical databases
  - Imposes restrictions on algorithm A generating output
- If A satisfies restrictions, then output provides privacy no matter what user/intruder knows ahead of time
- **Question**: how **useful** are algorithms that satisfy differential privacy?

This talk: Useful Statistical Inference

- Two situations where differential privacy compatible with statistical methodology
- In both cases: construct differentially private algorithm with same asymptotic error as best non-private algorithm
  - ➢ **Parametric**: for any* parametric model, there exists a private, efficient estimator (i.e. minimal variance)
  - ➢ **Nonparametric**: for any* distribution on [0,1], there is a private histogram estimator with same convergence rate as best (non-private) fixed-width estimator

Main Idea for both cases

- Add noise to carefully modified estimator
  - Several ways to design differentially private algorithms
  - Adding noise is the simplest
- Prove that required noise is less than inherent variability due to sampling



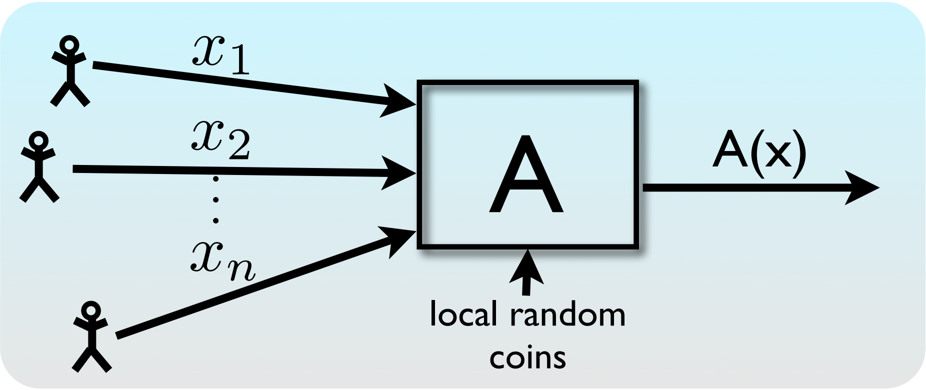
Bigger Goal

- Understanding how rigorous notions of privacy relate to statistical inference
  - (Also: crossing disciplinary boundaries requires understanding, and working with, other communities' language)
- First step: basic asymptotic theory
  - Cornerstone of statistical techniques
  - Qualitative statements
    - asymptotic regime allows for clean statements
    - highlights where techniques breakdown
  - Intuition for messier real settings

Reminder: differential privacy

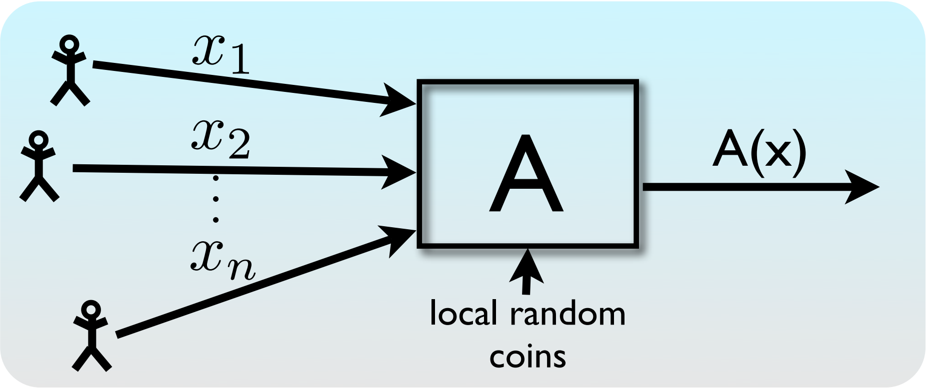
- Intuition:
  - Changes to my data not noticeable by users
  - Output is "independent" of my data

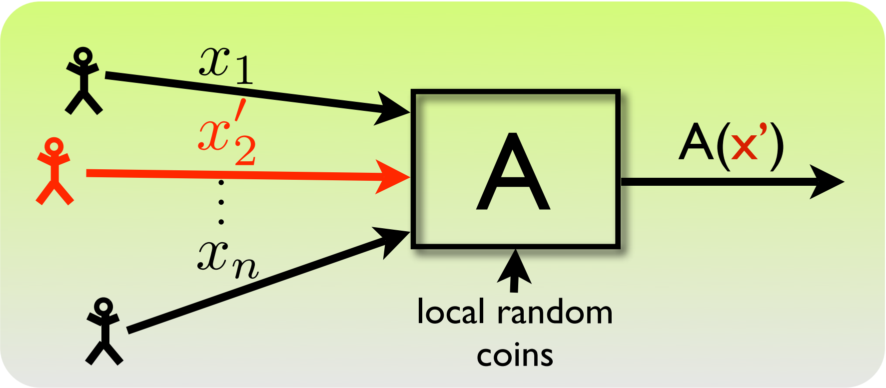
Defining Privacy [DiNi,DwNi,BDMN,DMNS]



- 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.

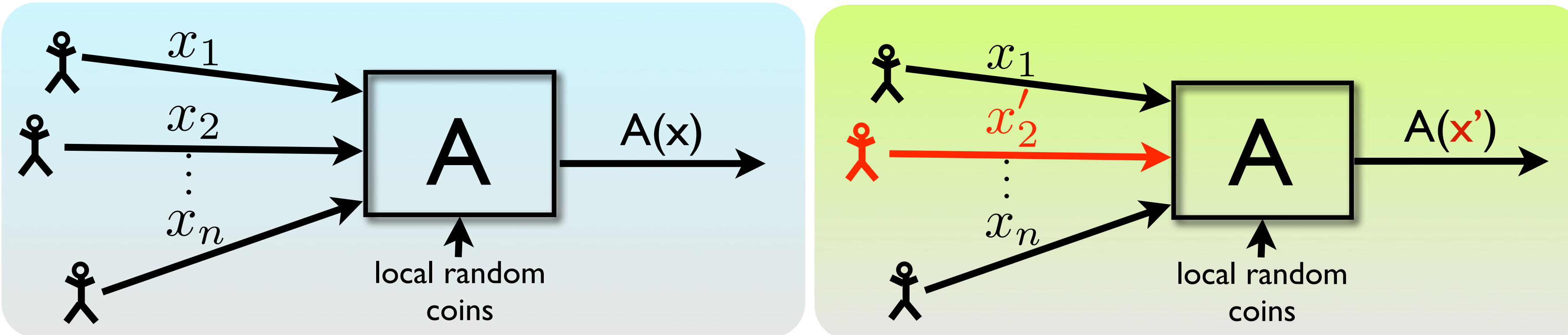
Defining Privacy [DiNi,DwNi,BDMN,DMNS]





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

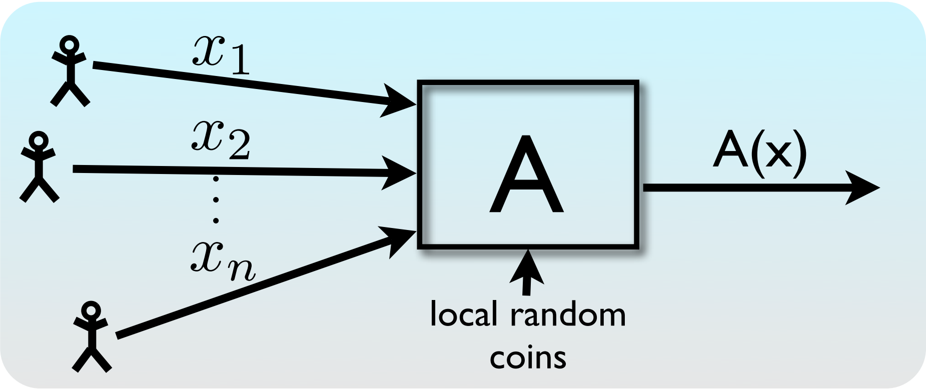
Defining Privacy [DiNi,DwNi,BDMN,DMNS]

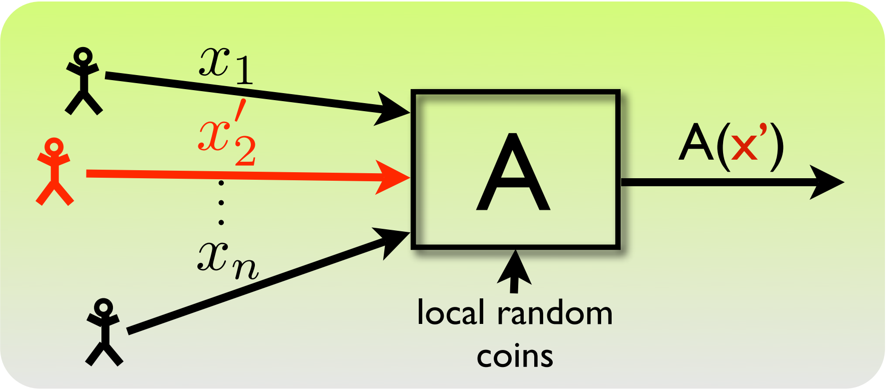


x' is a neighbor of x
if they differ in one data point

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]

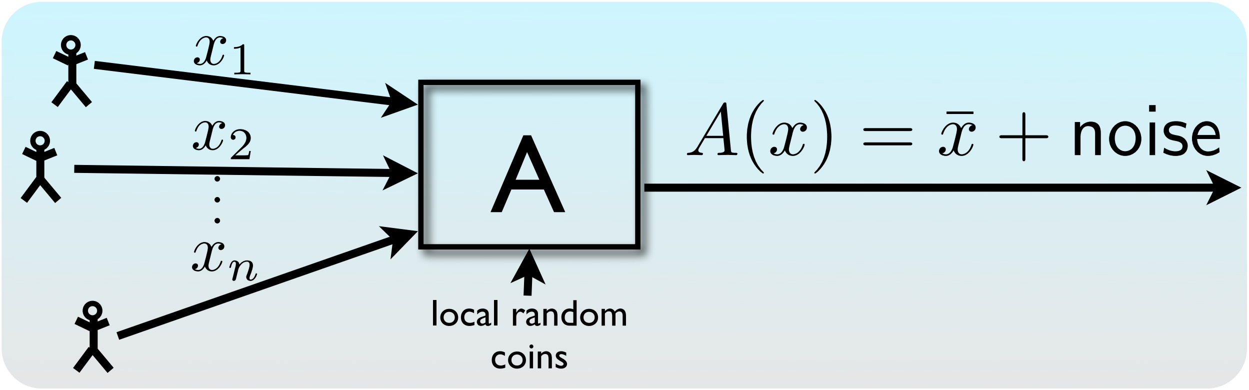
- ϵ cannot be too small (think $\frac{1}{10}$, not $\frac{1}{2^{50}}$)
- This is a condition on the **algorithm** (process) A
  - Saying "this output is safe" doesn't take into account how it was computed
- Meaningful semantics no matter what user knows ahead of time

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

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

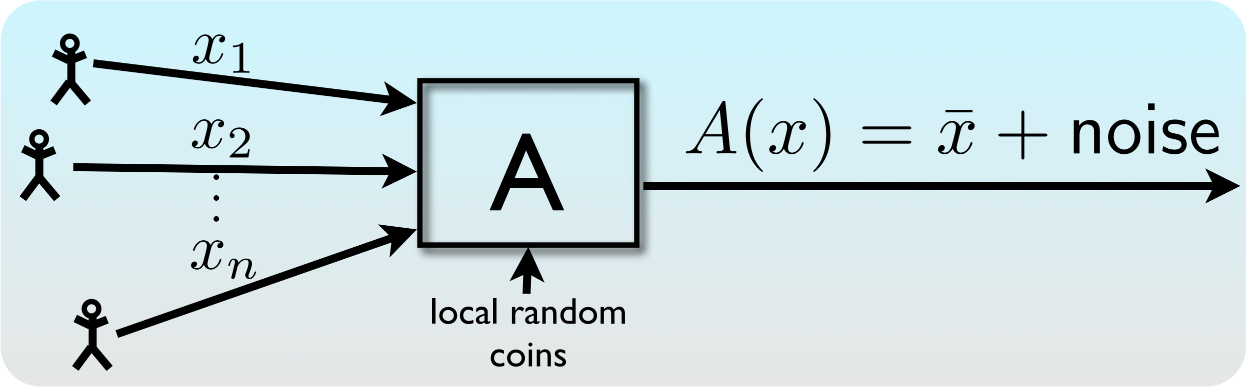
Example: Perturbing the Average



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

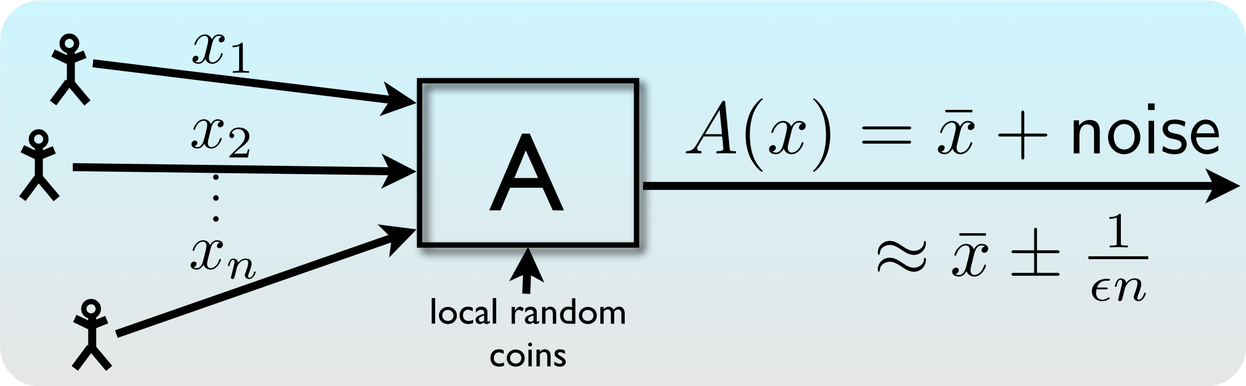
$$\bar{x} = \frac{1}{n} \sum_i x_i$$

Example: Perturbing the Average



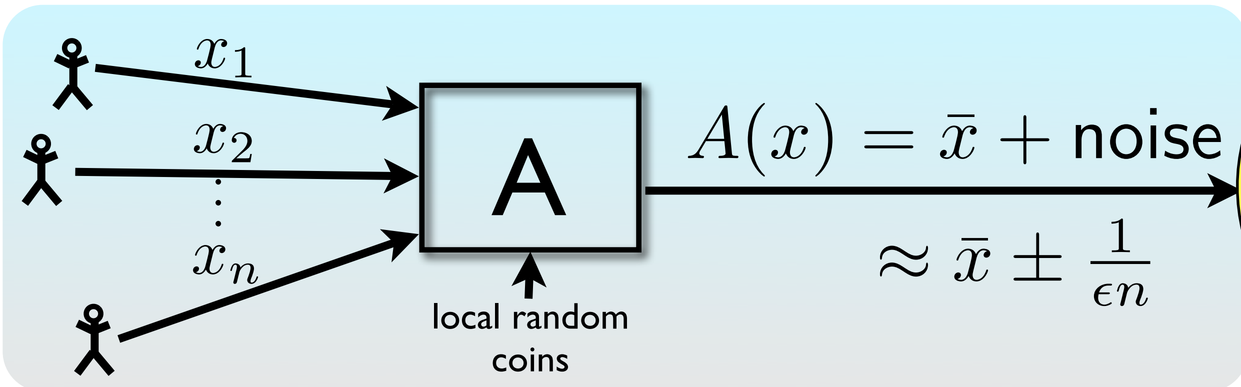
- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release sample mean $\bar{x} = \frac{1}{n}\sum_i x_i$

Example: Perturbing the Average



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

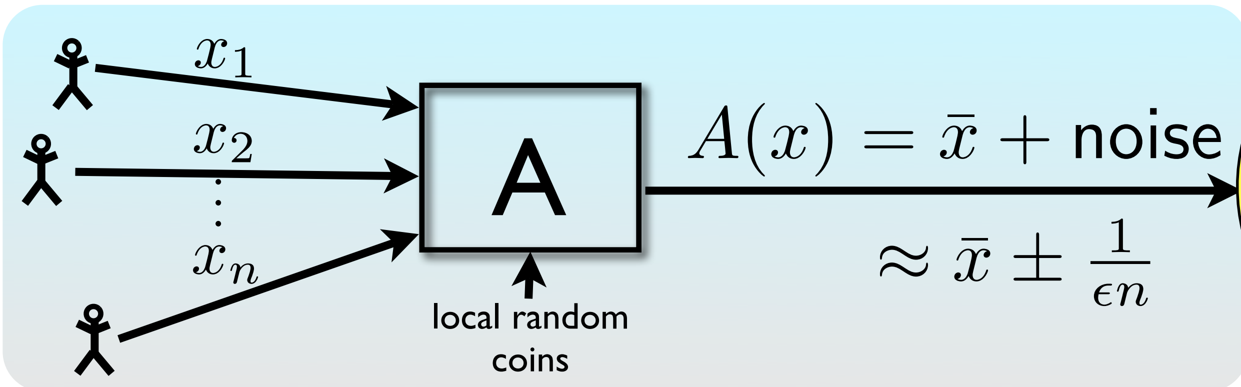
Example: Perturbing the Average



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

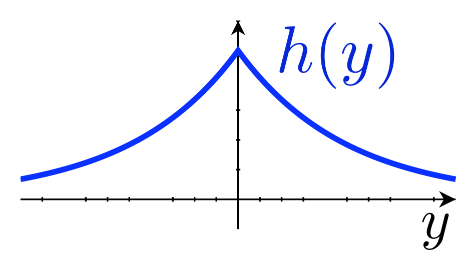
- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release sample mean $\bar{x} = \frac{1}{n} \sum_i x_i$
- **Claim**: If noise $\sim$ Lap$(\frac{1}{\epsilon n})$ then A is ϵ-differentially private

Example: Perturbing the Average

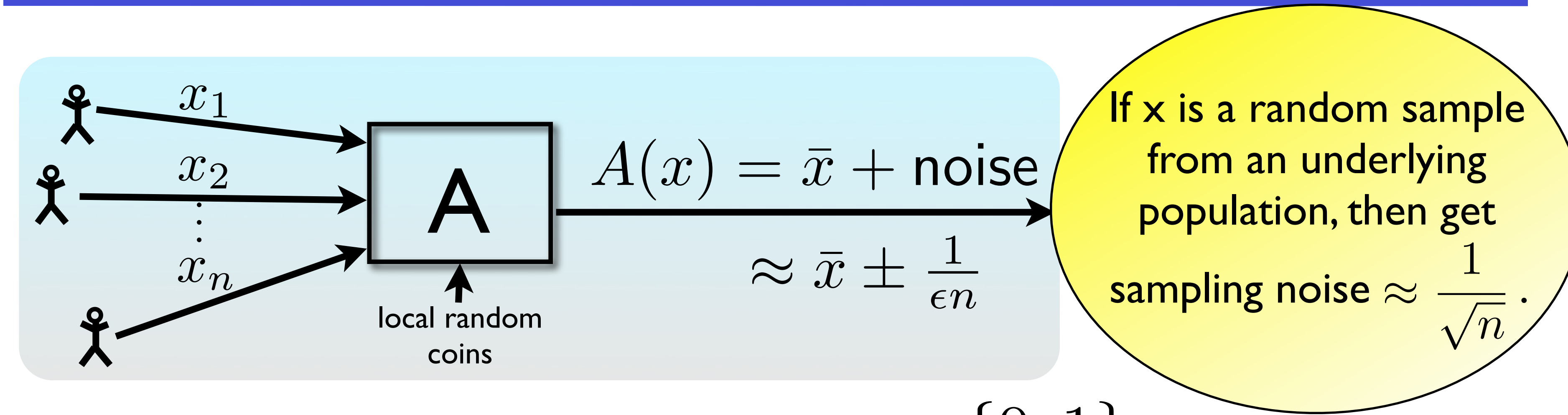


If x is a random sample from an underlying population, then get sampling noise $\approx \frac{1}{\sqrt{n}}$.

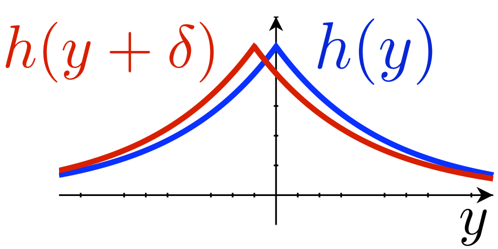
- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release sample mean $\bar{x} = \frac{1}{n} \sum_i x_i$
- **Claim**: If noise $\sim$ Lap$(\frac{1}{\epsilon n})$ then A is ϵ-differentially private
  - ➢ Laplace distribution Lap(λ) has density $h(y) \propto e^{-|y|/\lambda}$
  -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



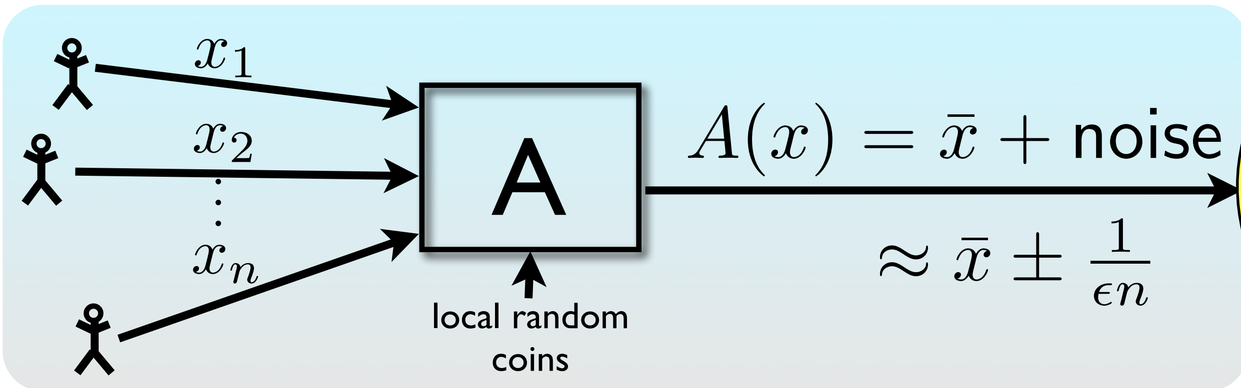
Example: Perturbing the Average



- Data points are binary responses $x_i \in \{0, 1\}$
- Server wants to release sample mean $\bar{x} = \frac{1}{n} \sum_i x_i$
- **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: $\frac{h(y)}{h(y+\delta)} \leq e^{\delta/\lambda}$

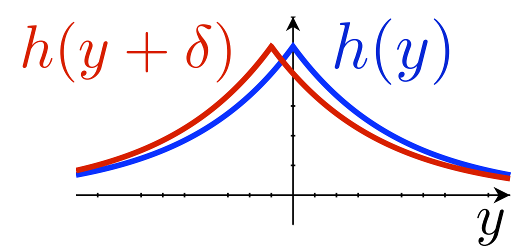


Example: Perturbing the Average



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 sample mean $\bar{x} = \frac{1}{n} \sum_i x_i$
- **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
  - ➢ $\delta = |\bar{x} - \bar{x}'| \leq \frac{1}{n} \implies \frac{\text{blue curve}}{\text{red curve}} \leq e^{\epsilon}$

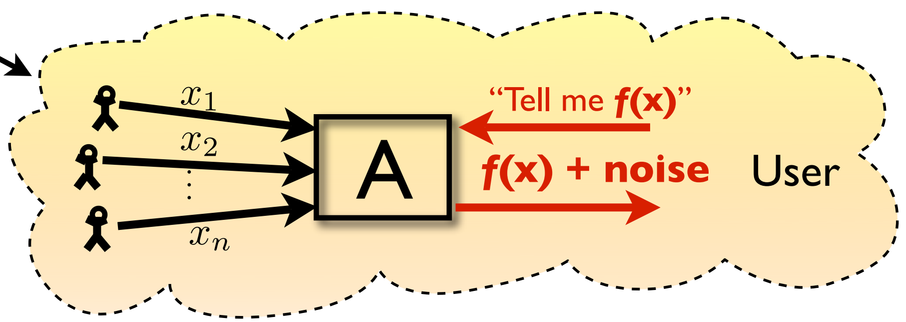


What can we compute privately?

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

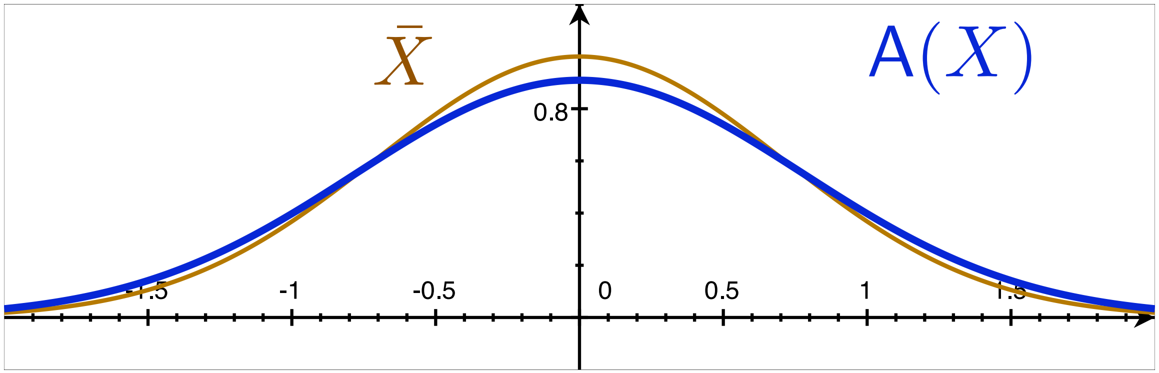
What computational tasks can we achieve privately?

- 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]



When Does Noise **Not** Matter?

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



When Does Noise **Not** Matter?

$\sqrt{d}$

When Does Noise **Not** Matter?

- Mean example generalizes to other statistics
- **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
  - $\frac{\text{A}(X) - T(X)}{\text{StdDev}(T(X))} \xrightarrow{P} 0$

$\sqrt{d}$

When Does Noise **Not** Matter?

- Mean example generalizes to other statistics
- **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
  - $\frac{\mathrm{A}(X) - T(X)}{\mathrm{StdDev}(T(X))} \xrightarrow{P} 0$
- Asymptotic result: Indicates that useful analysis possible
  - Requires more sophisticated processing for small n

$\sqrt{d}$

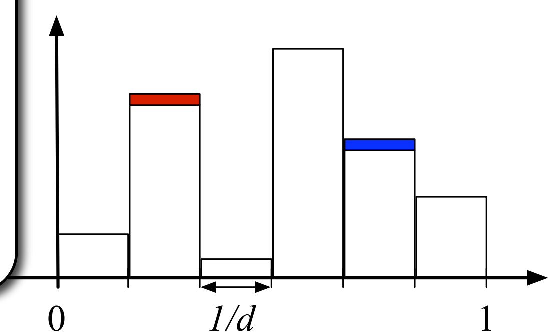
When Does Noise **Not** Matter?

- Mean example generalizes to other statistics
- **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
  - $\frac{\mathsf{A}(X) - T(X)}{\mathrm{StdDev}(T(X))} \xrightarrow{P} 0$
- Asymptotic result: Indicates that useful analysis possible
  - Requires more sophisticated processing for small n
- Noise degrades with dimension (can get noise ~ $\sqrt{d}$)
  - More information $\Longrightarrow$ less privacy
  - Research question: Is this necessary?

Two More Examples

- **Theorem:** For any well-behaved parametric family, one can construct a private efficient estimator A, if $\epsilon\sqrt[4]{n} \rightarrow \infty$
  - A(X) converges to MLE

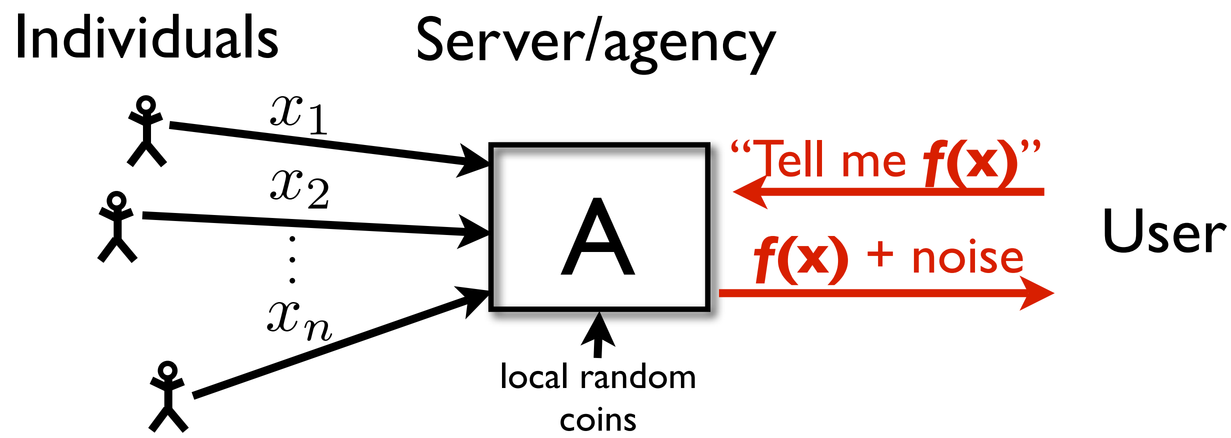
- For any smooth density h, if X_i i.i.d. ~ h, noisy histogram converges to h
  - Expected L_2 error $O(\frac{1}{\sqrt[3]{n}})$ if $\epsilon \geq \frac{1}{\sqrt[3]{n}}$



- Histogram Density Estimation
  - Calibrating noise to sensitivity

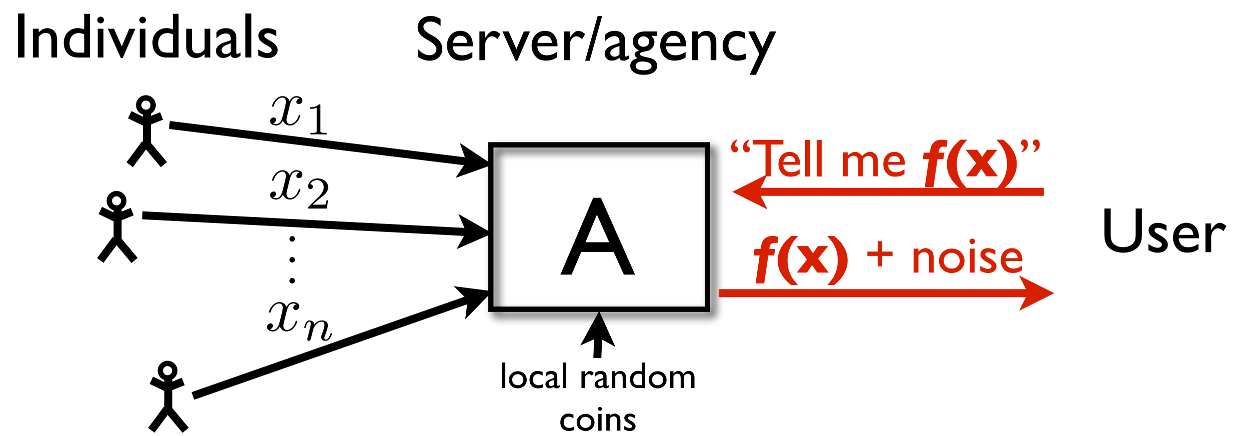
- Maximum Likelihood Estimator
  - Sub-sample and aggregate

Output Perturbation, more generally



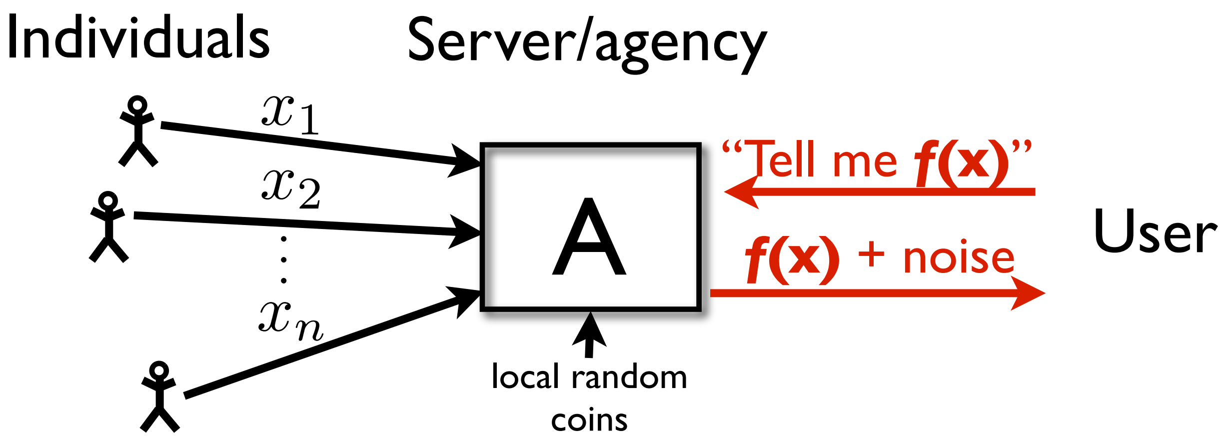
- May be interactive
  - Non-interactive: release pre-defined summary stats + noise
  - Interactive: respond to user requests
- May be repeated many times
  - Composition: q releases are jointly $q\epsilon$-differentially private
- How much noise is enough? (How much is too much?)

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: $\boxed{\mathrm{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1}$
- Example: $\mathrm{GS}_{\text{average}} = \frac{1}{n}$

Global Sensitivity [DMNS06]



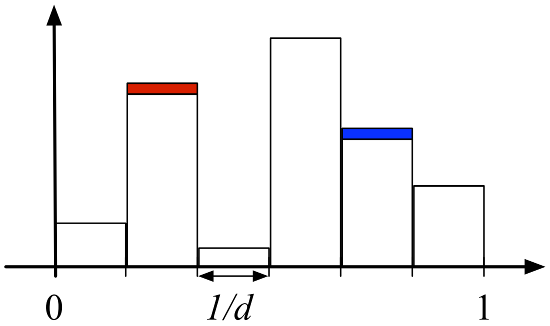
- Intuition: ***f*(x)** can be released accurately when ***f*** is insensitive to individual entries $x_1, x_2, \ldots, x_n$
- Global Sensitivity: $\mathrm{GS}_f = \max_{\text{neighbors } x, x'} \|f(x) - f(x')\|_1$
- Example: $\mathrm{GS}_{\text{average}} = \frac{1}{n}$

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

Example: Histograms

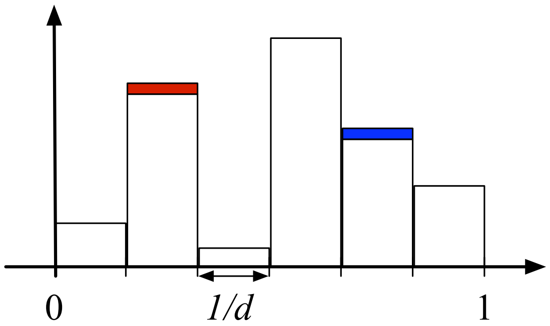
$f(x) = (n_1, n_2, ..., , n_d)$ where $n_j = \#\{i : x_i \text{ in } j\text{-th interval}\}$

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



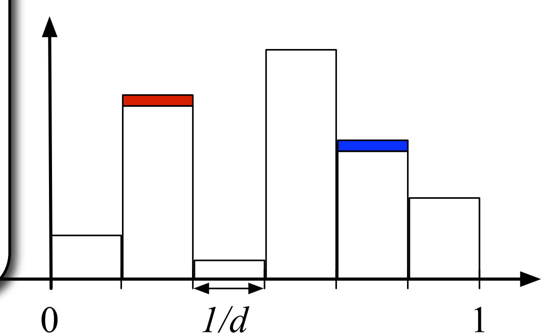
Example: Histograms

- 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 $n_j = \#\{i : x_i$ in j-th interval$\}$
  - $GS_f = 2$
  - Sufficient to add noise $\text{Lap}(1/\epsilon)$ to each count
    - Independent of the dimension



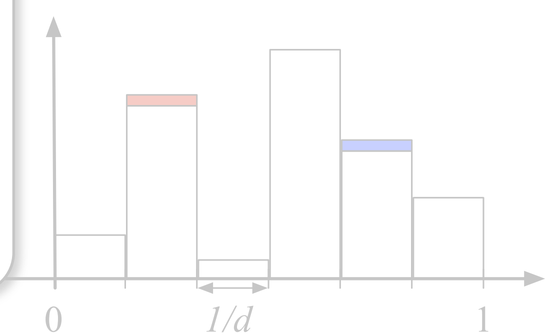
Example: Histograms

- 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 $n_j = \#\{i : x_i$ in j-th interval$\}$
  - $GS_f = 2$
  - Sufficient to add noise $\text{Lap}(1/\epsilon)$ 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 $x_1, x_2, \dots, x_n$ in ~~[0,1]~~ arbitrary domain D
  - Partition ~~[0,1] into d intervals of equal size~~ into d disjoint "bins"
  - $f(x) = (n_1, n_2, \dots, , n_d)$ where $n_j = \#\{i : x_i$ in j-th ~~interval}~~ bin
  - $GS_f = 2$
  - Sufficient to add noise $\text{Lap}(1/\epsilon)$ 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



More detail

- This actually shows that for any given bin width, can find noisy estimator that is close to non-noisy estimator

- Does not address how to choose bin width
  - Subject to extensive research
  - Common "bandwidth selection" criteria can be approximated privately
  - Two-stage process

- Histogram Density Estimation
  - Calibrating noise to sensitivity
- Maximum Likelihood Estimator
  - Sub-sample and aggregate

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?

What about MLE?

- Sometimes MLE is well-behaved,
  - e.g. observed proportion for binomial
- Sometimes we have no idea
  - e.g. no closed form expression for mildly complex loglinear models
  - Can have arbitrarily bad sensitivity
  - NB: Similar problems faced by robust statistics

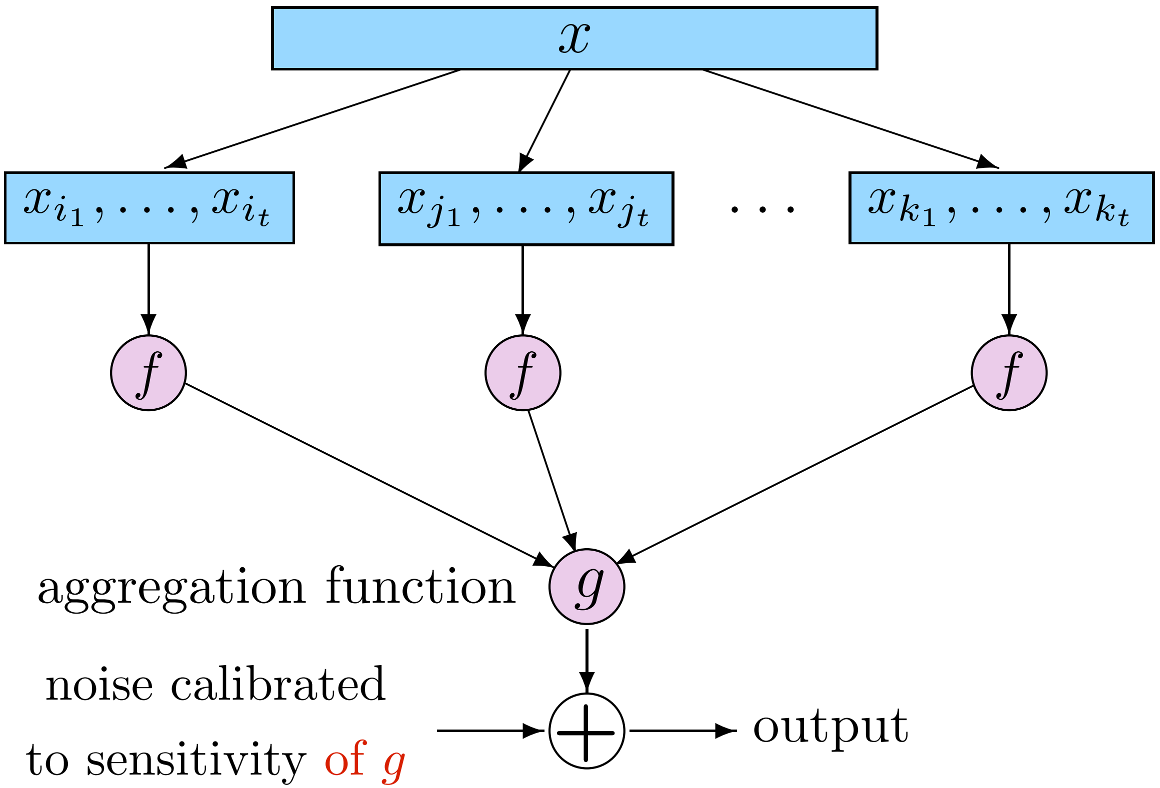
Getting Around Global Sensitivity

- 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
  - ➢ No need to "understand" structure of function

Sample-and-Aggregate Methodology

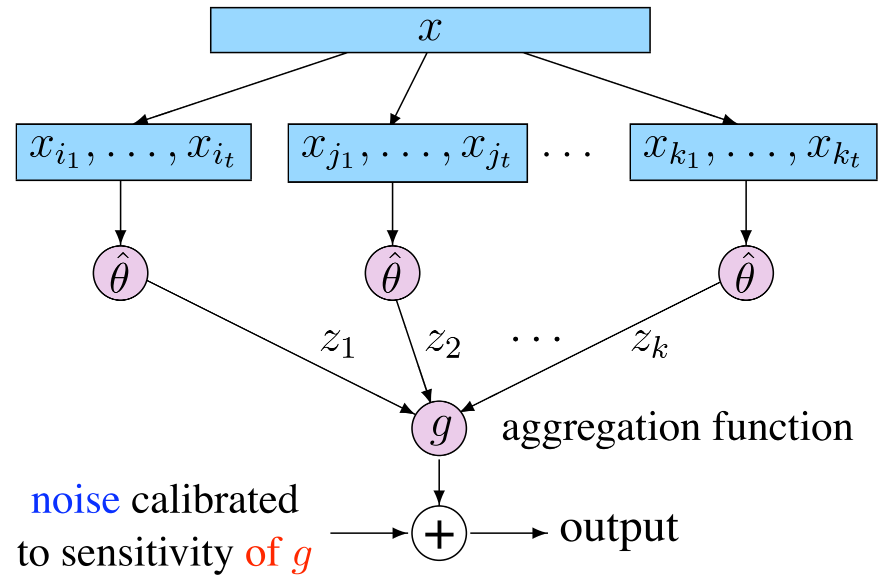
Intuition: Replace f with a less sensitive function $\tilde{f}$.

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



Example: Efficient Point Estimates

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

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?
  - This talk: statistical estimators that are "as good" as optimal non-private estimators
  - New aspect to "curse" of dimensionality
- Data privacy is now (even) more challenging than in past
  - Data vastly more varied and valuable
  - External information more available
  - How can we reason rigorously about data privacy?