Pinning Down “Privacy” in Statistical Databases

Adam Smith

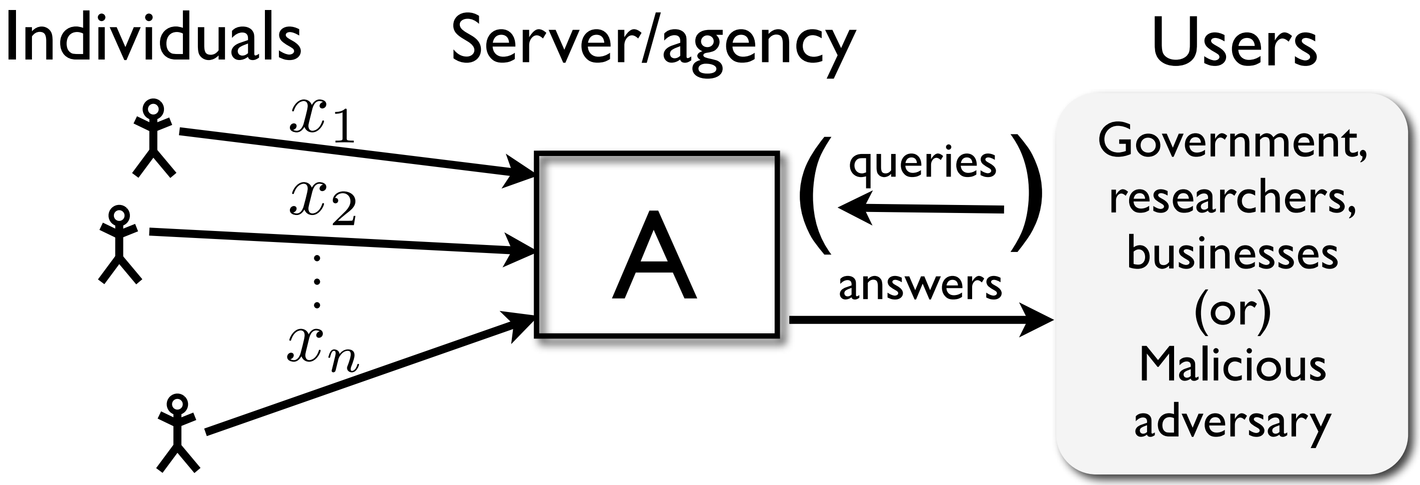
Computer Science & Engineering Department

Penn State

http://www.cse.psu.edu/~asmith

Crypto Tutorial, August 21, 2012

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

Privacy in Statistical Databases

- **Two conflicting goals**
  - Utility: Users can extract "aggregate" statistics
  - "Privacy": Individual information stays hidden

Privacy in Statistical Databases

- **Two conflicting goals**
  - Utility: Users can extract "aggregate" statistics
  - "Privacy": Individual information stays hidden
- **How can we define these precisely?**
  - Variations on model studied in
    - Statistics ("statistical disclosure control")
    - Data mining / database ("privacy-preserving data mining" *)
  - Since ~2002: Rigorous foundations & analysis

Privacy & Crypto

Image: Gary Larson

Privacy & Crypto

- No bright lines
  - ➢ Crypto: psychiatrist and patient
  - ➢ Data privacy: have to release some data at the expense of others

Image: Gary Larson

Privacy & Crypto

- No bright lines
  - Crypto: psychiatrist and patient
  - Data privacy: have to release some data at the expense of others
- Different from secure function evaluation
  - SFE: how do we securely distribute a computation we've agreed on?
  - Data privacy: what computation should we perform?

Image: Gary Larson

Privacy & Crypto

- How can crypto contribute?
  - Modeling
  - Attacks ("cryptanalysis")
    - More hacking!
    - Coherent principles
  - Distributed models
- How can crypto benefit?
  - Theory of "moderate" security
  - Applicable to areas such as anonymous communication, voting?

An overview of research on privacy?

An overview of research on privacy?

- Data privacy research is diverse
  - Researchers from crypto, learning, algorithms, databases, ...
  - Tools from lots of areas

An overview of research on privacy?

- Data privacy research is diverse
  - Researchers from crypto, learning, algorithms, databases, ...
  - Tools from lots of areas
- Great progress
  - We're 10 years ahead of where we were in 2002
  - Area still immature

An overview of research on privacy?

- Data privacy research is diverse
  - Researchers from crypto, learning, algorithms, databases, ...
  - Tools from lots of areas
- Great progress
  - We're 10 years ahead of where we were in 2002
  - Area still immature
- This talk
  - More tutorial than survey
  - Much has been left out
  - Not only my work
  - Sparse on references

An overview of research on privacy?

- Data privacy research is diverse
  - Researchers from crypto, learning, algorithms, databases, ...
  - Tools from lots of areas
- Great progress
  - We’re 10 years ahead of where we were in 2002
  - Area still immature
- This talk
  - More tutorial than survey
  - Much has been left out
  - Not only my work
  - Sparse on references

This talk

This talk

- **Act I: Attacks**
  - (Why is privacy hard?)
  - Reconstruction attacks

This talk

- **Act I: Attacks**
  - (Why is privacy hard?)
  - Reconstruction attacks
- **Act II: Definitions**
  - One approach: "differential" privacy
  - Variations on the theme

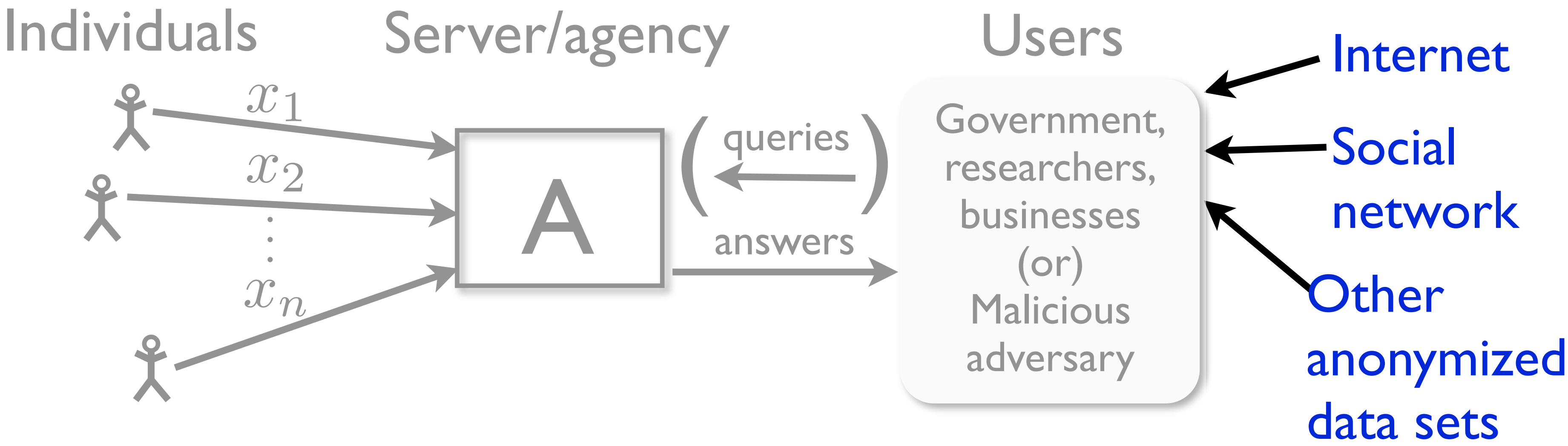
This talk

- **Act I: Attacks**
  - (Why is privacy hard?)
  - Reconstruction attacks
- **Act II: Definitions**
  - One approach: "differential" privacy
  - Variations on the theme
- **Act III: Algorithms**
  - Basic techniques: noise addition, exponential sampling
  - Answering many queries
  - Exploiting "local" sensitivity

This talk

- **Act I: Attacks**
  - ➢ (Why is privacy hard?)
  - ➢ Reconstruction attacks
- **Act II: Definitions**
  - ➢ One approach: "differential" privacy
  - ➢ Variations on the theme
- **Act III: Algorithms**
  - ➢ Basic techniques: noise addition, exponential sampling
  - ➢ Answering many queries
  - ➢ Exploiting "local" sensitivity

External Information



- Users have external information sources
  - ➢ Can't assume we know the sources
  - ➢ Can't ignore them!
- Anonymization schemes are regularly broken

- Warm-up: fine-grained releases
  - Netflix
  - Composition
- Reconstruction attacks
  - Based on approximate linear statistics
  - Based on synthetic data

Netflix Data Release [Narayanan, Shmatikov 2008]

- Ratings for subset of movies and users
- Usernames replaced with random IDs
- Some additional perturbation

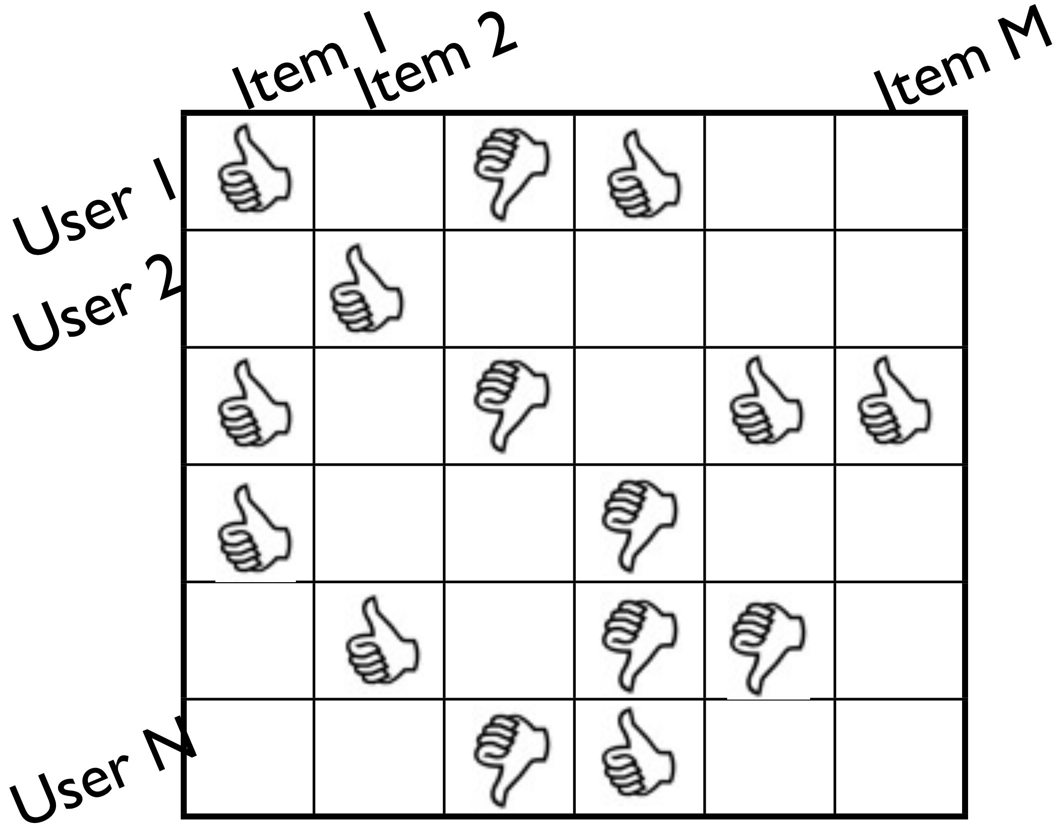
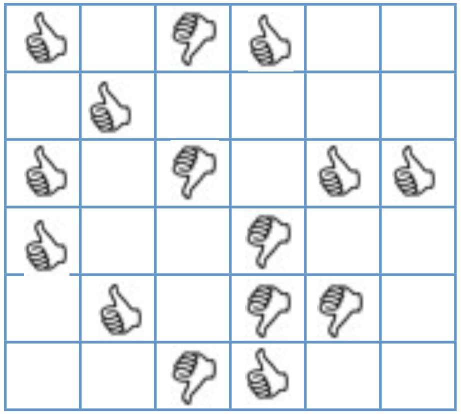
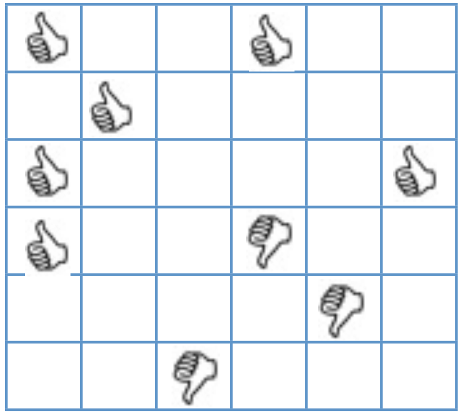


Image credit: Arvind Narayanan

Netflix Data Release [Narayanan, Shmatikov 2008]



+



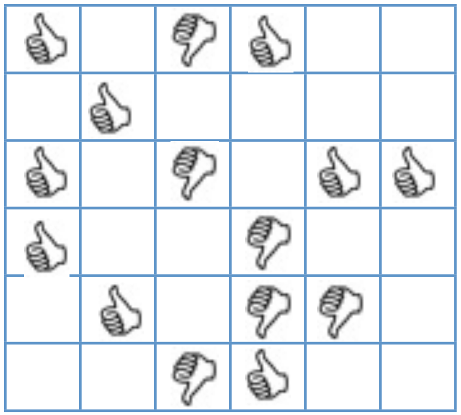
Alice
Bob
Charlie
Danielle
Erica
Frank

Anonymized
NetFlix data

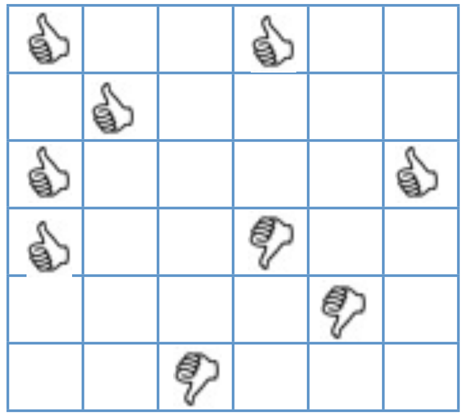
Public, incomplete
IMDB data

Image credit: Arvind Narayanan

Netflix Data Release [Narayanan, Shmatikov 2008]



+

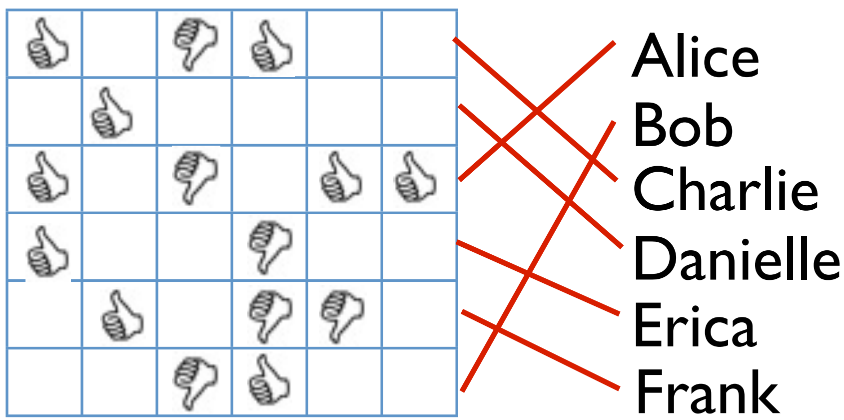


Alice
Bob
Charlie
Danielle
Erica
Frank

Anonymized
NetFlix data

Public, incomplete
IMDB data

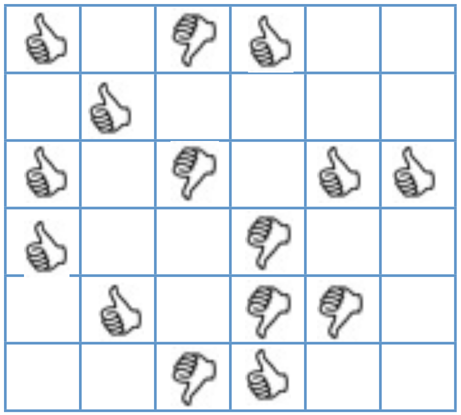
=

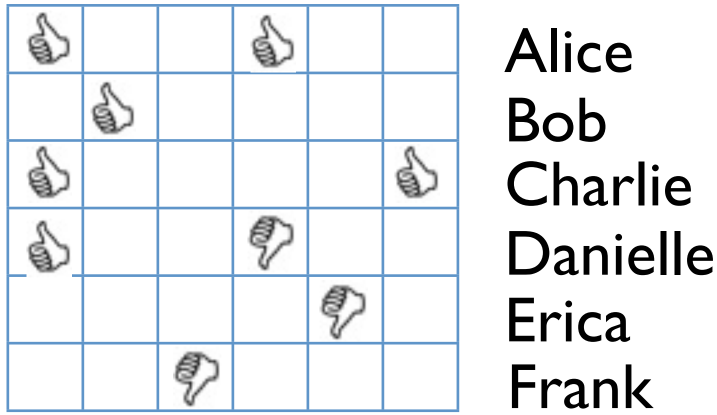


Identified NetFlix Data

Image credit: Arvind Narayanan

Netflix Data Release [Narayanan, Shmatikov 2008]

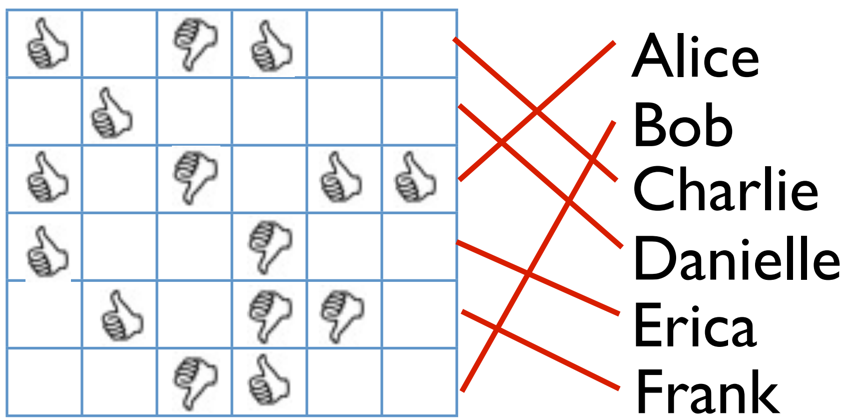




Anonymized
NetFlix data

Public, incomplete
IMDB data

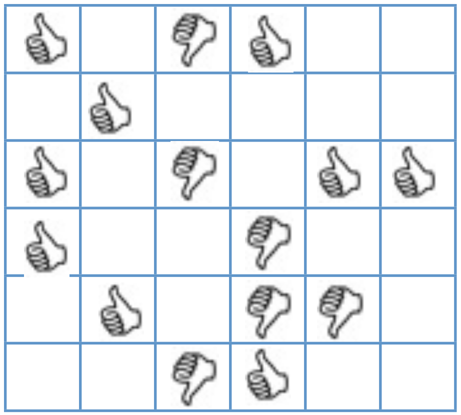
On average, four movies uniquely identify user



Identified NetFlix Data

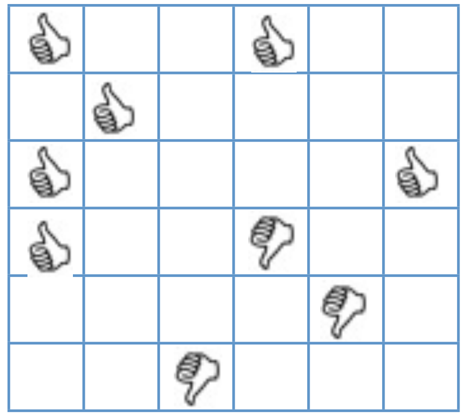
Image credit: Arvind Narayanan

Netflix Data Release [Narayanan, Shmatikov 2008]



Anonymized
NetFlix data

+

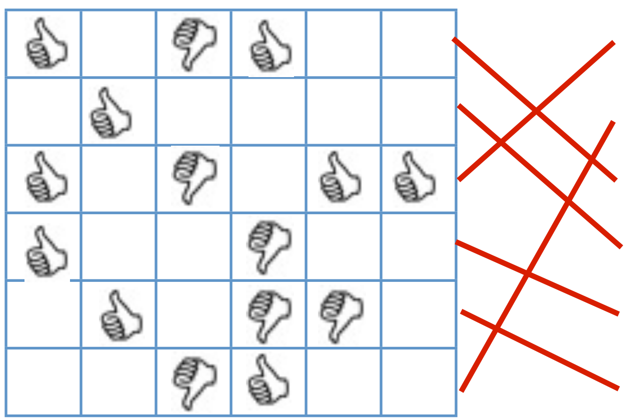


Alice
Bob
Charlie
Danielle
Erica
Frank

Public, incomplete
IMDB data

On average, four movies uniquely identify user

=



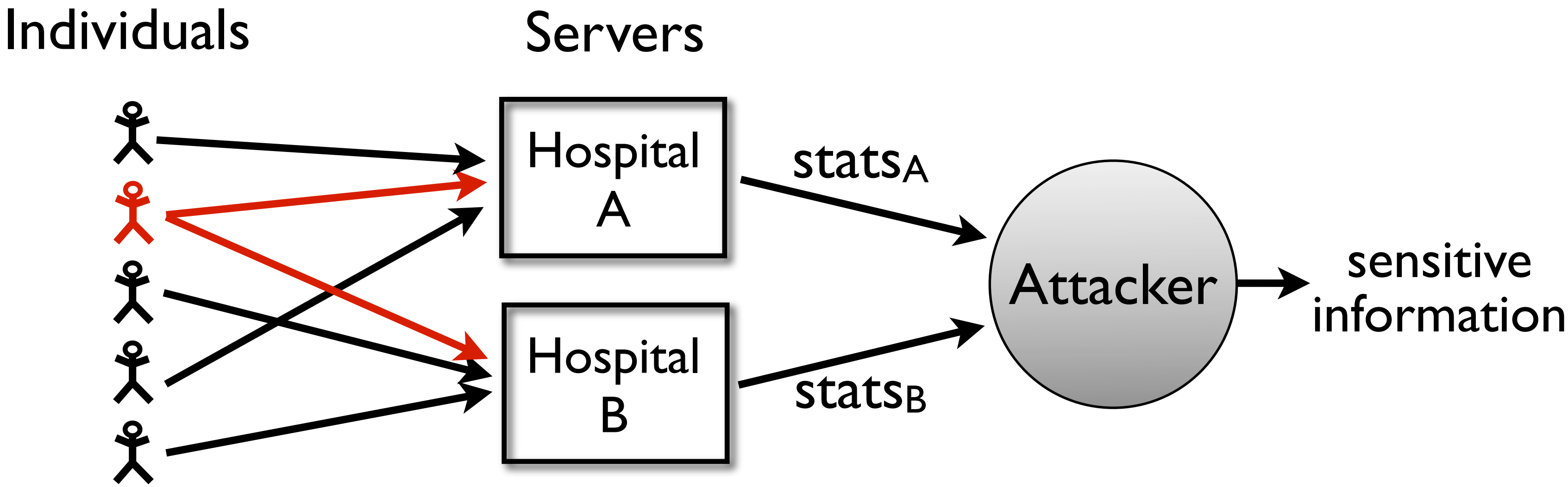
Alice
Bob
Charlie
Danielle
Erica
Frank

Identified NetFlix Data

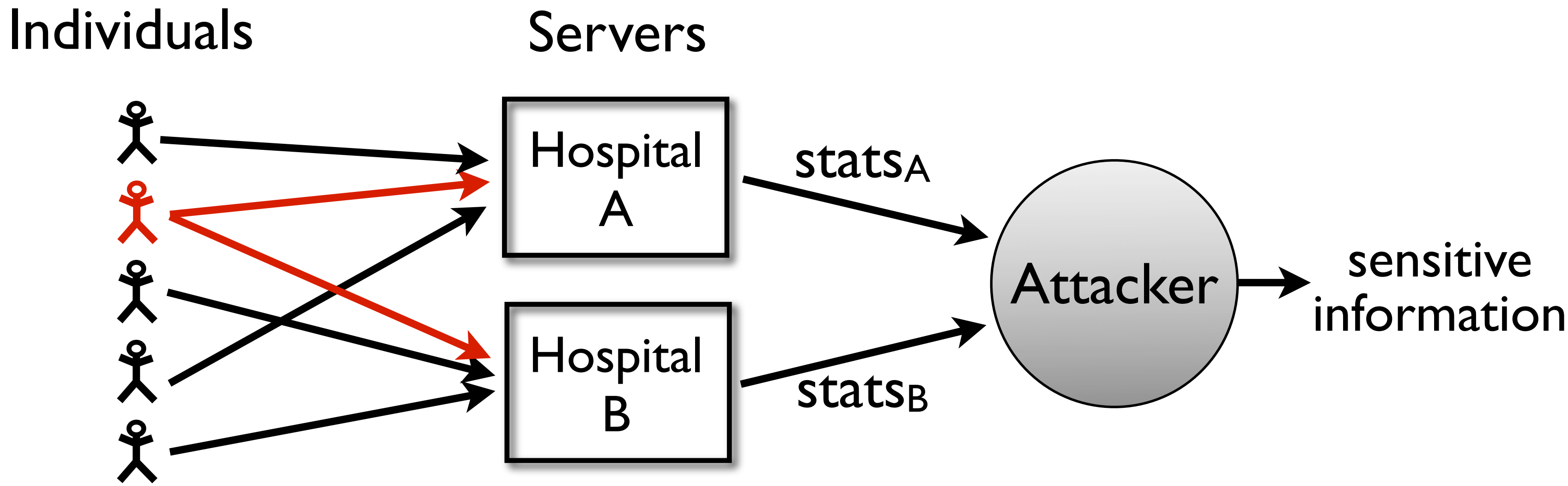
Second round of Netflix competition postponed

Image credit: Arvind Narayanan

“Composition” Attacks [Ganta, Kasiviswanathan, S., *KDD* 2008]

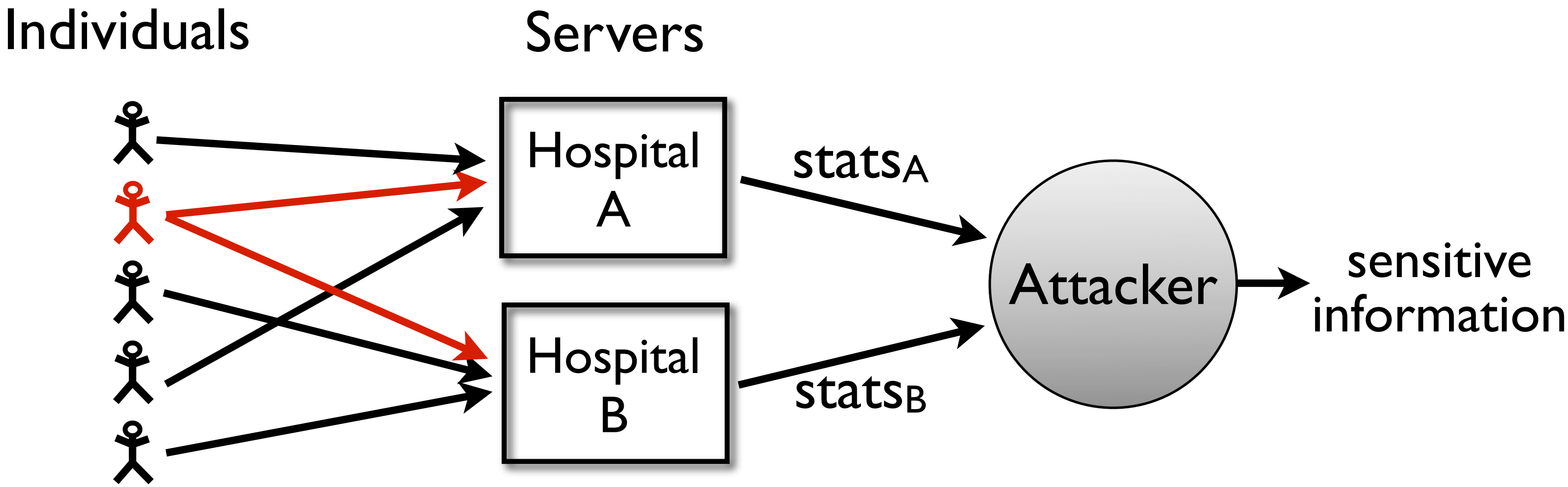


“Composition” Attacks [Ganta, Kasiviswanathan, S., *KDD* 2008]



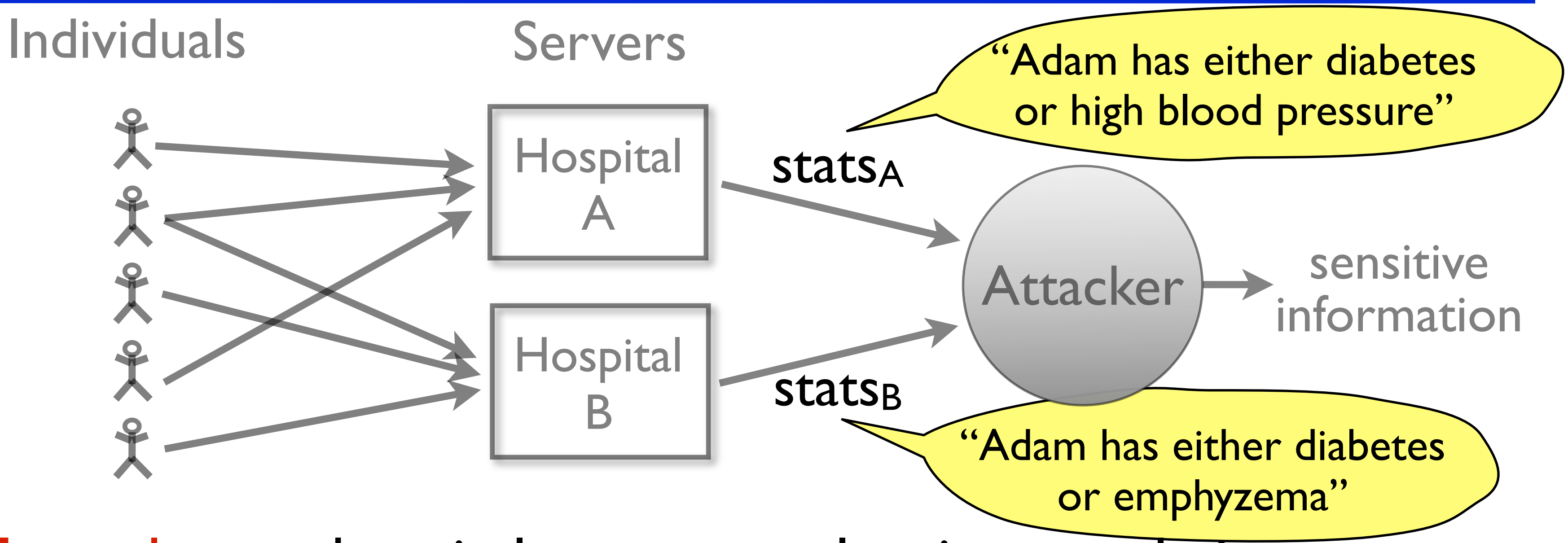
- Example: two hospitals serve overlapping populations
  - What if they independently release “anonymized” statistics?

"Composition" Attacks [Ganta, Kasiviswanathan, S., *KDD* 2008]



- Example: two hospitals serve overlapping populations
  - What if they independently release "anonymized" statistics?
- Composition attack: Combine independent releases
  - Popular anonymization schemes leak lots of information

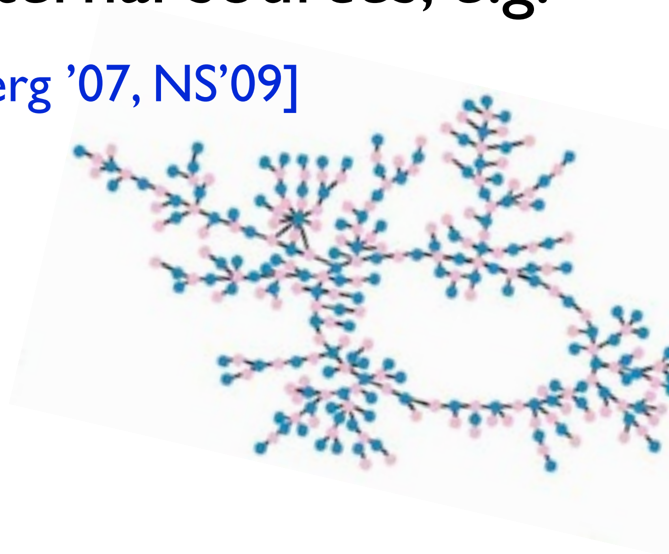
"Composition" Attacks [Ganta, Kasiviswanathan, S., *KDD* 2008]



- Example: two hospitals serve overlapping populations
  - What if they independently release "anonymized" statistics?
- Composition attack: Combine independent releases
  - Popular anonymization schemes leak lots of information

Other attacks

- Reidentifying individuals based on external sources, e.g.
  - ➢ Social networks [Backstrom, Dwork, Kleinberg '07, NS'09]
  - ➢ Computer networks [Coull, Wright, Monrose, Collins, Reiter '07, Ribeiro, Chen, Miklau, Townsley 08]
  - ➢ Genetic data (GWAS) [Homer et al. '08, ...]
  - ➢ Advertising systems [Korolova]



Is the problem granularity?

Is the problem granularity?

- Examples so far: releasing individual information

Is the problem granularity?

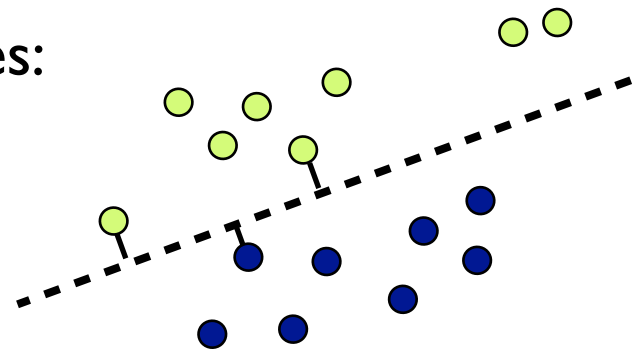
- Examples so far: releasing individual information
- Problems:

Is the problem granularity?

- Examples so far: releasing individual information
- Problems:
  - ➢ Composition
    - Average salary before/after professor resigns

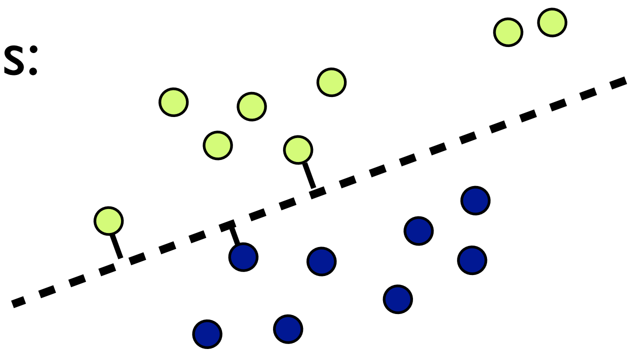
Is the problem granularity?

- Examples so far: releasing individual information
- Problems:
  - ➢ Composition
    - Average salary before/after professor resigns
  - ➢ "Global" result can reveal specific values:
    - "Support Vector Machine" output depends on only a few inputs

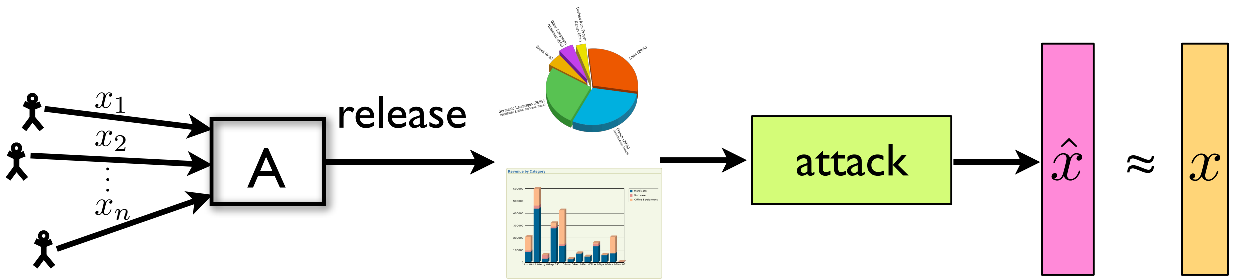


Is the problem granularity?

- Examples so far: releasing individual information
- Problems:
  - Composition
    - Average salary before/after professor resigns
  - "Global" result can reveal specific values:
    - "Support Vector Machine" output depends on only a few inputs
  - Statistics may together encode data
    - Reconstruction attacks: Too many, "too accurate" stats $\Rightarrow$ reconstruct the data
    - Robust even to fairly significant noise



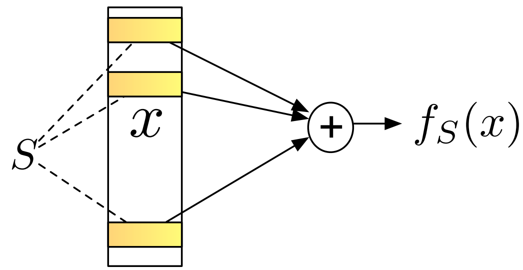
Reconstruction Attacks [DiNi03]



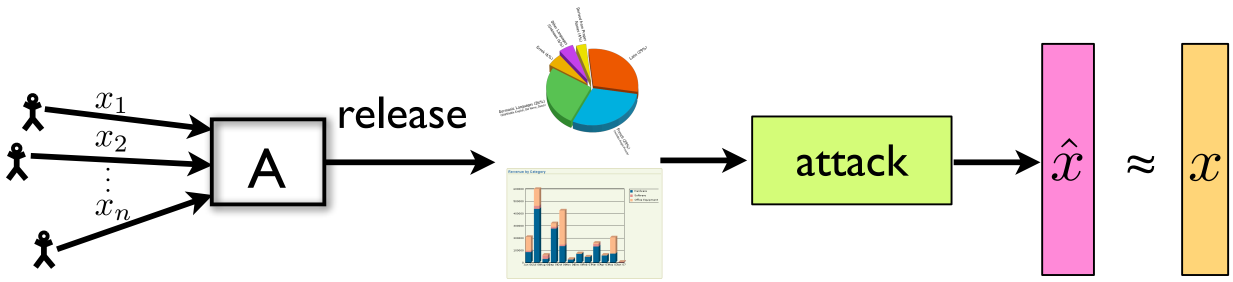
$$\hat{f} \approx f(x)$$

Concrete setting: n users, each with secret $x(i) \in \{0, 1\}$.
Subset query: for $S \subseteq \{1, ...n\}$, let

$$f_S(x) = \frac{1}{n} \sum_{i \in S} x(i) = \frac{1}{n} \langle \chi_S, \vec{x} \rangle$$



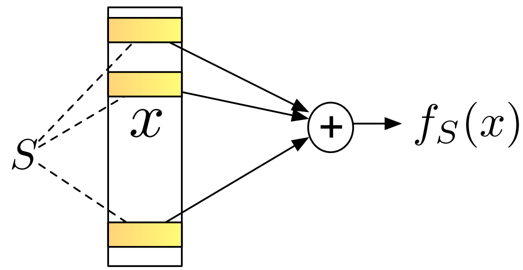
Reconstruction Attacks [DiNi03]



$$\hat{f} \approx f(x)$$

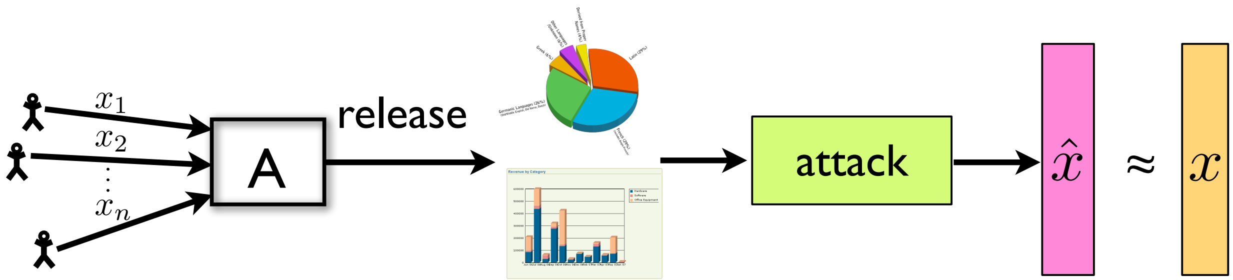
Concrete setting: n users, each with secret $x(i) \in \{0,1\}$.
Subset query: for $S \subseteq \{1, \ldots n\}$, let

$$f_S(x) = \frac{1}{n} \sum_{i \in S} x(i) = \frac{1}{n} \langle \chi_S, \vec{x} \rangle$$



What sets of subset queries $S_1, \ldots, S_m$ allow reconstruction?

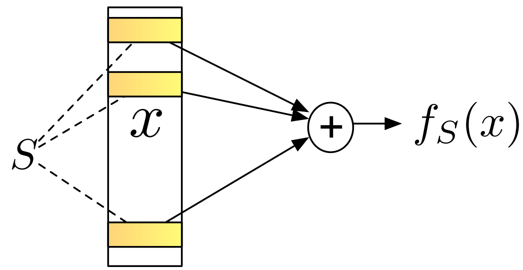
Reconstruction Attacks [DiNi03]



$$\hat{f} \approx f(x)$$

Concrete setting: n users, each with secret $x(i) \in \{0, 1\}$.
Subset query: for $S \subseteq \{1, ...n\}$, let

$$f_S(x) = \frac{1}{n} \sum_{i \in S} x(i) = \frac{1}{n} \langle \chi_S, \vec{x} \rangle$$



What sets of subset queries $S_1, ..., S_m$ allow reconstruction?

- \# queries m
- Error $d_{Hamming}(\hat{x}, x)$, for distortion $\alpha = \max_i |\hat{f}_{S_i} - f_{S_i}(x)|$
- Running time

Can we release all subset queries?

	[DiNi03]
# queries m	2^n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert \hat{f}_{S_i} - f_{S_i}(x) \rvert$	$4\alpha n$
Running time	2^n

Can we release all subset queries?

	[DiNi03]
# queries m	2^n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert \hat{f}_{S_i} - f_{S_i}(x) \rvert$	$4\alpha n$
Running time	2^n

Attack successful for any nontrivial error $\alpha = o(1)$.

Can we release all subset queries?

	[DiNi03]
# queries m	2^n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert \hat{f}_{S_i} - f_{S_i}(x) \rvert$	$4\alpha n$
Running time	2^n

Attack successful for any nontrivial error $\alpha = o(1)$.

Algorithm:

- For $y \in \{0,1\}^n$, write Hamming distance in terms of subset queries:

$$d_{Hamming}(y, x) = n \cdot f_{S_0}(x) \qquad + |S_1| - n \cdot f_{S_1}(x)$$

Can we release all subset queries?

	[DiNi03]
# queries m	2^n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert \hat{f}_{S_i} - f_{S_i}(x)\rvert$	$4\alpha n$
Running time	2^n

Attack successful for any nontrivial error $\alpha = o(1)$.

Algorithm:

- For $y \in \{0,1\}^n$, write Hamming distance in terms of subset queries:

$$\begin{aligned} d_{Hamming}(y,x) &= n \cdot f_{S_0}(x) \qquad + \lvert S_1\rvert - n \cdot f_{S_1}(x) \\ \hat{d}_y &= n \cdot \hat{f}_{S_0} \qquad\quad + \lvert S_1\rvert - n \quad \hat{f}_{S_1} \end{aligned}$$

Can we release all subset queries?

	[DiNi03]
# queries m	2^n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert \hat{f}_{S_i} - f_{S_i}(x) \rvert$	$4\alpha n$
Running time	2^n

Attack successful for any nontrivial error $\alpha = o(1)$.

Algorithm:

- For $y \in \{0,1\}^n$, write Hamming distance in terms of subset queries:

$$\begin{aligned} d_{Hamming}(y,x) &= n \cdot f_{S_0}(x) \qquad &+|S_1| - n \cdot f_{S_1}(x) \\ \hat{d}_y &= n \cdot \hat{f}_{S_0} \qquad &+|S_1| - n \quad \hat{f}_{S_1} \end{aligned}$$

- Output $\hat{x} = \arg\min_{y \in \{0,1\}^n} \hat{d}_y$

A few subset queries? [DiNi03,DMT07,DY08]

	[DiNi03]	[DiNi03,DMT07,DY08]
# queries m	2^n	n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert\hat{f}_{S_i} - f_{S_i}(x)\rvert$	$4\alpha n$	$2(\alpha\sqrt{n})n$
Running time	2^n	$O(n \log n)$

A few subset queries? [DiNi03,DMT07,DY08]

	[DiNi03]	[DiNi03,DMT07,DY08]
# queries m	2^n	n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert \hat{f}_{S_i} - f_{S_i}(x) \rvert$	$4\alpha n$	$2(\alpha\sqrt{n})n$
Running time	2^n	$O(n \log n)$

Attack successful for error $\alpha = o(1/\sqrt{n})$.

A few subset queries? [DiNi03,DMT07,DY08]

	[DiNi03]	[DiNi03,DMT07,DY08]
# queries m	2^n	n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert \hat{f}_{S_i} - f_{S_i}(x) \rvert$	$4\alpha n$	$2(\alpha\sqrt{n})n$
Running time	2^n	$O(n \log n)$

A few subset queries? [DiNi03,DMT07,DY08]

	[DiNi03]	[DiNi03,DMT07,DY08]
# queries m	2^n	n
Error $d_{Hamming}(\hat{x}, x)$ $\alpha = \max_i \lvert\hat{f}_{S_i} - f_{S_i}(x)\rvert$	$4\alpha n$	$2(\alpha\sqrt{n})n$
Running time	2^n	$O(n \log n)$

Algorithm:

- Queries come from the rows of ± 1 Hadamard matrix:
  - $H_1 = (1) \qquad H_n = \begin{pmatrix} H_{n/2} & H_{n/2} \\ H_{n/2} & -H_{n/2} \end{pmatrix}$
  - H_n has all eigenvalues $\pm\sqrt{n}$.
- Using n subset queries (one per row), can derive
  $z = \frac{1}{n} H_n x + e$ where $\|e\|_\infty \leq 2\alpha$
- Compute $\hat{x}' = (n \cdot H_n^{-1})z = x + e'$ where $\|e'\|_2 \leq 2\alpha n$
- **Round** to $\{0,1\}^n$ to get $\hat{x}$

Beyond Subset Queries

Beyond Subset Queries

- These attacks can be extended
  - Handle some very distorted queries
  - Exploit sparsity of secret vector

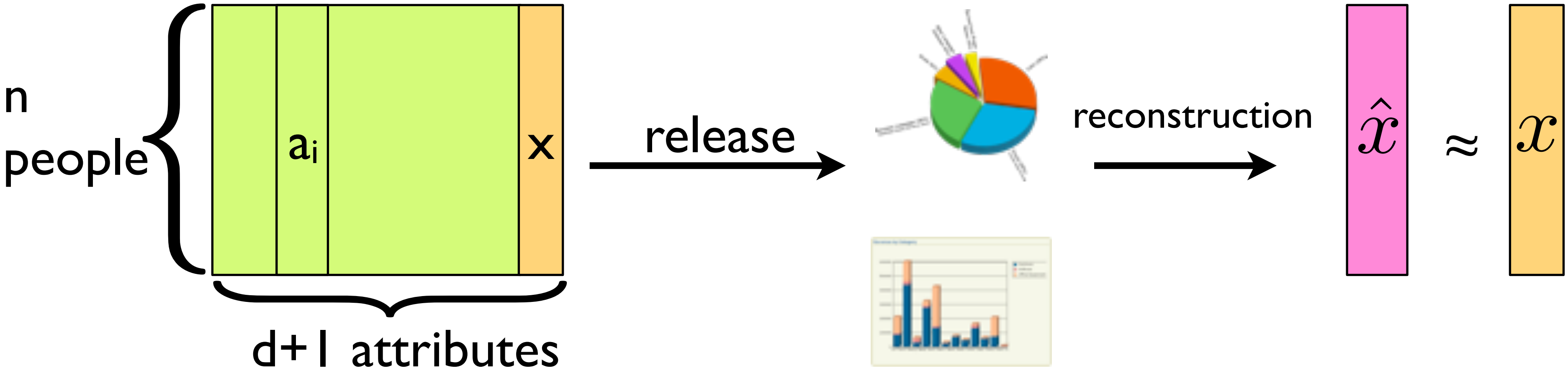
Beyond Subset Queries

- These attacks can be extended
  - ➢ Handle some very distorted queries
  - ➢ Exploit sparsity of secret vector
- So far: unnatural queries
  - ➢ Algebraically defined or uniformly random
  - ➢ Require "naming rows"

Beyond Subset Queries

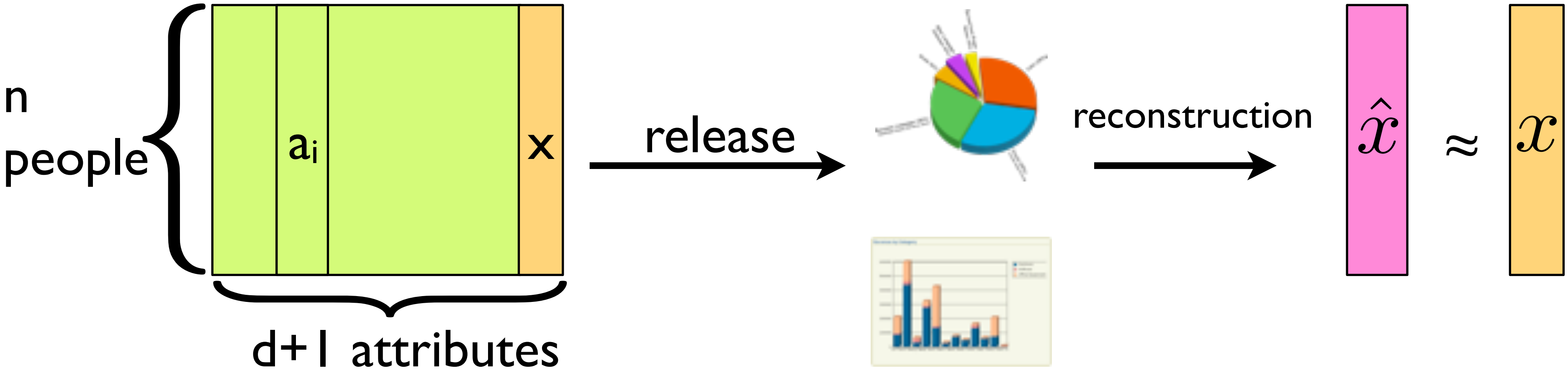
- These attacks can be extended
  - ➢ Handle some very distorted queries
  - ➢ Exploit sparsity of secret vector
- So far: unnatural queries
  - ➢ Algebraically defined or uniformly random
  - ➢ Require "naming rows"
- Natural, symmetric queries? Yes!
  - ➢ [KRSU'10] **marginal tables**
    - Each person's data is a row in a table
    - **k-way marginal**: distribution of some k attributes
  - ➢ [KRS'12] regression analysis, decision tree classifiers, ...

Reconstruction from Marginals [KRSU'10]



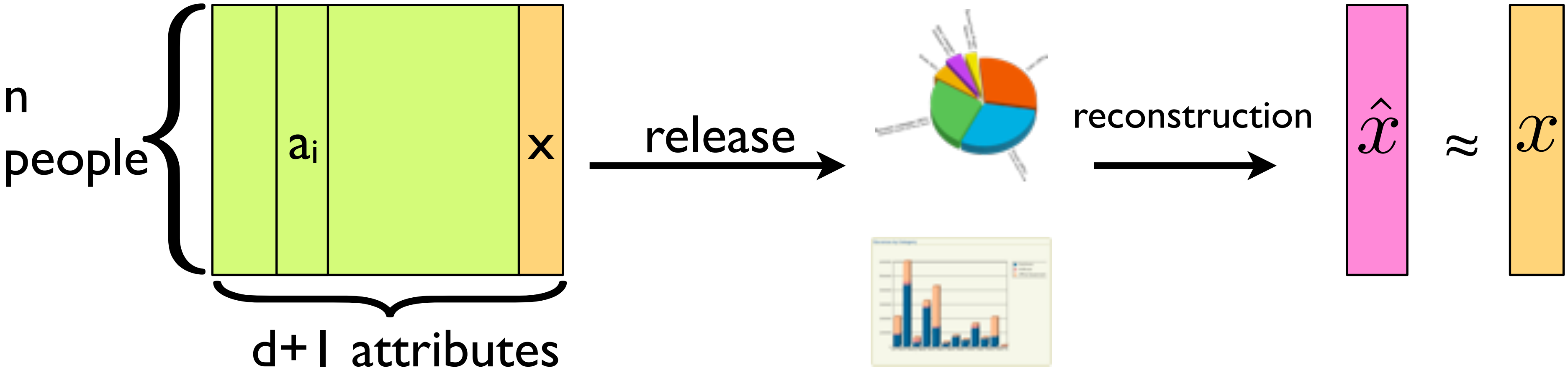
Reconstruction from Marginals [KRSU'10]

- Data set: d "public" attributes per person, 1 "sensitive"



Reconstruction from Marginals [KRSU'10]

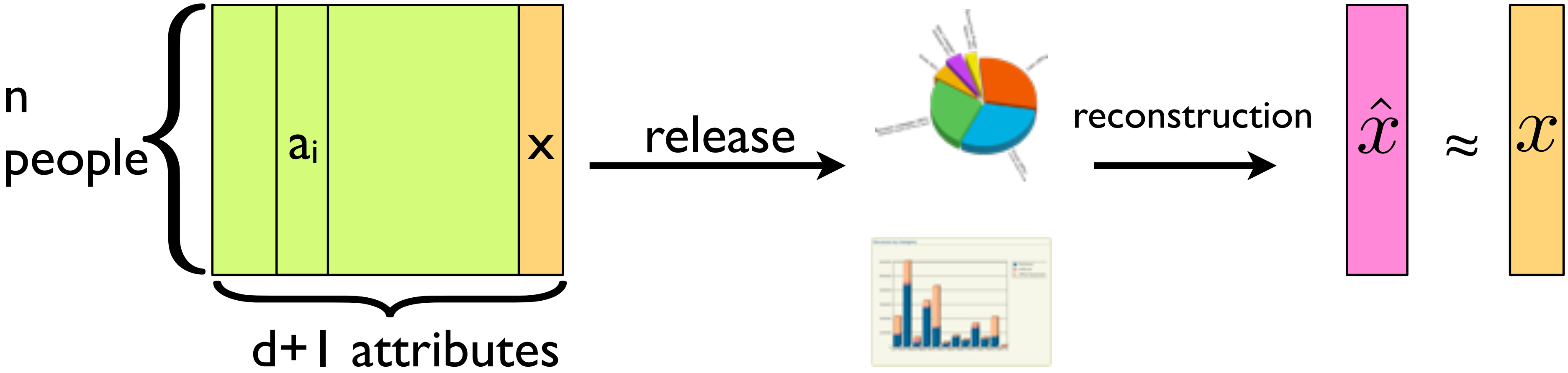
- Data set: d "public" attributes per person, 1 "sensitive"



- Suppose release allows learning 2-way marginals
  - ➢ 2-way marginals are subset queries!
  - ➢ If a_i are uniformly random and d > n, then $d_{Ham}(\hat{x}, x) = o(n)$

Reconstruction from Marginals [KRSU'10]

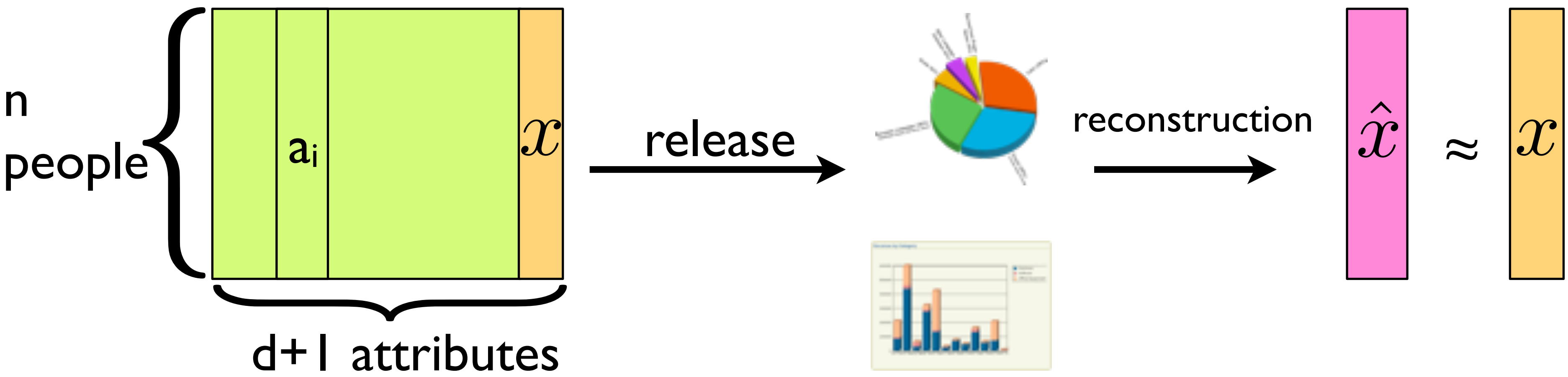
- Data set: d "public" attributes per person, 1 "sensitive"



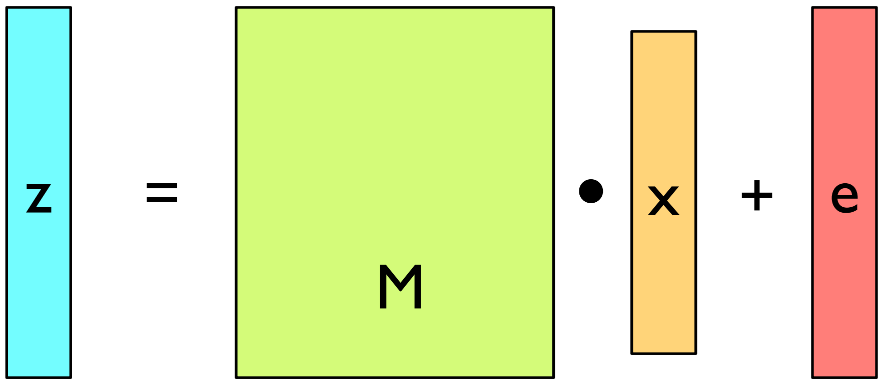
- Suppose release allows learning 2-way marginals
  - 2-way marginals are subset queries!
  - If a_i are uniformly random and d > n, then $d_{Ham}(\hat{x}, x) = o(n)$
- **Theorem:** With k-way marginals, $d \gg n^{\frac{1}{k-1}}$ suffices

Reconstruction from Marginals [KRSU'10]

- Data set: d "public" attributes per person, 1 "sensitive"



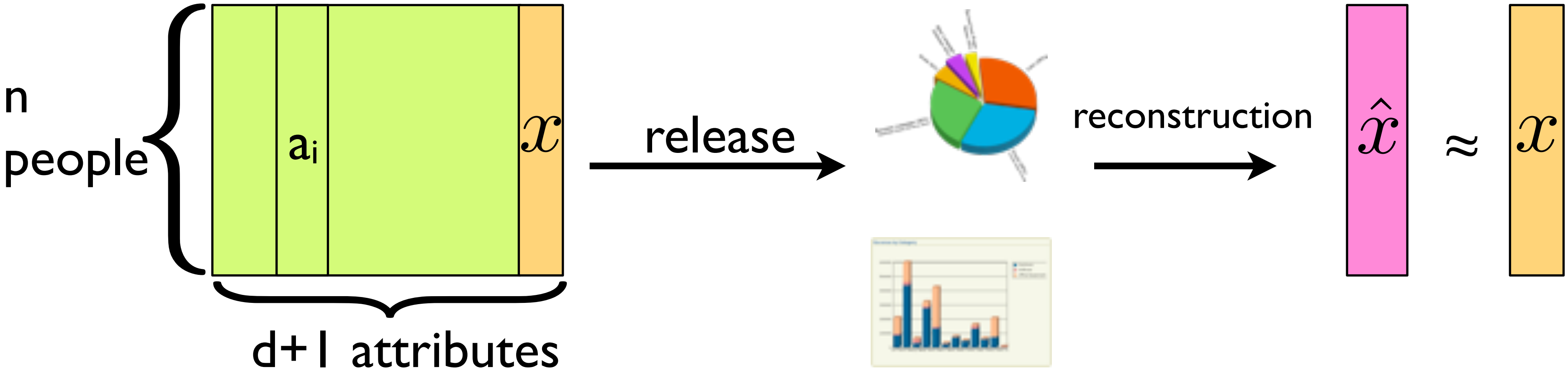
- Idea: view statistics as noisy linear encoding Mx + e



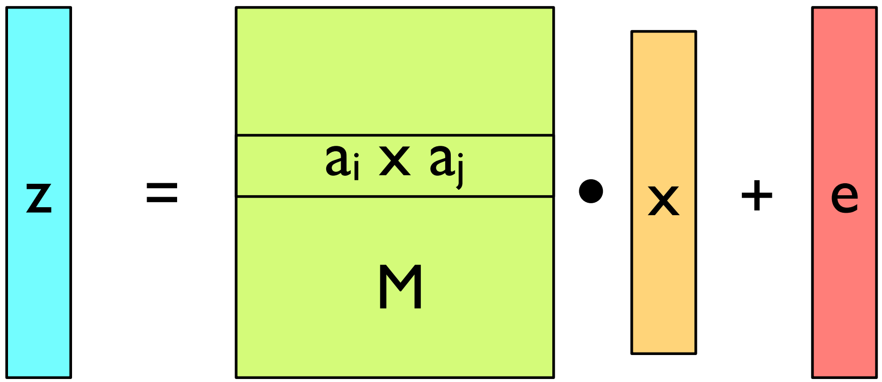
- Signal processing: Reconstruction uses geometry of matrix M

Reconstruction from Marginals [KRSU'10]

- Data set: d "public" attributes per person, 1 "sensitive"



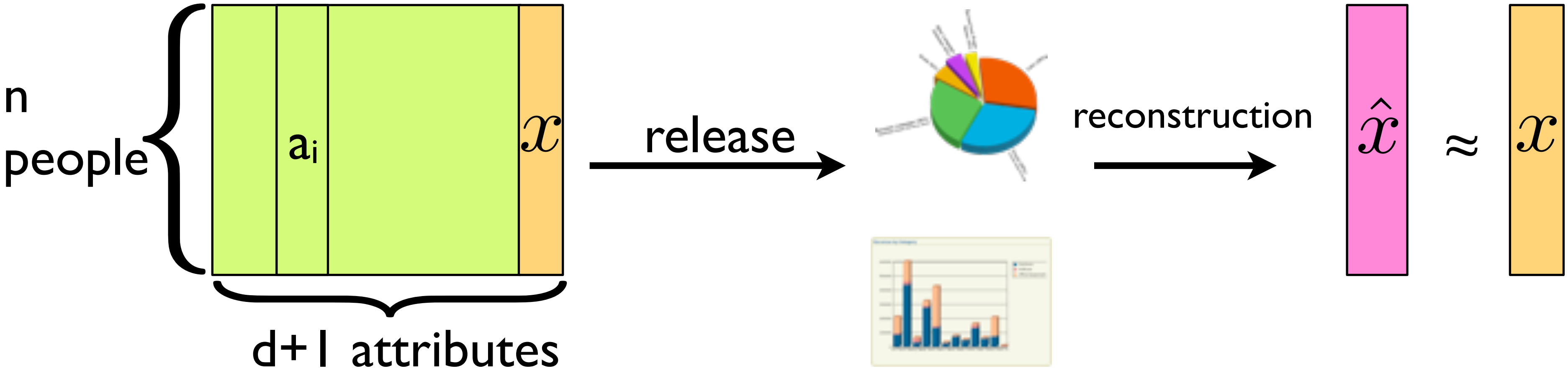
- Idea: view statistics as noisy linear encoding Mx + e



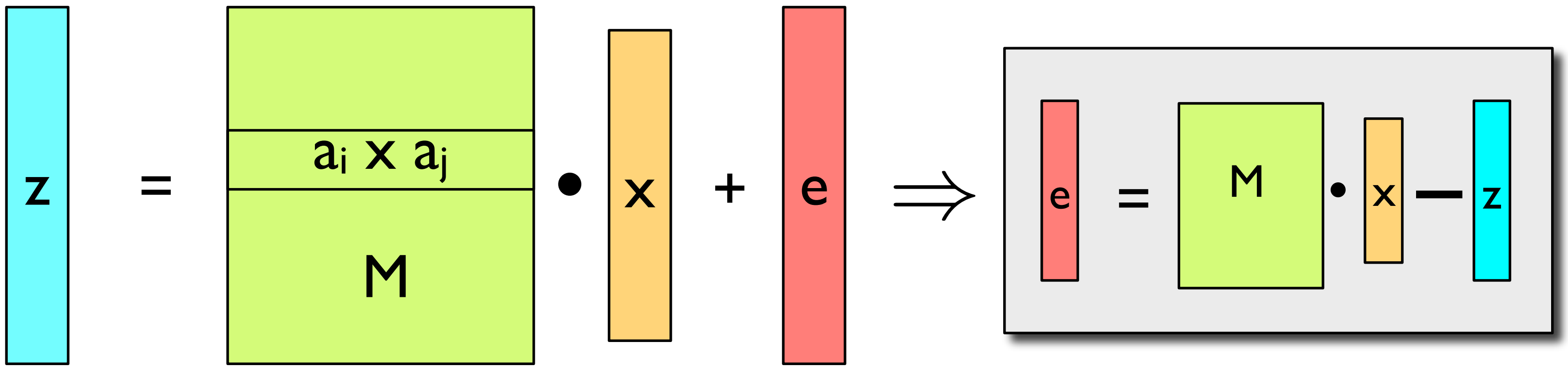
- Signal processing: Reconstruction uses geometry of matrix M

Reconstruction from Marginals [KRSU'10]

- Data set: d "public" attributes per person, 1 "sensitive"

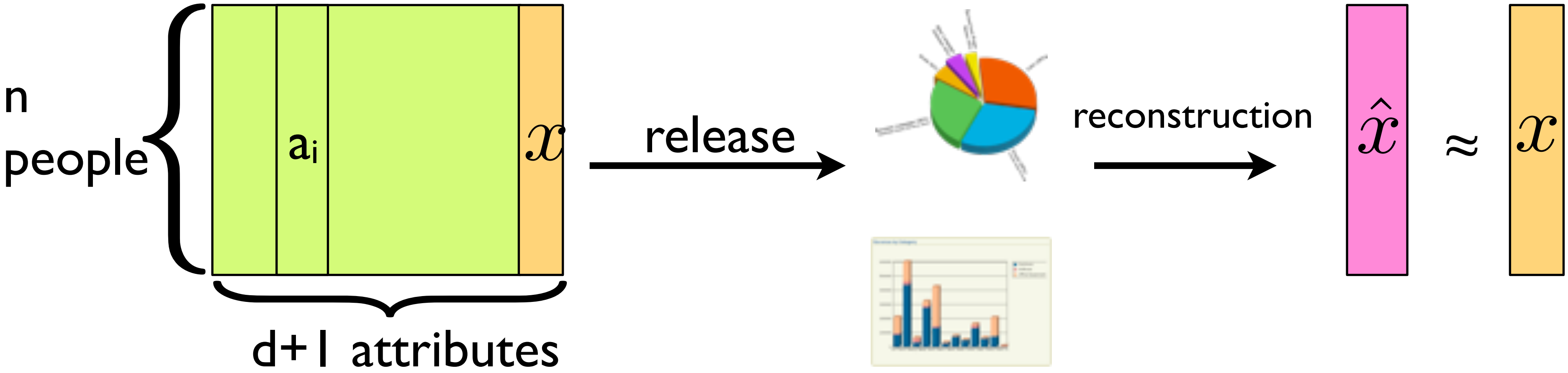


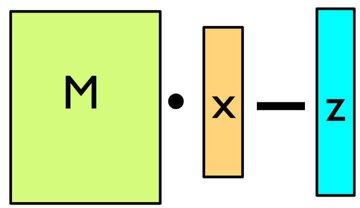
- Idea: view statistics as noisy linear encoding Mx + e



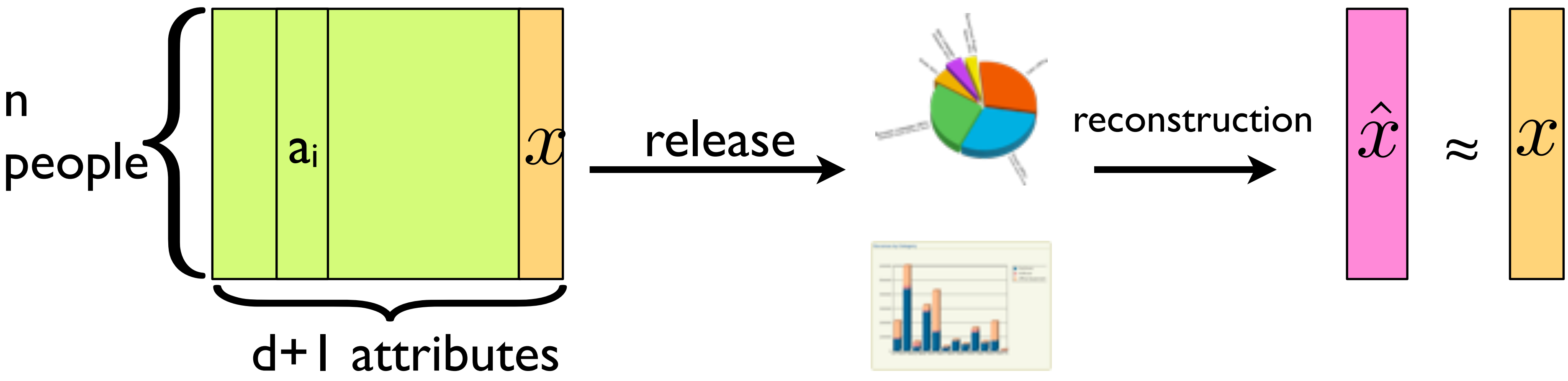
- Signal processing: Reconstruction uses geometry of matrix M

Reconstruction from Marginals

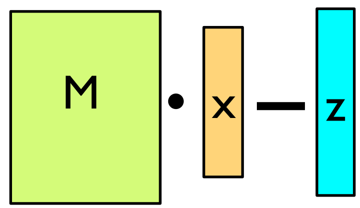




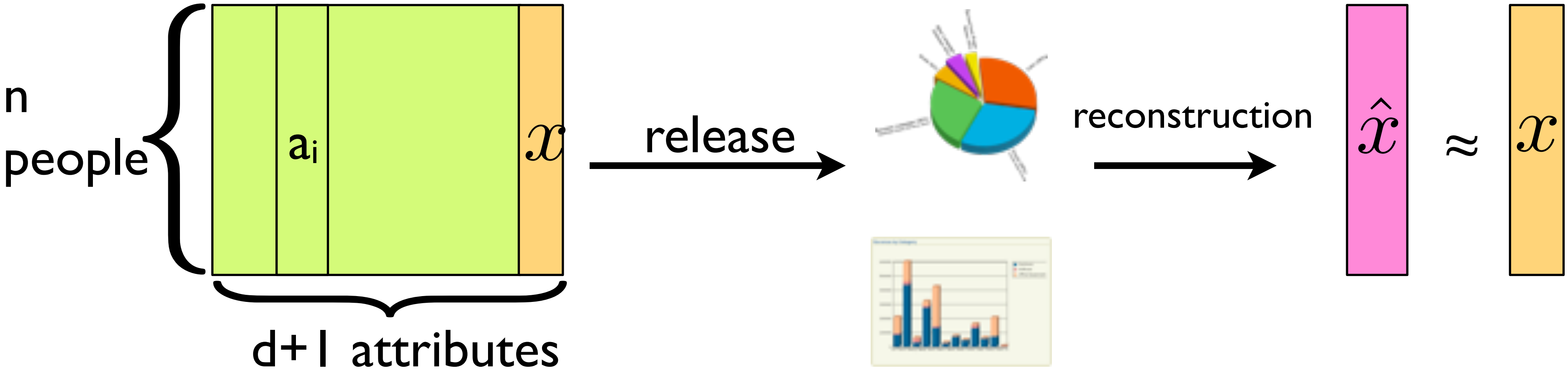
Reconstruction from Marginals



- Minimize estimated error in ℓ_p
  - p=2: least singular values
  - p=1: "Euclidean section"



Reconstruction from Marginals



- Minimize estimated error in ℓ_p
  - ➢ p=2: least singular values
  - ➢ p=1: "Euclidean section"

$$\hat{x} = \text{argmin}_x \left\| M \cdot x - z \right\|_p$$

Attacks on data privacy

Attacks on data privacy

- So far:
  - Many ad hoc examples
    - E.g., Netflix, ...
  - Some general principles
    - E.g., Composition
  - Sophisticated reconstruction attacks
    - Draws on theory of coding and signal processing
  - Lower bounds for various classes of release mechanisms
    - Sometimes based on crypto objects [DNRRV, UV]

Attacks on data privacy

- So far:
  - ➢ Many ad hoc examples
    - E.g., Netflix, ...
  - ➢ Some general principles
    - E.g., Composition
  - ➢ Sophisticated reconstruction attacks
    - Draws on theory of coding and signal processing
  - ➢ Lower bounds for various classes of release mechanisms
    - Sometimes based on crypto objects [DNRRV, UV]
- Still missing:
  - ➢ Systematic understanding
  - ➢ Suite of standard attack techniques (à la differential/linear cryptanalysis?)

Lessons

Lessons

- Even if releasing only "aggregate" statistics, we can't release everything
  - ➢ We release some information at the expense of other kinds
  - ➢ Inherent tradeoff very different from "crypto as usual"

Lessons

- Even if releasing only "aggregate" statistics, we can't release everything
  - ➢ We release some information at the expense of other kinds
  - ➢ Inherent tradeoff very different from "crypto as usual"

- Even a single "aggregate" statistic can be hard to reason about

Lessons

- Even if releasing only "aggregate" statistics, we can't release everything
  - ➢ We release some information at the expense of other kinds
  - ➢ Inherent tradeoff very different from "crypto as usual"

- Even a single "aggregate" statistic can be hard to reason about

Lessons

- Even if releasing only "aggregate" statistics, we can't release everything
  - ➢ We release some information at the expense of other kinds
  - ➢ Inherent tradeoff very different from "crypto as usual"

- Even a single "aggregate" statistic can be hard to reason about

- What does "aggregate" mean?

This talk

- **Act I: Attacks**
  - ➢ (Why is privacy hard?)
  - ➢ Reconstruction attacks
- **Act II: Definitions**
  - ➢ One approach: "differential" privacy
  - ➢ Variations on the theme
- **Act III: Algorithms**
  - ➢ Basic techniques: noise addition, exponential sampling
  - ➢ Answering many queries
  - ➢ Exploiting "local" sensitivity

This talk

- **Act I: Attacks**
  - ➢ (Why is privacy hard?)
  - ➢ Reconstruction attacks
- **Act II: Definitions**
  - ➢ One approach: "differential" privacy
  - ➢ Variations on the theme
- **Act III: Algorithms**
  - ➢ Basic techniques: noise addition, exponential sampling
  - ➢ Answering many queries
  - ➢ Exploiting "local" sensitivity

- "Aggregate" ≈ stability to small changes in input
- Handles arbitrary external information
- Burgeoning field of research

This talk

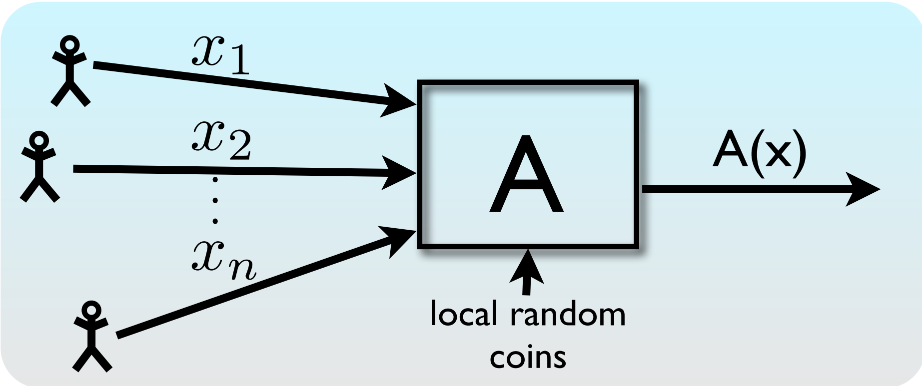
- **Act I: Attacks**
  - ➢ (Why is privacy hard?)
  - ➢ Reconstruction attacks
- **Act II: Definitions**
  - ➢ One approach: "differential" privacy
  - ➢ Variations on the theme
- **Act III: Algorithms**
  - ➢ Basic techniques: noise addition, exponential sampling
  - ➢ Answering many queries
  - ➢ Exploiting "local" sensitivity

- "Aggregate" ≈ stability to small changes in input
- Handles arbitrary external information
- Burgeoning field of research

Differential Privacy [DMNS2006, Dw2006]

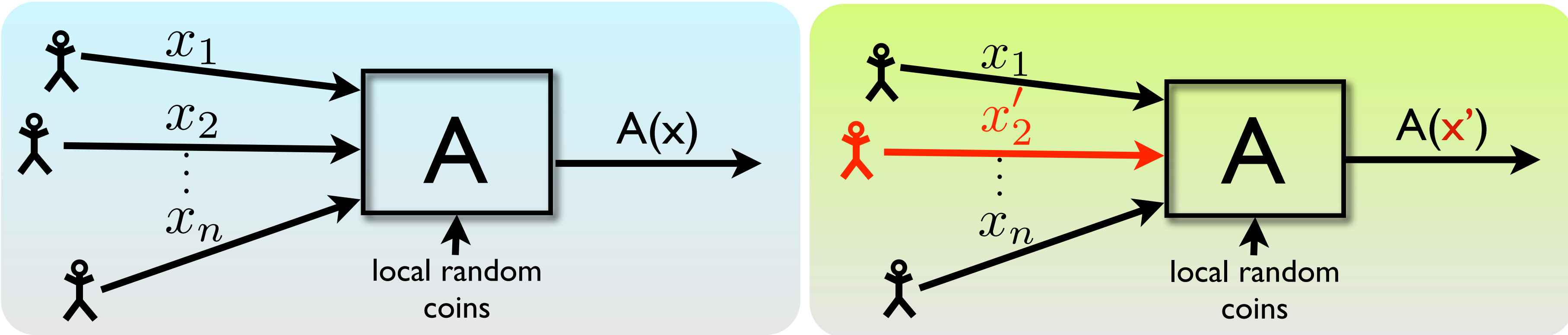
- Intuition:
  - Changes to my data not noticeable by users
  - Output is "independent" of my data

Differential Privacy [DMNS2006, Dw2006]



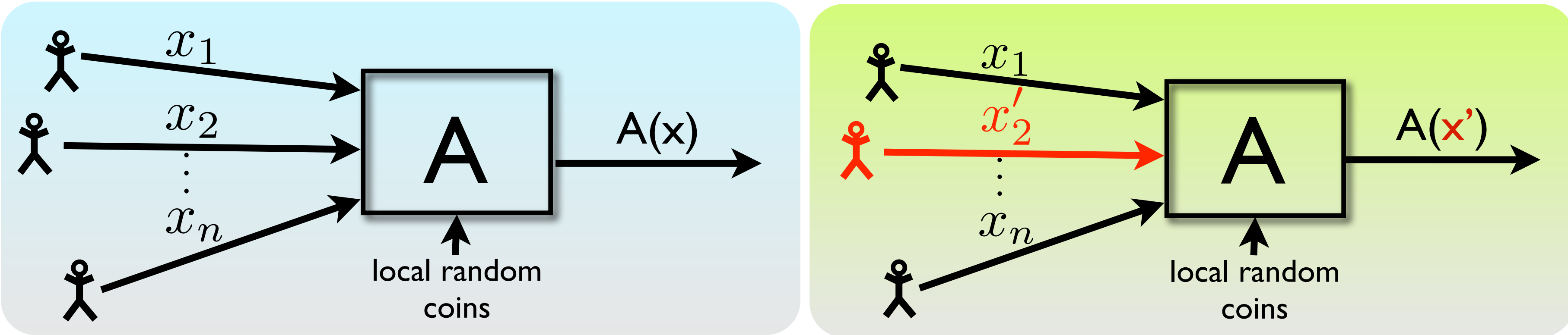
- Data set $\mathbf{x} = (x_1, ..., x_n) \in D^n$
  - ➢ Domain D can be numbers, categories, tax forms
  - ➢ Think of x as **fixed** (not random)
- A = **randomized** procedure
  - ➢ A(x) is a random variable
  - ➢ Randomness might come from adding noise, resampling, etc.

Differential Privacy [DMNS2006, Dw2006]



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

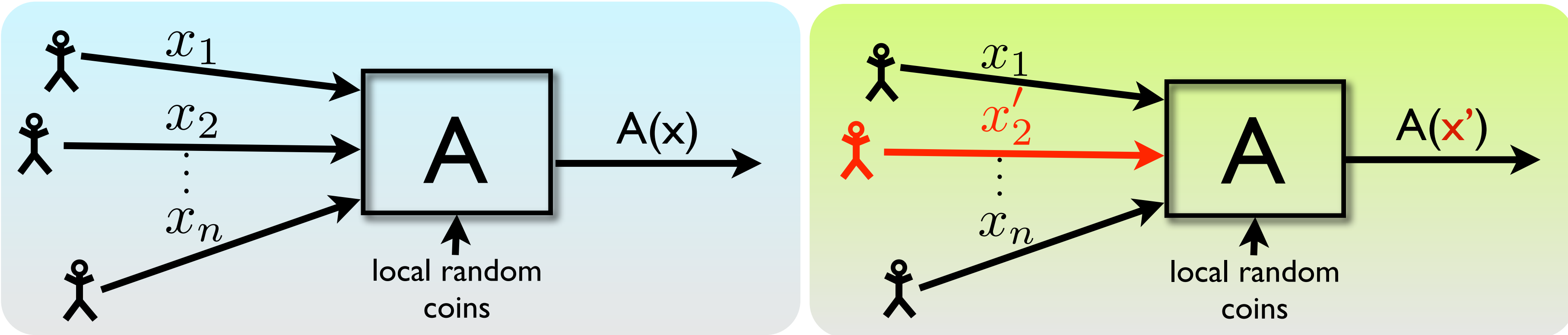
Differential Privacy [DMNS2006, Dw2006]



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

Neighboring databases induce **close** distributions on outputs

Differential Privacy [DMNS2006, Dw2006]



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

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

Differential Privacy [DMNS2006, Dw2006]

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

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

Differential Privacy [DMNS2006, Dw2006]

- This is a condition on the **algorithm** A
  - ➢ Saying a particular output is private makes no sense

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

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

Differential Privacy [DMNS2006, Dw2006]

- This is a condition on the **algorithm** A
  - ➢ Saying a particular output is private makes no sense
- Choice of distance measure matters

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

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

Differential Privacy [DMNS2006, Dw2006]

- This is a condition on the **algorithm** A
  - ➢ Saying a particular output is private makes no sense
- Choice of distance measure matters
- What is ε?
  - ➢ Measure of information leakage
  - ➢ Not too small (think $\frac{1}{10}$, not $\frac{1}{2^{50}}$)

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

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

Differential Privacy [DMNS2006, Dw2006]

- This is a condition on the **algorithm** A
  - ➢ Saying a particular output is private makes no sense
- Choice of distance measure matters
- What is ε?
  - ➢ Measure of information leakage
  - ➢ Not too small (think $\frac{1}{10}$, not $\frac{1}{2^{50}}$)

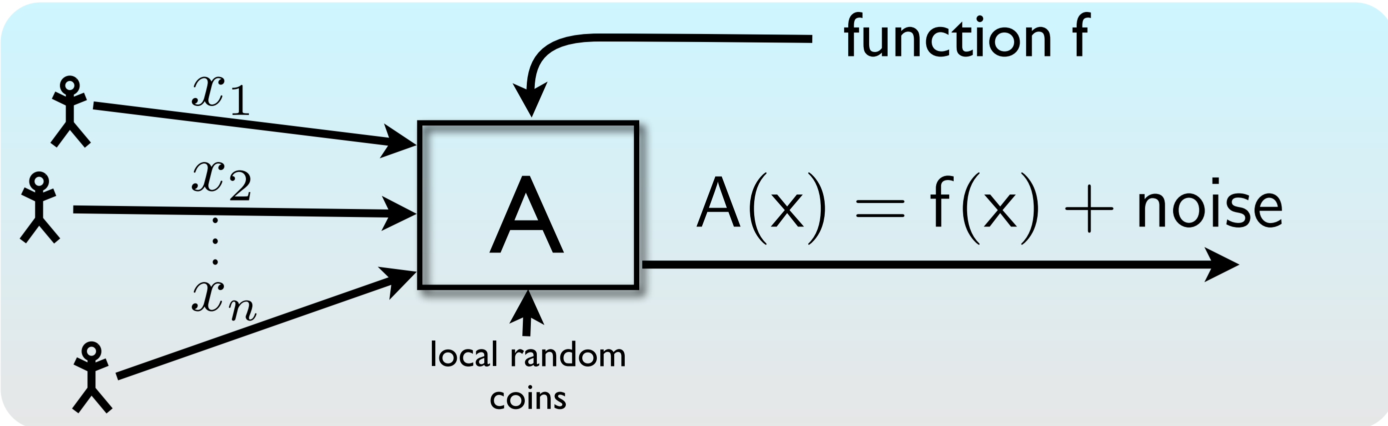
more on these later

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

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

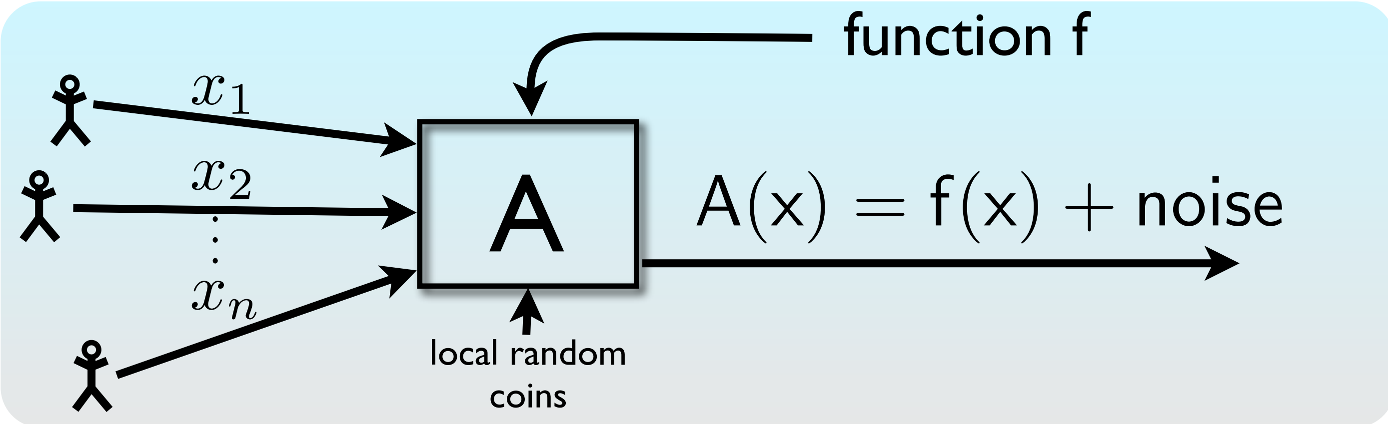
Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



$$f(x) \in \mathbb{R}^p$$

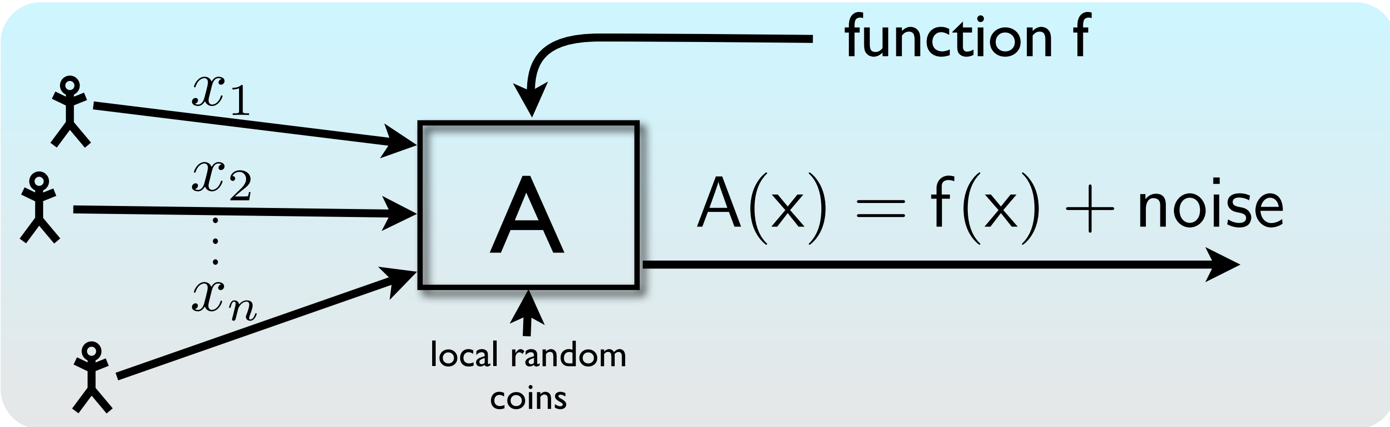
$$x_i \in \{0,1\}, \; f(x) = \frac{1}{n}\sum x_i$$

Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



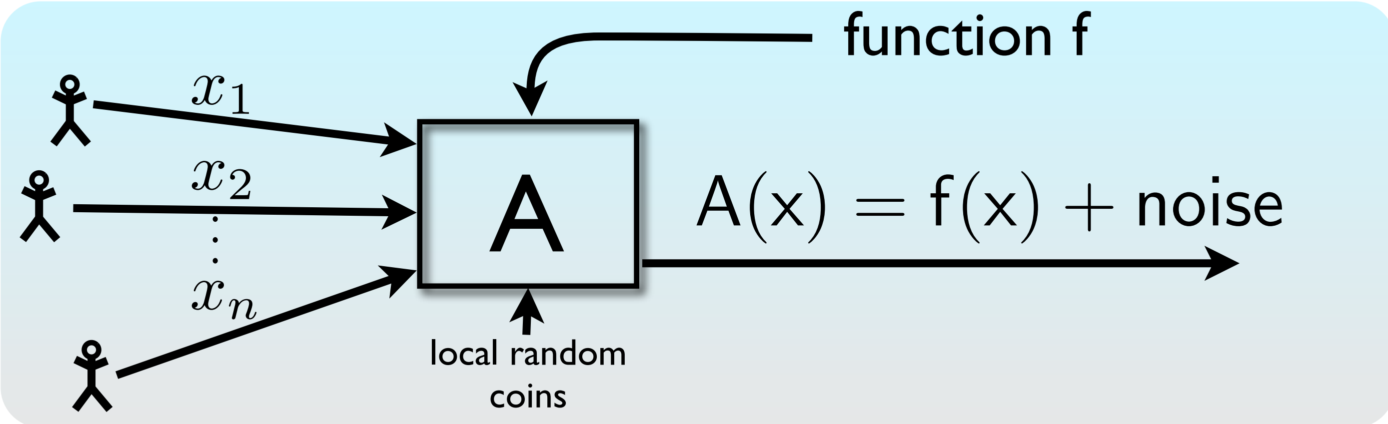
- Say we want to release a summary $f(x) \in \mathbb{R}^p$
  - e.g., proportion of diabetics: $x_i \in \{0, 1\}, \; f(x) = \frac{1}{n} \sum x_i$

Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



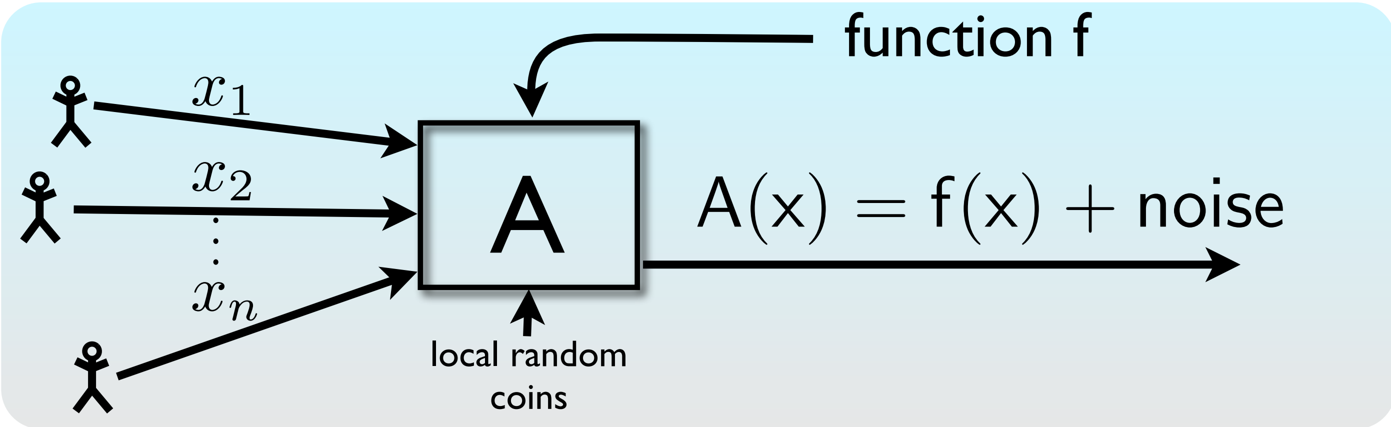
- Say we want to release a summary $f(x) \in \mathbb{R}^p$
  - e.g., proportion of diabetics: $x_i \in \{0,1\}$, $f(x) = \frac{1}{n} \sum x_i$
- Simple approach: add noise to f(x)
  - How much noise is needed?

Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]

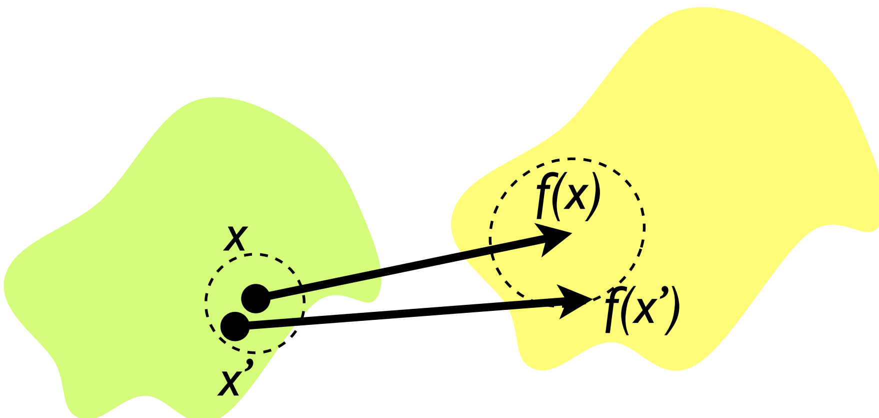


- Say we want to release a summary $\mathsf{f(x)} \in \mathbb{R}^{\mathsf{p}}$
  - ➢ e.g., proportion of diabetics: $\mathsf{x_i} \in \{0,1\},\ \mathsf{f(x)} = \frac{1}{\mathsf{n}} \sum \mathsf{x_i}$
- Simple approach: add noise to f(x)
  - ➢ How much noise is needed?
- Intuition: f(x) can be released accurately when f is insensitive to individual entries $x_1, x_2, \ldots, x_n$

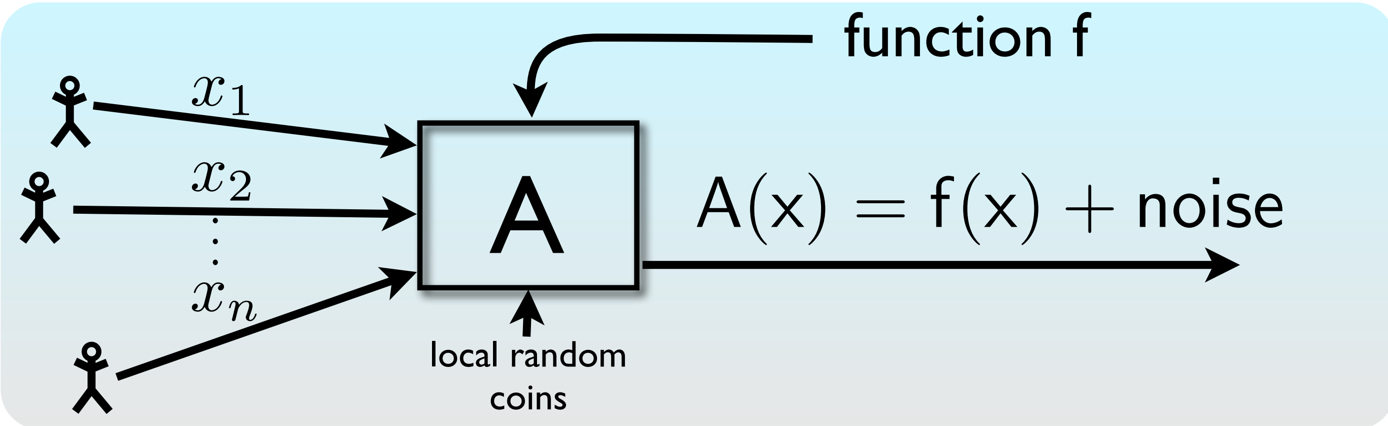
Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



- **Global Sensitivity:** $GS_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
  - Example: $GS_{\text{proportion}} = \frac{1}{n}$



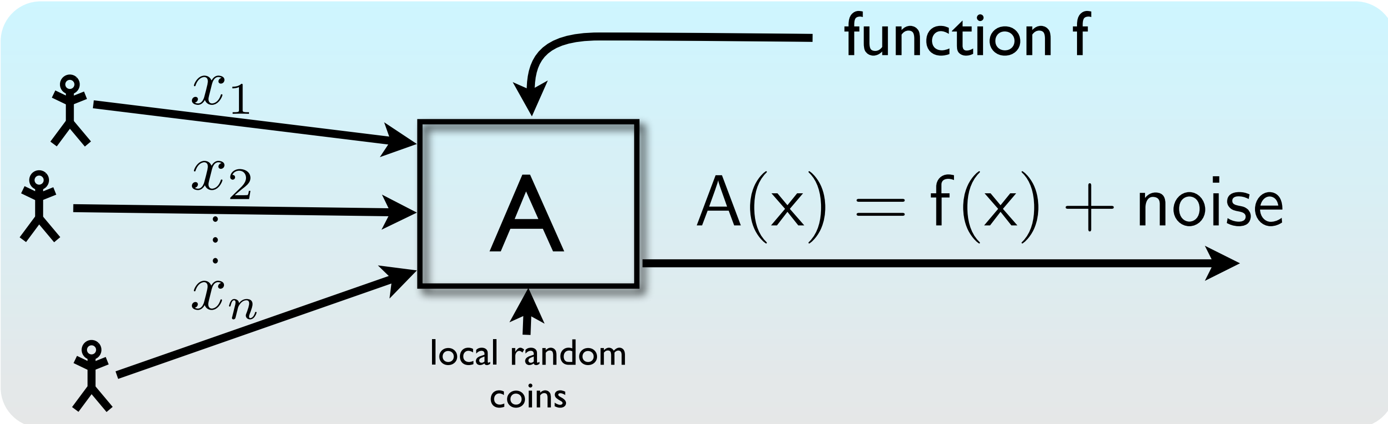
Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



$$\mathsf{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$$

$$\mathsf{GS}_{\text{proportion}} = \frac{1}{\mathsf{n}}$$

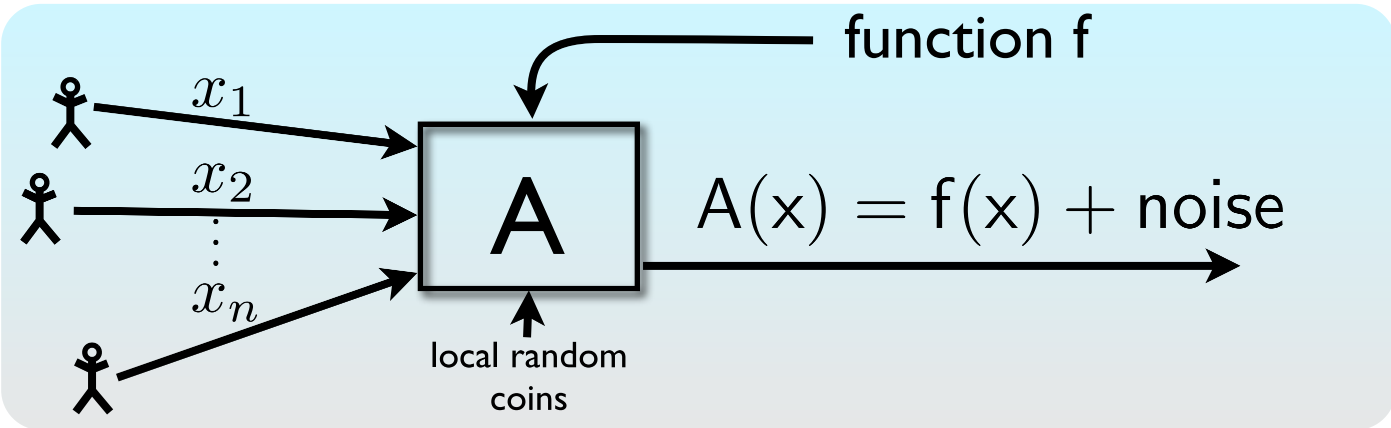
Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



- **Global Sensitivity:** $\mathsf{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
   - ➢ Example: $\mathsf{GS}_{\text{proportion}} = \frac{1}{n}$

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

Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]

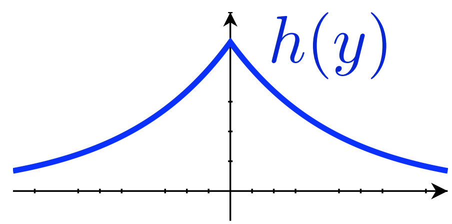


- **Global Sensitivity:** $\mathsf{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
  - Example: $\mathsf{GS}_{\text{proportion}} = \frac{1}{n}$

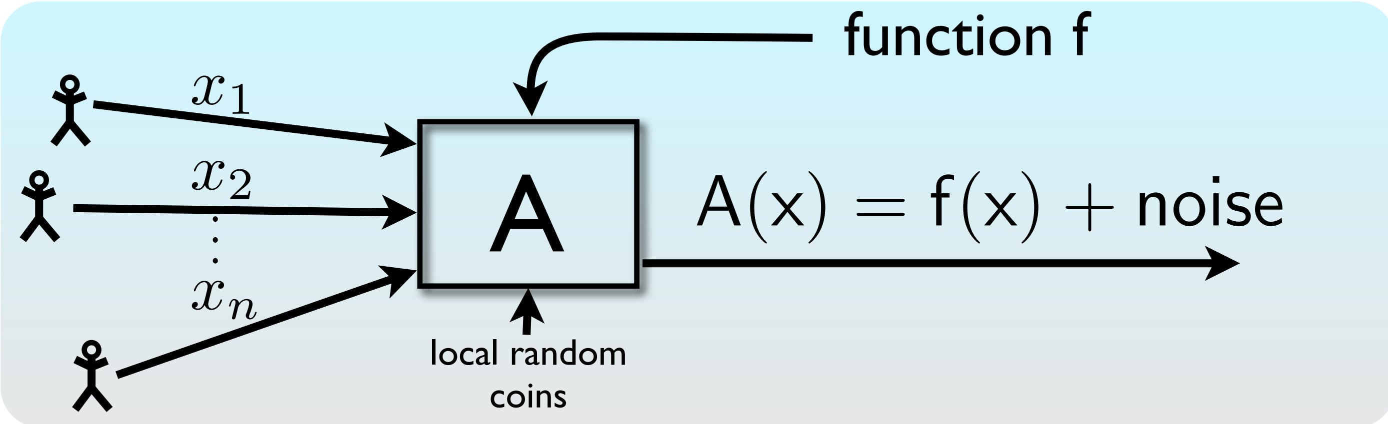
Theorem: If $\mathsf{A}(\mathsf{x}) = f(\mathsf{x}) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\epsilon}\right)$, then A is ϵ-differentially private.

- Laplace distribution $\mathsf{Lap}(\lambda)$ has density

$$h(y) \propto e^{-|y|/\lambda}$$



Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]

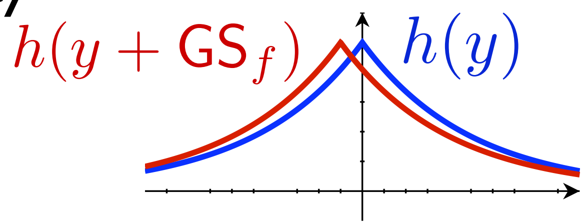


- **Global Sensitivity:** $\mathsf{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
  - Example: $\mathsf{GS}_{\text{proportion}} = \frac{1}{n}$

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

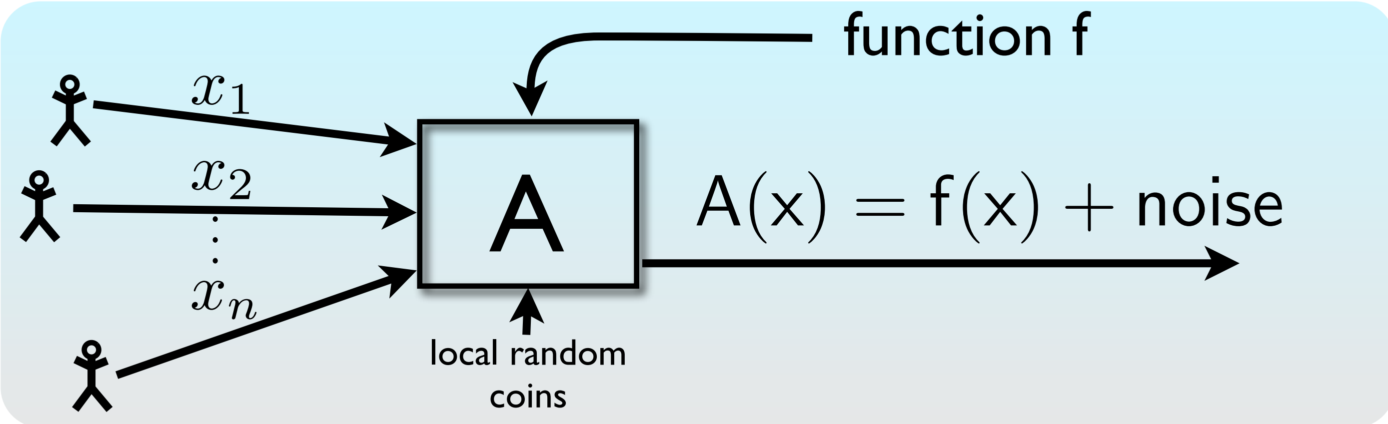
- Laplace distribution $\mathsf{Lap}(\lambda)$ has density

$$h(y) \propto e^{-|y|/\lambda}$$



- Changing one point translates curve

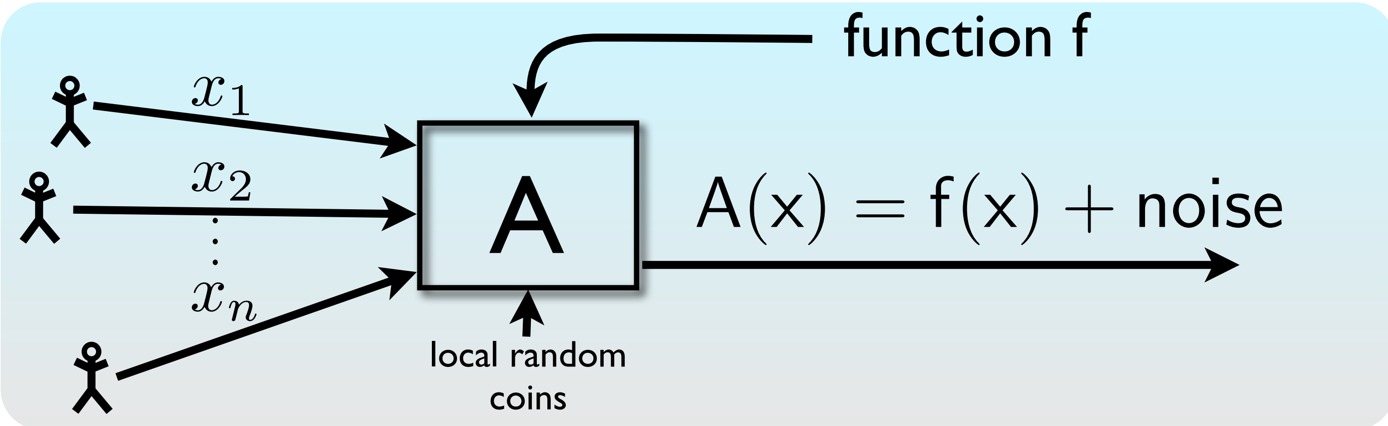
Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



$$\mathrm{GS}_{\text{proportion}} = \frac{1}{n}$$

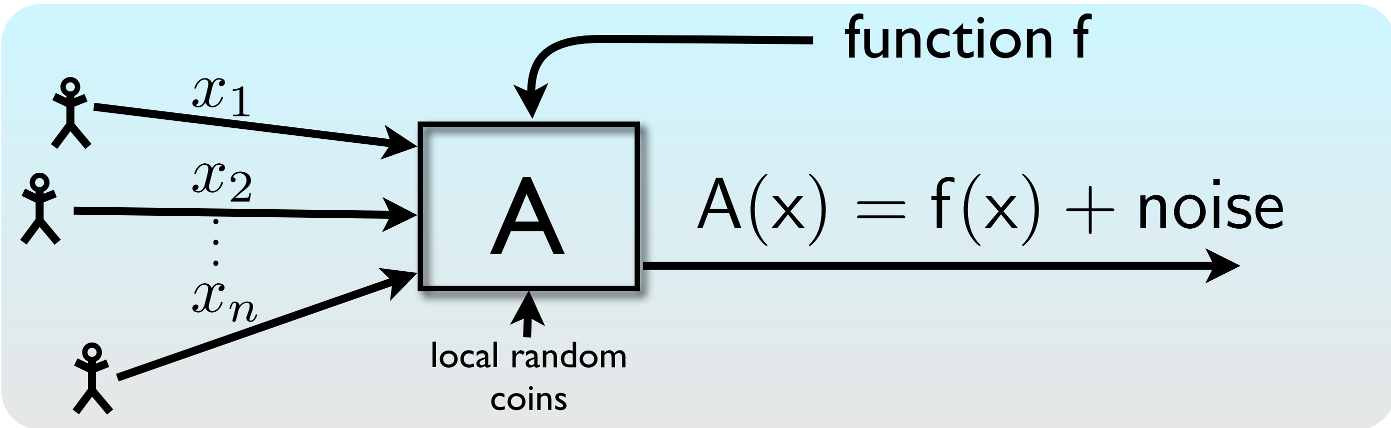
$$A(x) = \text{proportion} \pm \frac{1}{\epsilon n}$$

Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]

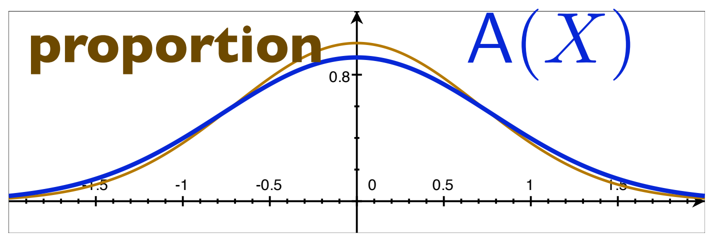


- Example: proportion of diabetics
  - ➢ $\mathsf{GS}_{\text{proportion}} = \frac{1}{n}$
  - ➢ Release $\mathsf{A(x)} = \text{proportion} \pm \frac{1}{\epsilon n}$

Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



- Example: proportion of diabetics
  - $GS_{\text{proportion}} = \frac{1}{n}$
  - Release $A(x) = \text{proportion} \pm \frac{1}{\epsilon n}$
- Is this **a lot**?
  - If x is a random sample from a large underlying population, then **sampling noise** $\approx \frac{1}{\sqrt{n}}$
  - A(x) "as good as" real proportion



Using global sensitivity

$$\mathrm{GS}_f = \max_{\text{neighbors } x, x'} \|f(x) - f(x')\|_1$$

- Many natural functions have low sensitivity
  - e.g., **histogram**, mean, **covariance matrix**, distance to a function, estimators with bounded "sensitivity curve", strongly convex optimization problems
- Laplace mechanism can be a programming interface
  - Many algorithms can be expressed as a sequence of low-sensitivity queries [BDMN '05, FFKN'09, MW'10]
  - Implemented in several systems [McSherry '09, Roy et al. '10, Haeberlen et al. '11, Moharan et al. '12]

Interpreting the definition

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

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

Interpreting the definition

- ε cannot be negligible
  - ➢ $A(0^n)$ and $A(1^n)$ at distance at most $n\varepsilon$
  - ➢ Need $\varepsilon \gg 1/n$ to get utility

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

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

Interpreting the definition

- ε cannot be negligible
  - A(0^n) and A(1^n) at distance at most nε
  - Need ε ≫ 1/n to get utility
- Why this distance measure?
  - Consider a mechanism that publishes 1 random person's data
    - Stat. Diff. (A(x),A(x')) = 1/n
  - Need a "worst case" distance measure

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

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

Interpreting the definition

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

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

Interpreting the definition

- **Composition Lemma:**
  If A_1 and A_2 are **ε**-differentially private,
  then joint output (A_1, A_2) is **2ε**-differentially private.

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

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

Neighboring databases induce **close** distributions on outputs

Interpreting the definition

- **Composition Lemma:**
  If A_1 and A_2 are **ε**-differentially private,
  then joint output (A_1, A_2) is **2ε**-differentially private.

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

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

Interpreting the definition

- **Composition Lemma:**
  If A_1 and A_2 are **ε**-differentially private,
  then joint output (A_1,A_2) is **2ε**-differentially private.

- Meaningful in the presence of arbitrary external information

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

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

Interpreting Differential Privacy

Interpreting Differential Privacy

- A naïve hope:

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

Interpreting Differential Privacy

- A naïve hope:

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

- Suppose you know that I smoke
  - A public health study could teach you that I am at risk for cancer
  - But it didn't matter whether or not my data was part of it.

Interpreting Differential Privacy

- A naïve hope:

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

- Suppose you know that I smoke
  - A public health study could teach you that I am at risk for cancer
  - But it didn't matter whether or not my data was part of it.

Interpreting Differential Privacy

- A naïve hope:

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

- Suppose you know that I smoke
  - A public health study could teach you that I am at risk for cancer
  - But it didn't matter whether or not my data was part of it.
- **Theorem** [DN'06, KM'11]: Learning things about individuals is unavoidable in the presence of external information

Interpreting Differential Privacy

- A naïve hope:

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

- Suppose you know that I smoke
  - ➢ A public health study could teach you that I am at risk for cancer
  - ➢ But it didn't matter whether or not my data was part of it.
- **Theorem** [DN'06, KM'11]: Learning things about individuals is unavoidable in the presence of external information
- Differential privacy implies:
  No matter what you know ahead of time,

> You learn (almost) the same things about me whether or not my data is used

  - ➢ This has a clean Bayesian interpretation [GKS'08]

Features or bugs?

Features or bugs?

- May not protect sensitive global information, e.g.

Features or bugs?

- May not protect sensitive global information, e.g.
  - Clinical data: Smoking and cancer

Features or bugs?

- May not protect sensitive global information, e.g.
  - Clinical data: Smoking and cancer
  - Financial transactions: firm-level trading strategies

Features or bugs?

- May not protect sensitive global information, e.g.
  - Clinical data: Smoking and cancer
  - Financial transactions: firm-level trading strategies
  - Social data: what if my presence affects everyone else? [KM'11]
    - The annoying colleague example

Features or bugs?

- May not protect sensitive global information, e.g.
  - ➢ Clinical data: Smoking and cancer
  - ➢ Financial transactions: firm-level trading strategies
  - ➢ Social data: what if my presence affects everyone else? [KM'11]
    - The annoying colleague example
  - ➢ Exact (deterministic) information about this data set
    - E.g., I know the differences in population between all 50 states
    - Differentially private release allows my to learn the populations exactly

Features or bugs?

- May not protect sensitive global information, e.g.
  - Clinical data: Smoking and cancer
  - Financial transactions: firm-level trading strategies
  - Social data: what if my presence affects everyone else? [KM'11]
    - The annoying colleague example
  - Exact (deterministic) information about this data set
    - E.g., I know the differences in population between all 50 states
    - Differentially private release allows my to learn the populations exactly
- Leakage accumulates
  - ε adds up with many releases
  - Inevitable in some form?
  - How do we set ε?

Variations on the approach

Variations on the approach

- Predecessors [DDN'03,EGS'03,DN'04,BDMN'05]

Variations on the approach

- Predecessors [DDN'03,EGS'03,DN'04,BDMN'05]
- (ε,δ)- differential privacy
  - Require $\Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) \leq e^{\epsilon} \cdot \Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) + \delta$
  - Similar semantics to (ε,0)- diffe.p. when δ ≪ 1/n

Variations on the approach

- Predecessors [DDN'03,EGS'03,DN'04,BDMN'05]
- (ε,δ)- differential privacy
  - Require $\Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) \leq e^{\epsilon} \cdot \Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) + \delta$
  - Similar semantics to (ε,0)- diffe.p. when δ « 1/n
- Computational variants [MPRV09,MMPRTV10,GKY11]

Variations on the approach

- Predecessors [DDN'03,EGS'03,DN'04,BDMN'05]
- (ε,δ)- differential privacy
  - Require $\Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) \leq e^{\epsilon} \cdot \Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) + \delta$
  - Similar semantics to (ε,0)- diffe.p. when δ ≪ 1/n
- Computational variants [MPRV09,MMPRTV10,GKY11]
- Distributional variants [RHMS'09,BBGLT'11,...]
  - Assume something about adversary's prior distribution
  - Deterministic releases
  - Poor composition guarantees

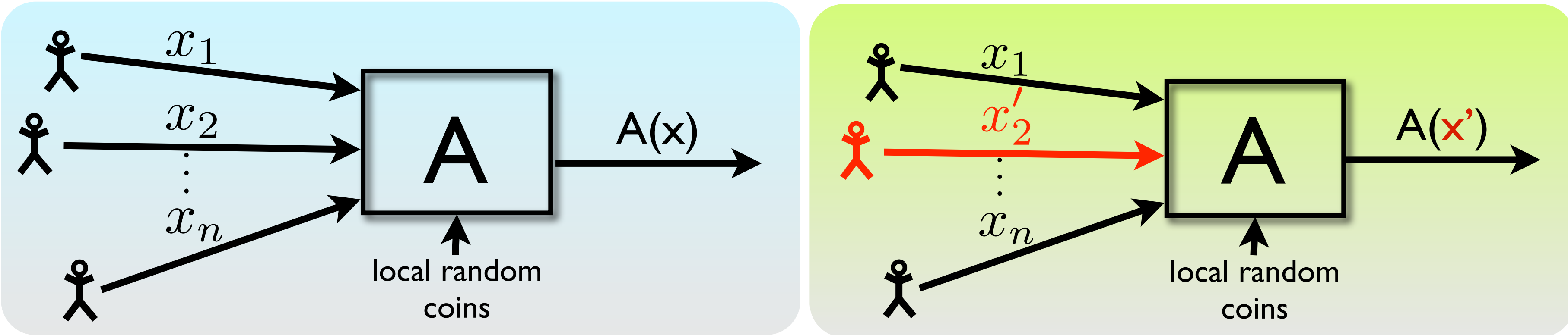
Variations on the approach

- Predecessors [DDN'03,EGS'03,DN'04,BDMN'05]
- (ε,δ)- differential privacy
  - Require $\Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) \leq e^{\epsilon} \cdot \Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) + \delta$
  - Similar semantics to (ε,0)- diffe.p. when δ ≪ 1/n
- Computational variants [MPRV09,MMPRTV10,GKY11]
- Distributional variants [RHMS'09,BBGLT'11,...]
  - Assume something about adversary's prior distribution
  - Deterministic releases
  - Poor composition guarantees
- Generalizations
  - [BLR'08, GLP'11] simulation-based definitions
  - [KM'12] "Pufferfish": vast generalization, tricky to instantiate

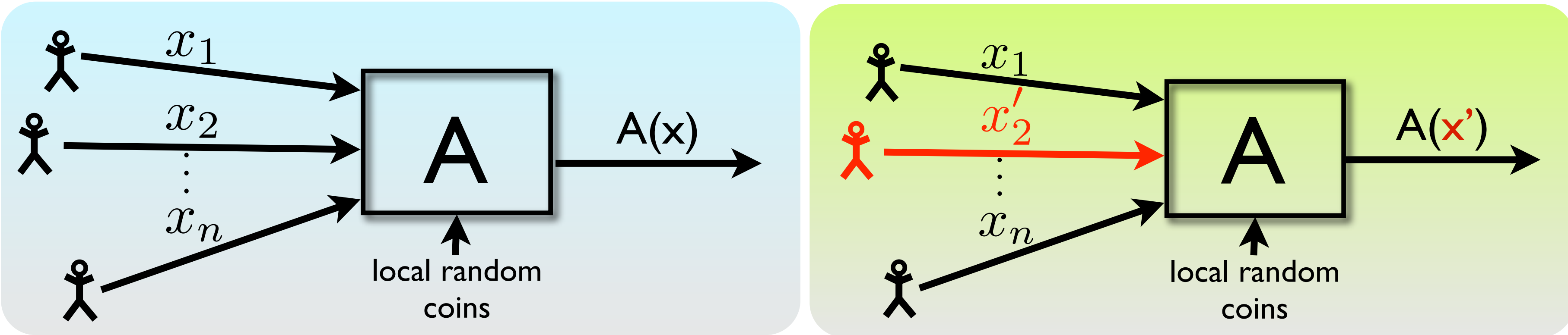
Variations on the approach

- Predecessors [DDN'03,EGS'03,DN'04,BDMN'05]
- (ε,δ)- differential privacy
  - Require $\Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) \leq e^{\epsilon} \cdot \Pr(\mathsf{A}(\mathsf{x}) \in \mathsf{S}) + \delta$
  - Similar semantics to (ε,0)- diffe.p. when δ « 1/n
- Computational variants [MPRV09,MMPRTV10,GKY11]
- Distributional variants [RHMS'09,BBGLT'11,...]
  - Assume something about adversary's prior distribution
  - Deterministic releases
  - Poor composition guarantees
- Generalizations
  - [BLR'08, GLP'11] simulation-based definitions
  - [KM'12] "Pufferfish": vast generalization, tricky to instantiate
- Crowd-blending privacy [GHLP'12]

What can we compute privately?

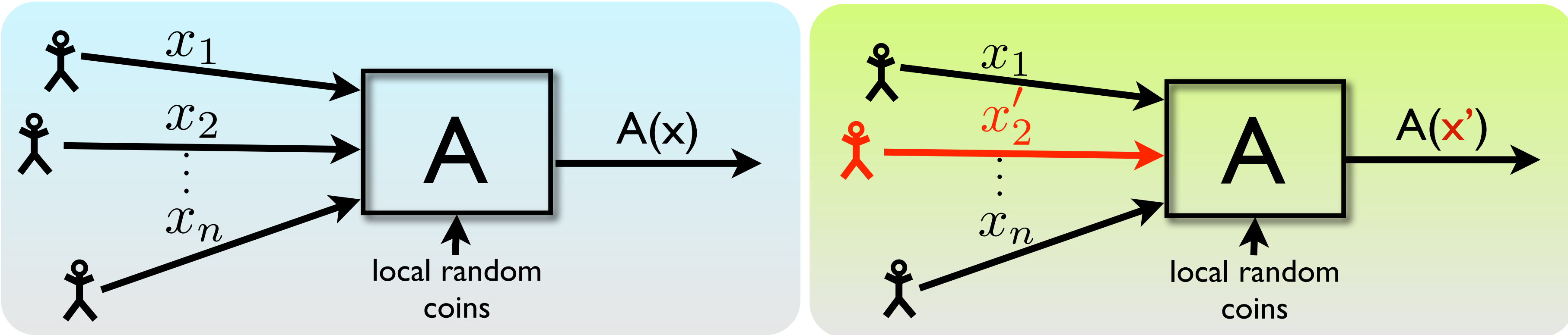


What can we compute privately?



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

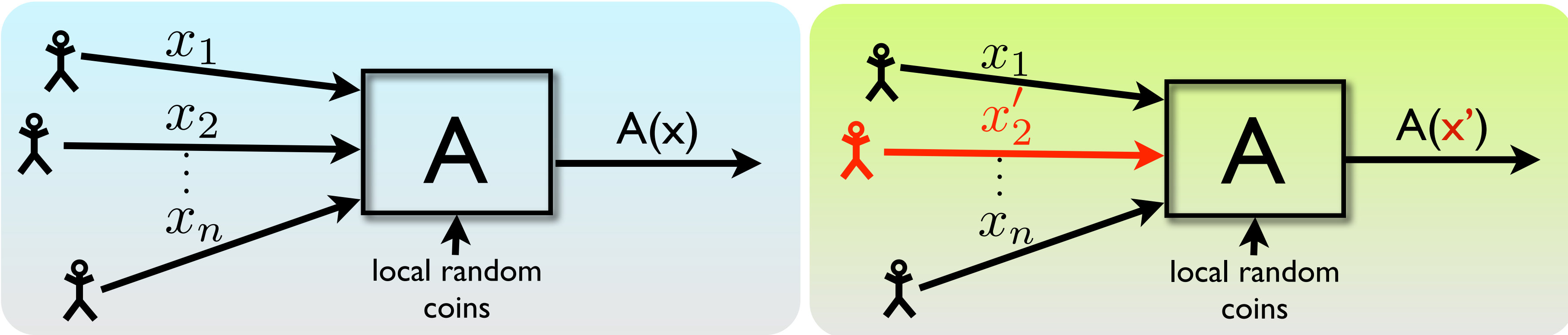
What can we compute privately?



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

What computational tasks can we achieve privately?

What can we compute privately?

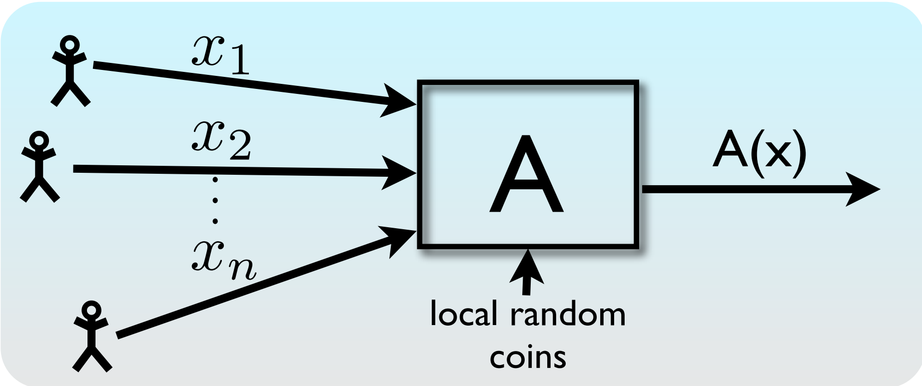


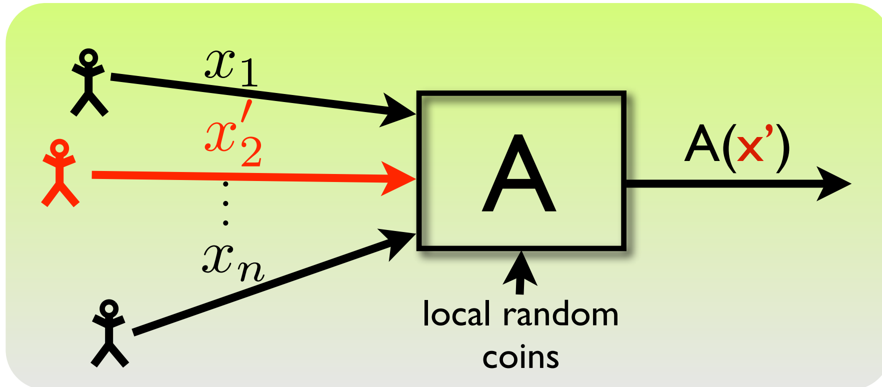
- "Privacy" = change in one input leads to small change in output distribution

What computational tasks can we achieve privately?

- General tools for reasoning about leakage

What can we compute privately?



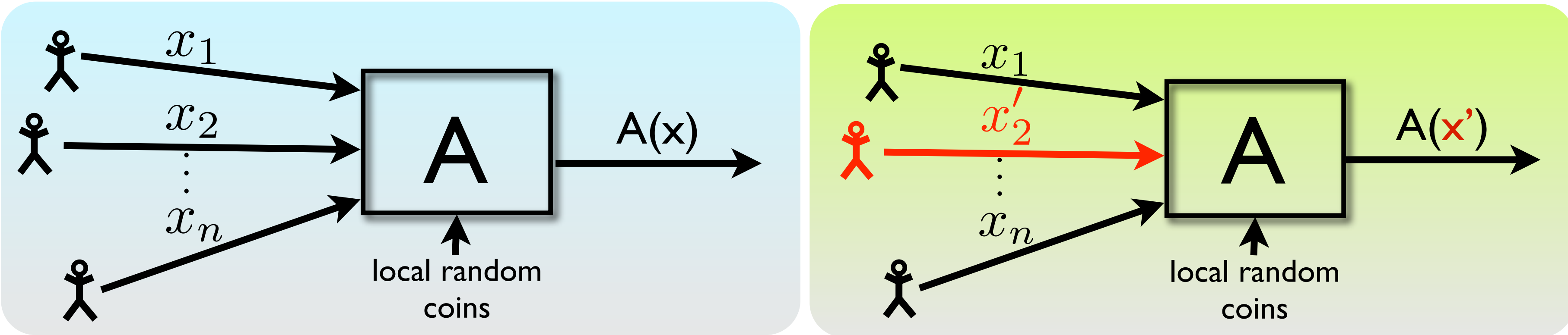


- "Privacy" = change in one input leads to small change in output distribution

What computational tasks can we achieve privately?

- General tools for reasoning about leakage
- Lots of recent work, interesting questions
  - ➢ STOC,FOCS,SODA, PODS, SIGMOD,VLDB, KDD, CCS, S&P, Usenix Sec., NIPS, COLT, Crypto/Eurocrypt,TCC, SIGCOMM, JSM, JASA ...

What can we compute privately?

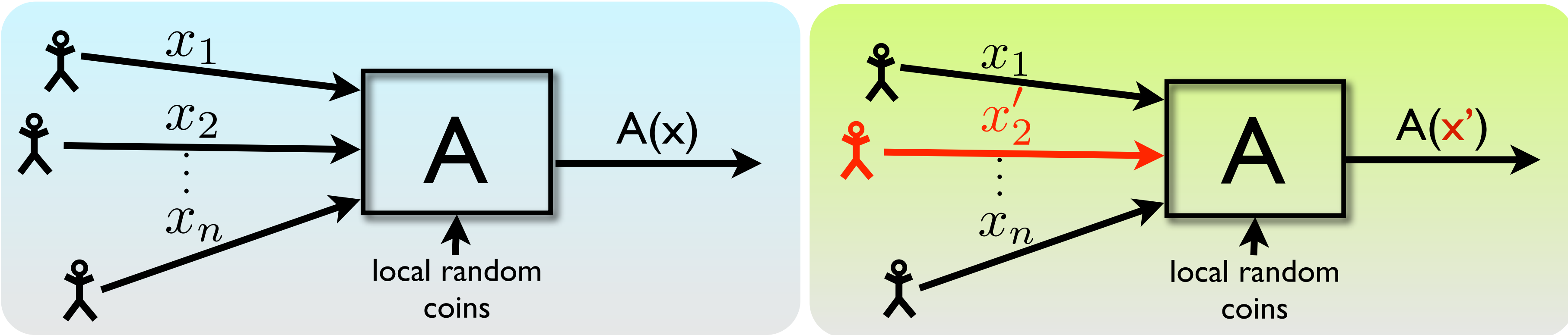


- "Privacy" = change in one input leads to small change in output distribution

What computational tasks can we achieve privately?

- General tools for reasoning about leakage
- Lots of recent work, interesting questions
  - STOC, FOCS, SODA, PODS, SIGMOD, VLDB, KDD, CCS, S&P, Usenix Sec., NIPS, COLT, Crypto/Eurocrypt, TCC, SIGCOMM, JSM, JASA ...

What can we compute privately?

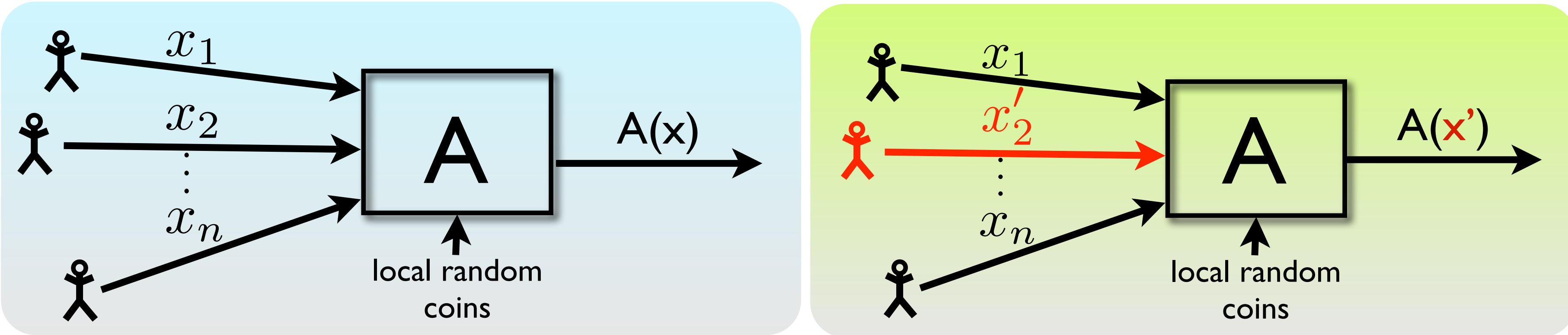


- "Privacy" = change in one input leads to small change in output distribution

What computational tasks can we achieve privately?

- General tools for reasoning about leakage
- Lots of recent work, interesting questions
  - STOC,FOCS,SODA, PODS, SIGMOD,VLDB, KDD, CCS, S&P, Usenix Sec., NIPS, COLT, Crypto/Eurocrypt,TCC, SIGCOMM, JSM, JASA ...

What can we compute privately?

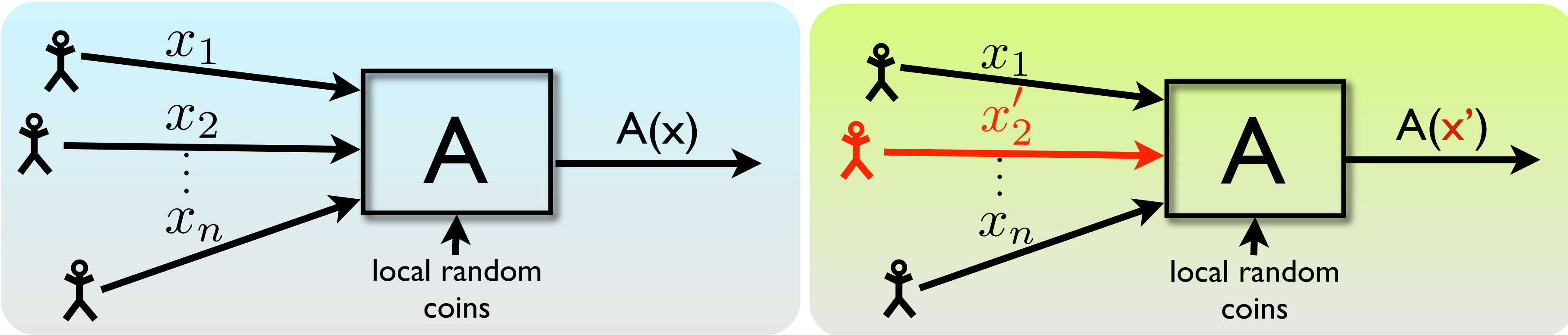


- "Privacy" = change in one input leads to small change in output distribution

What computational tasks can we achieve privately?

- General tools for reasoning about leakage
- Lots of recent work, interesting questions
  - STOC,FOCS,SODA, PODS, SIGMOD,VLDB, KDD, CCS, S&P, Usenix Sec., NIPS, COLT, Crypto/Eurocrypt, TCC, SIGCOMM, JSM, JASA ...

What can we compute privately?

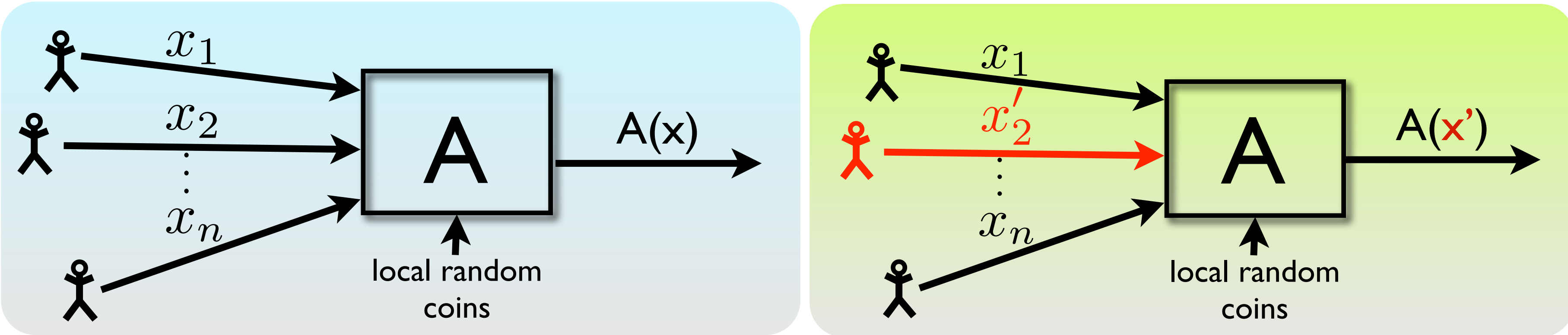


- "Privacy" = change in one input leads to small change in output distribution

What computational tasks can we achieve privately?

- General tools for reasoning about leakage
- Lots of recent work, interesting questions
  - STOC, FOCS, SODA, PODS, SIGMOD, VLDB, KDD, CCS, S&P, Usenix Sec., NIPS, COLT, Crypto/Eurocrypt, TCC, SIGCOMM, JSM, JASA ...

What can we compute privately?

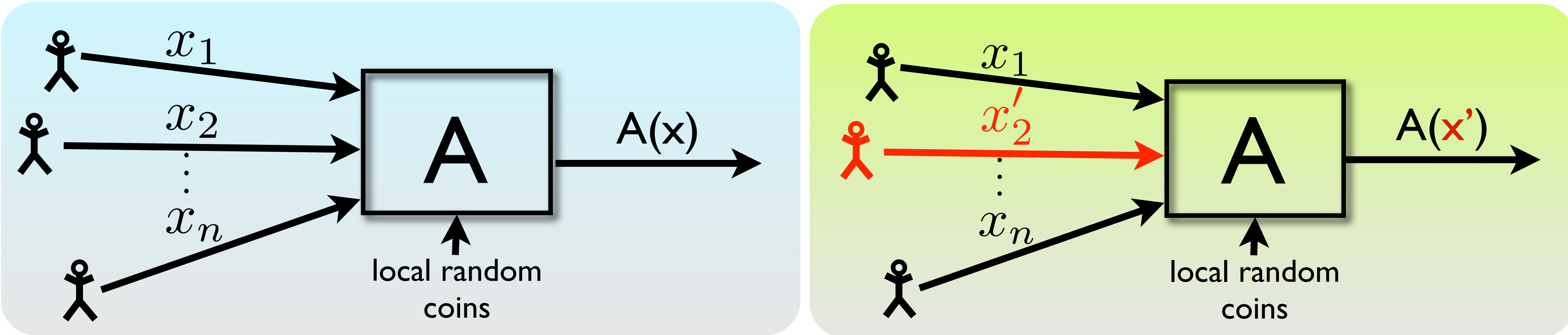


- "Privacy" = change in one input leads to small change in output distribution

What computational tasks can we achieve privately?

- General tools for reasoning about leakage
- Lots of recent work, interesting questions
  - STOC, FOCS, SODA, PODS, SIGMOD, VLDB, KDD, CCS, S&P, Usenix Sec., NIPS, COLT, Crypto/Eurocrypt, TCC, SIGCOMM, JSM, JASA ...

What can we compute privately?

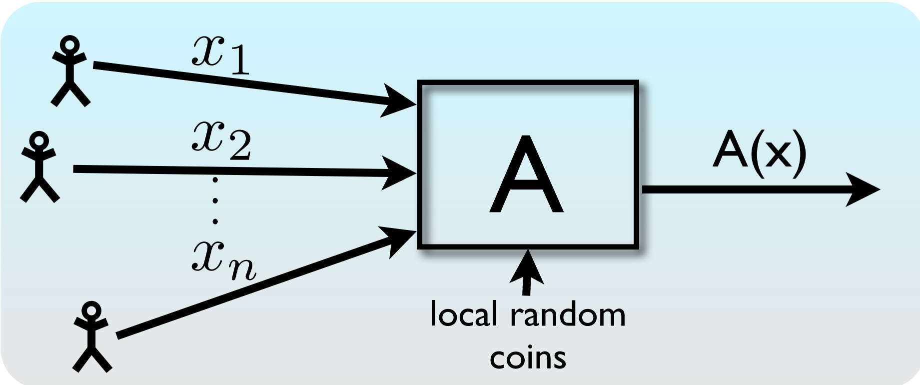


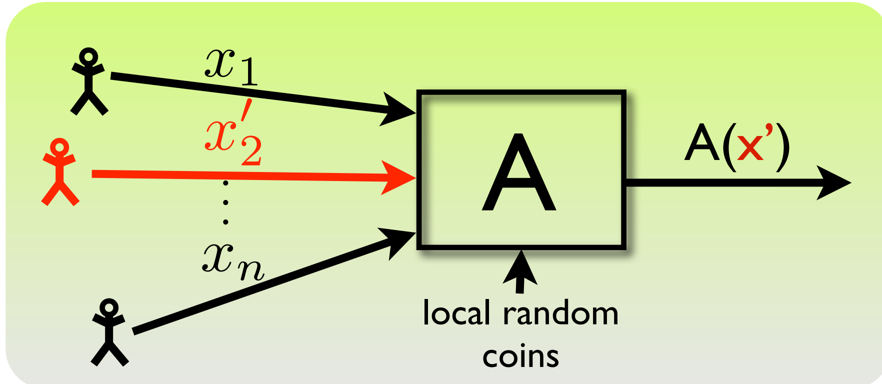
- "Privacy" = change in one input leads to small change in output distribution

What computational tasks can we achieve privately?

- General tools for reasoning about leakage
- Lots of recent work, interesting questions
  - ➢ STOC,FOCS,SODA, PODS, SIGMOD,VLDB, KDD, CCS, S&P, Usenix Sec., NIPS, COLT, Crypto/Eurocrypt,TCC, SIGCOMM, JSM, JASA ...

What can we compute privately?





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

What computational tasks can we achieve privately?

- General tools for reasoning about leakage
- Lots of recent work, interesting questions
  - STOC, FOCS, SODA, PODS, SIGMOD, VLDB, KDD, CCS, S&P, Usenix Sec., NIPS, COLT, Crypto/Eurocrypt, TCC, SIGCOMM, JSM, JASA ...

This talk

- **Act I: Attacks**
  - (Why is privacy hard?)
  - Reconstruction attacks
- **Act II: Definitions**
  - One approach: "differential" privacy
  - Variations on the theme
- **Act III: Algorithms**
  - Basic techniques: noise addition, exponential sampling
  - Answering many queries
  - Exploiting "local" sensitivity

This talk

- **Act I: Attacks**
  - ➢ (Why is privacy hard?)
  - ➢ Reconstruction attacks
- **Act II: Definitions**
  - ➢ One approach: "differential" privacy
  - ➢ Variations on the theme
- **Act III: Algorithms**
  - ➢ Basic techniques: noise addition, exponential sampling
  - ➢ Answering many queries
  - ➢ Exploiting "local" sensitivity

Differentially Private Algorithms

- Tools and Techniques
  - ➢ Laplace Mechanism
  - ➢ Exponential Mechanism
  - ➢ Algorithms for many queries
  - ➢ Local Sensitivity-based techniques
- Theoretical Foundations
  - ➢ Feasibility results: Learning, optimization, synthetic data, statistics
  - ➢ Connections to game theory, learning, robustness
- Domain-specific algorithms
  - ➢ Networking, clinical data, social networks, ...
- Systems
  - ➢ Programming Languages, Query Languages, Attacks

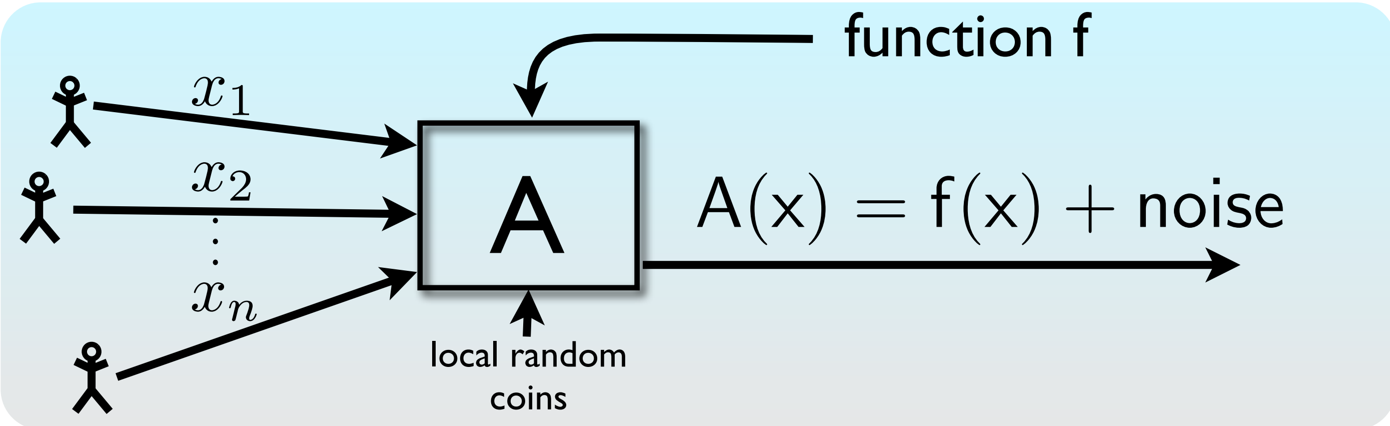
Differentially Private Algorithms

- Tools and Techniques
  - Laplace Mechanism
  - Exponential Mechanism
  - Algorithms for many queries
  - Local Sensitivity-based techniques

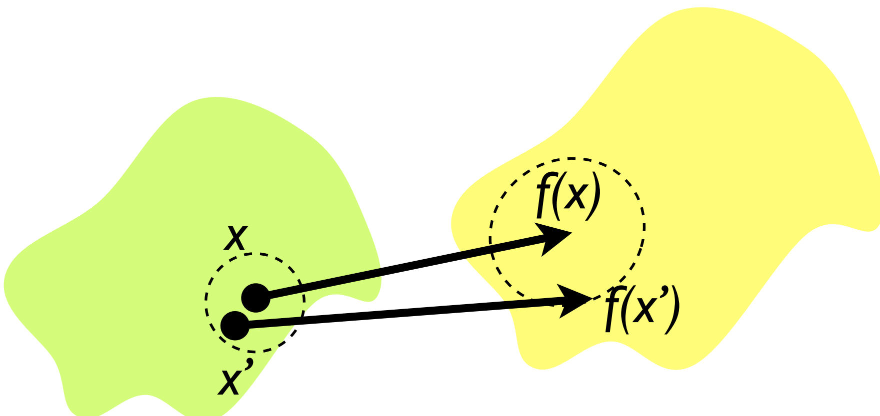
- Theoretical Foundations
  - Feasibility results: Learning, optimization, synthetic data, statistics
  - Connections to game theory, learning, robustness
- Domain-specific algorithms
  - Networking, clinical data, social networks, ...
- Systems
  - Programming Languages, Query Languages, Attacks

Basic Technique 1:
Noise Addition

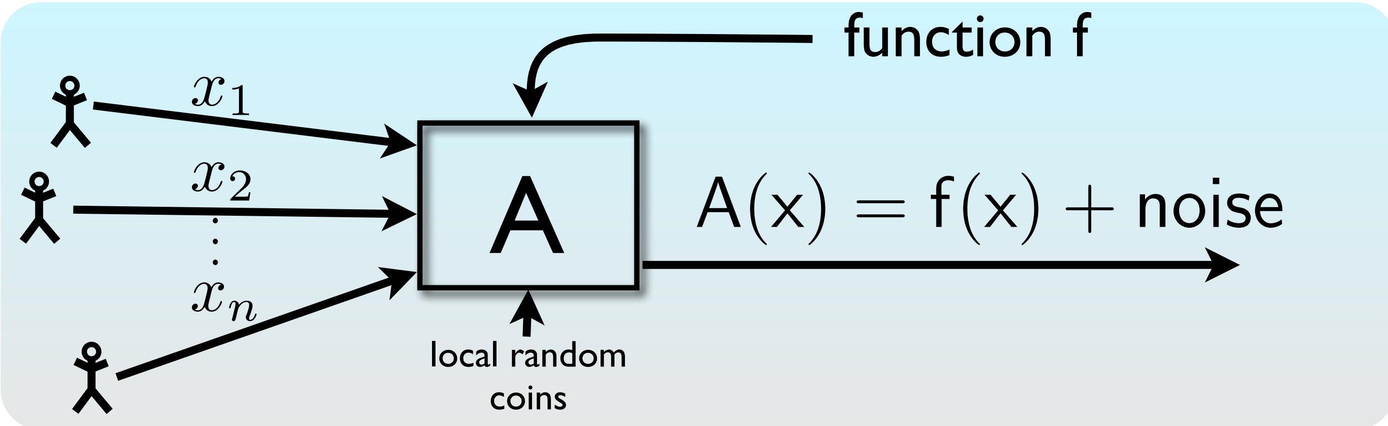
Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



- **Global Sensitivity:** $GS_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
  - Example: $GS_{\text{proportion}} = \frac{1}{n}$



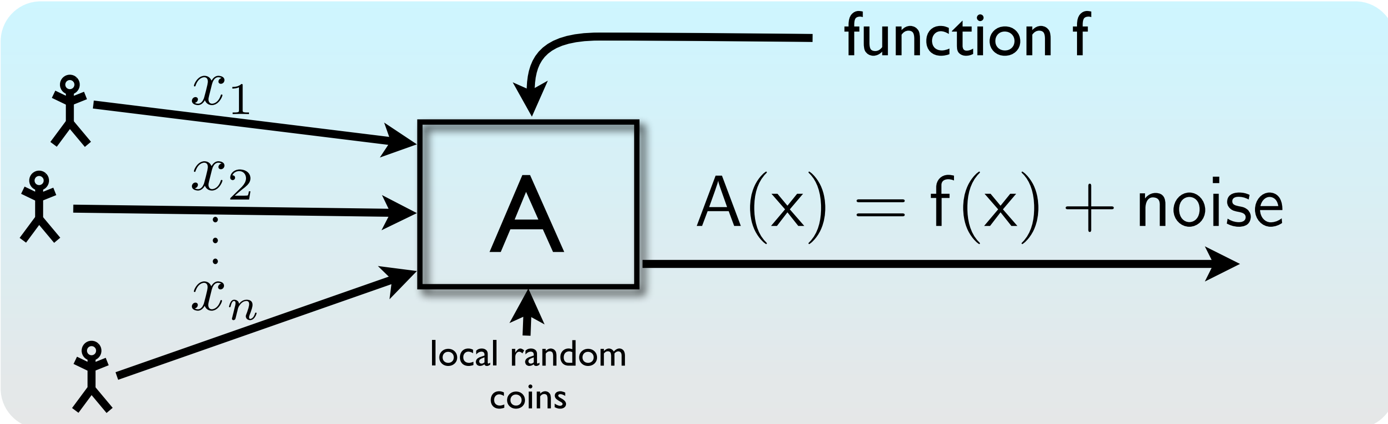
Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



$$\mathrm{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$$

$$\mathrm{GS}_{\text{proportion}} = \frac{1}{n}$$

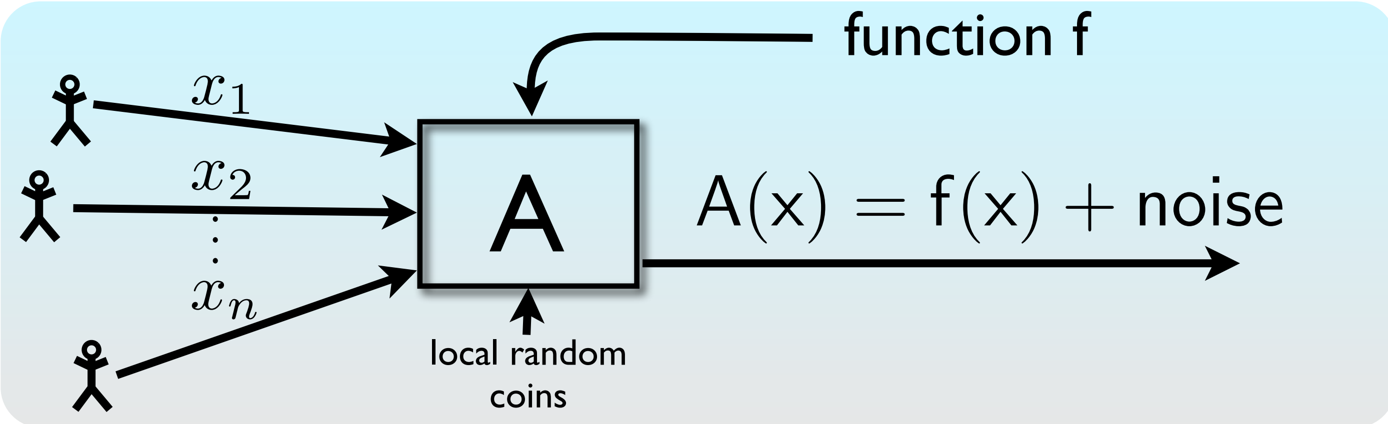
Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



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

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

Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]

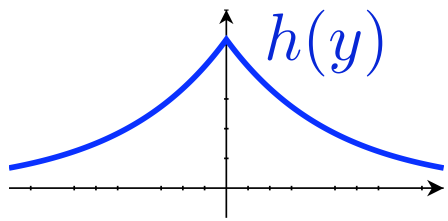


- **Global Sensitivity:** $\mathrm{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
  - Example: $\mathrm{GS}_{\text{proportion}} = \frac{1}{n}$

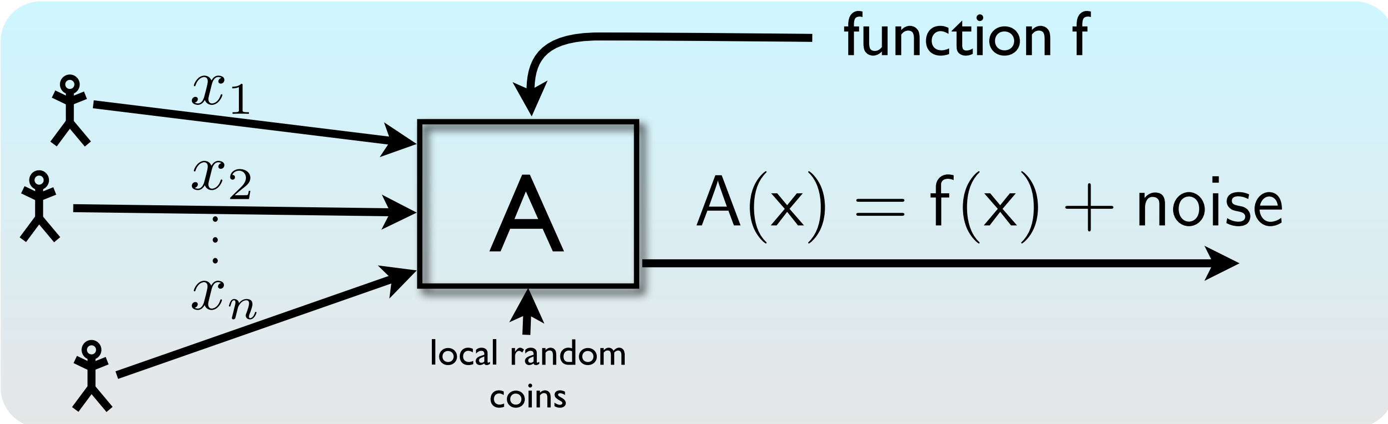
Theorem: If $A(x) = f(x) + \mathrm{Lap}\left(\frac{\mathrm{GS}_f}{\epsilon}\right)$, then A is ϵ-differentially private.

- Laplace distribution $\mathrm{Lap}(\lambda)$ has density

$$h(y) \propto e^{-|y|/\lambda}$$



Example: Noise Addition [Dwork, McSherry, Nissim, S. 2006]



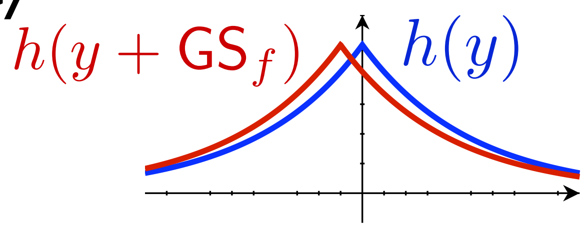
- **Global Sensitivity:** $\mathsf{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
  - Example: $\mathsf{GS}_{\text{proportion}} = \frac{1}{n}$

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

$$h(y) \propto e^{-|y|/\lambda}$$

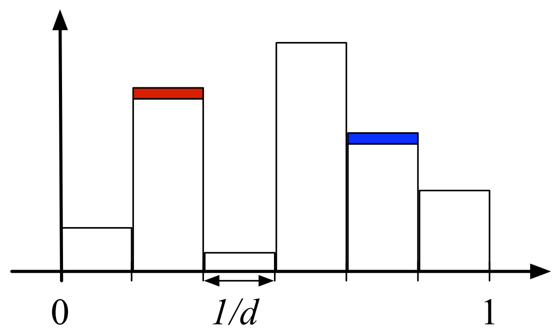
- Changing one point translates curve



Example: Histograms

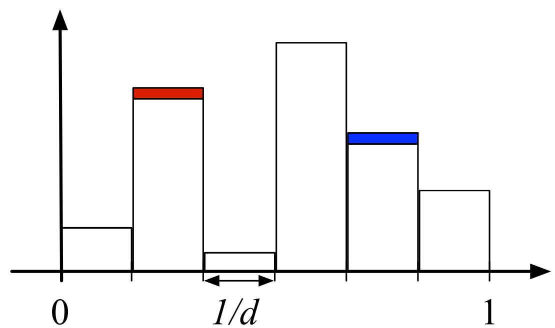
$$\mathsf{f(x)} = (\mathsf{n_1}, \mathsf{n_2}, \ldots, \mathsf{n_d}) \text{ where } \mathsf{n_j} = \#\{\mathsf{i} : \mathsf{x_i} \text{ in } j\text{-th bin}\}$$

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



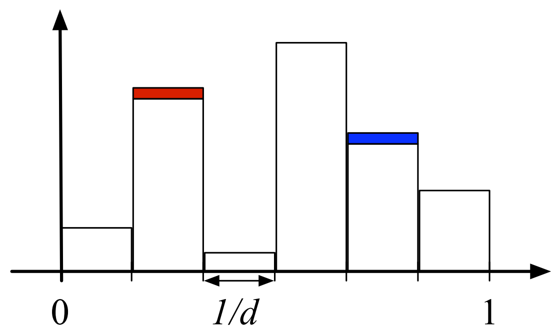
Example: Histograms

- Say $x_1, x_2, \ldots, x_n$ in domain D
  - Partition D into d disjoint bins
  - $f(x) = (n_1, n_2, \ldots, n_d)$ where $n_j = \#\{i : x_i \text{ in } j\text{-th bin}\}$
  - $GS_f = 1$
  - Sufficient to add noise $\mathrm{Lap}(1/\epsilon)$ to each count



Example: Histograms

- Say $x_1, x_2, \ldots, x_n$ in domain D
  - Partition D into d disjoint bins
  - $f(x) = (n_1, n_2, \ldots, n_d)$ where $n_j = \#\{i : x_i \text{ in } j\text{-th bin}\}$
  - $GS_f = 1$
  - Sufficient to add noise $\mathrm{Lap}(1/\epsilon)$ to each count
- Examples
  - Histogram on the line
  - Populations of 50 states
  - Marginal tables
    - bins = possible combinations of attributes



Marginal 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-)
  - Add constant noise to counts to achieve differential privacy
  - Change to proportions is $O(\frac{1}{n})$
- Problems for practice:
  - Some entries may be negative. Multiple tables inconsistent.
  - [BCDKMT07] Multiple noisy tables can be "rounded" to a consistent set of tables corresponding to real data.

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)

Variants in other metrics

- Consider $f : \mathcal{D}^n \to \mathbb{R}^d$
- Global Sensitivity: $GS_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_{\cancel{1}\,2}$

Theorem: If $A(x) = f(x) + \cancel{\mathrm{Lap}\left(\frac{GS_f}{\epsilon}\right)^d}$ $N\left(0, \left(\frac{GS_f \cdot 3 \cdot \sqrt{\ln(1/\delta)}}{\epsilon}\right)^2\right)$, then A is $\cancel{\epsilon}$ (ϵ, δ)-differentially private.

- Example: Ask for counts of d predicates
  - f(x) = vector of counts.
  - $GS_f = \sqrt{d}$
  - Add noise $\frac{\sqrt{d \ln(1/\delta)}}{\epsilon}$ per entry instead of $\frac{d}{\epsilon}$

Basic Technique 2:
Exponential Sampling

Exponential Sampling [McSherry-Talwar 2007]

- Sometimes noise addition makes no sense
  - ➢ mode of a distribution
  - ➢ minimum cut in a graph
  - ➢ classification rule

- [MT07] Motivation: auction design
  - ➢ Differential privacy implies approximate truthfulness
  - ➢ Generated line of work on privacy and game theory
- Subsequently applied very broadly

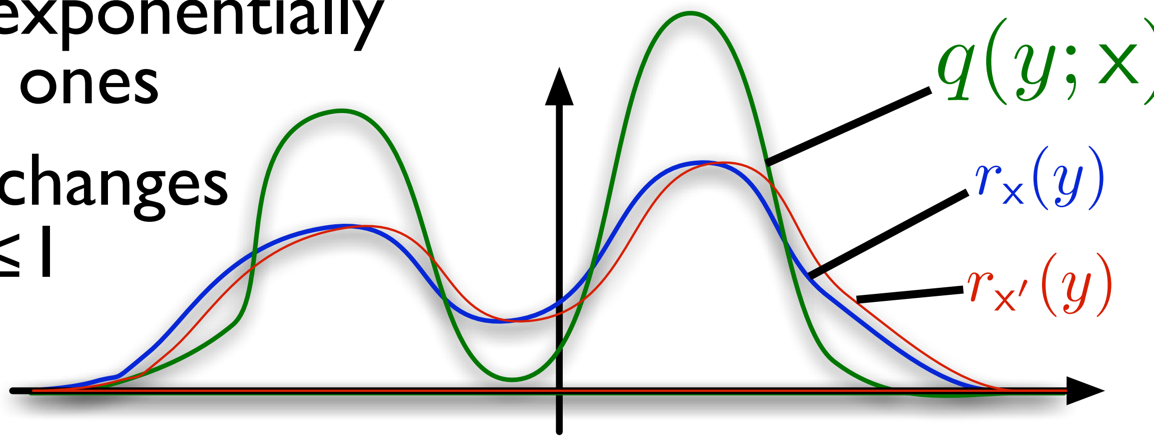
Example: Voting

- Data: x_i = {websites visited by student i today}
- Range: Y = {website names}
- For each name y, let q(y; x) = #{i : x_i contains y}
- Goal: output the most frequently visited site

Mechanism: Given x,

- Output website y_0 with probability $r_{\mathsf{x}}(y) \propto \exp(\epsilon q(y;\mathsf{x}))$

- Utility: Popular sites exponentially more likely than rare ones
- Privacy: One person changes websites' scores by ≤ 1



Example: Voting

Mechanism: Given x,

- Output website y_0 with probability $r_{\mathsf{x}}(y) \propto \exp(\epsilon q(y;\mathsf{x}))$

- **Claim:** Mechanism is 2ε-differentially private
- Proof: $\dfrac{r_{\mathsf{x}}(y)}{r_{\mathsf{x}'}(y)} = \dfrac{e^{\epsilon q(y;\mathsf{x})}}{e^{\epsilon q(y;\mathsf{x}')}} \cdot \dfrac{\sum_{z\in Y} e^{\epsilon q(z;\mathsf{x}')}}{\sum_{z\in Y} e^{\epsilon q(z;\mathsf{x})}} \leq e^{2\epsilon}$

- **Claim:** If most popular website has score *T*, then

$$\mathbb{E}[q(y_0;x)] \geq T - (\log|Y|)/\epsilon$$

- Proof: Output y is bad if q(y;x) < T - k
  - $\Pr(\text{bad outputs}) \leq \dfrac{\Pr(\text{bad outputs})}{\Pr(\text{best output})} \leq \dfrac{|Y|e^{\epsilon(T-k)}}{e^{\epsilon T}} \leq e^{\log|Y| - \epsilon k}$
  - Get expectation bound via formula $E(Z) = \sum_{k>0} \Pr(Z \geq k)$

Exponential Sampling

Ingredients:

- Set of outputs Y with prior distribution p(y)
- Score function q(y;x) such that
  for all outputs y, neighbors x,x': |q(y;x) - q(y;x')| ≤ 1

Mechanism: Given x,

- Output y_0 from Y with probability $r_{\mathsf{x}}(y) \propto p(y)e^{-\epsilon q(y;\mathsf{x})}$

- **Example** [KLNRS'08]:
  - ➢ Y= set of possible classifiers (say, discretized half-planes)
  - ➢ q(y;x) = -(error rate of classifier y on data x)
  - ➢ Output a classifier with expected
    error rate (OPT + log|Y| /ε n)

- **Corollary**: Every PAC learnable class
  is privately PAC learnable.

Exponential Sampling

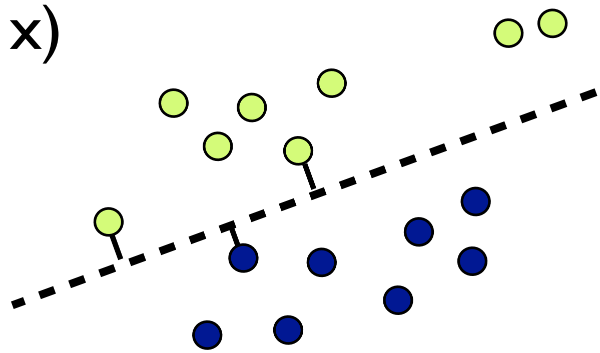
Ingredients:

- Set of outputs Y with prior distribution p(y)
- Score function q(y;x) such that
  for all outputs y, neighbors x,x': $|q(y;x) - q(y;x')| \le 1$

Mechanism: Given x,

- Output y_0 from Y with probability $r_x(y) \propto p(y) e^{-\epsilon q(y;x)}$

- **Example** [KLNRS'08]:
  - Y= set of possible classifiers (say, discretized half-planes)
  - q(y;x) = -(error rate of classifier y on data x)
  - Output a classifier with expected error rate (OPT + log|Y| /ε n)



- **Corollary**: Every PAC learnable class is privately PAC learnable.

Using Exponential Sampling

- Mechanism above very general
  - Every differentially private mechanism is an instance!
  - Still a useful design perspective
- Perspective used explicitly for
  - Learning discrete classifiers [KLNRS'08]
  - Synthetic data generation [BLR'08,HLM'10]
  - Convex Optimization [CM'08,CMS'10]
  - Frequent Pattern Mining [BLST'10]
  - Genome-wide association studies [FUS'11]
  - High-dimensional sparse regression [KST'12]

Releasing Many Functions

Linear Queries

Data x = multi-set in domain D

- Represent as vector $\vec{x} \in \mathbb{R}^{|D|}$: $\vec{x}(i) = \frac{\#\text{occurrences of } i \text{ in } x}{n}$

Linear Queries are functions $f : D \to [0, 1]$,

- Answer of f on x is $\sum_{i \in x} f(i) = \left\langle \vec{f}, \vec{x} \right\rangle$
- Special cases: Subset queries (with right representation), most low-sensitivity queries people use

Goal: given queries $f_1, .., f_m$, release $\hat{f}_1, ..., \hat{f}_m$ to minimize

$$error = \max_j \left| \hat{f}_j - \langle f_j, x \rangle \right|$$

How low can *error* be in terms of $m, n, |D|$?

Linear Queries

Goal: given queries $f_1, .., f_m$, minimize $error = \max_j \left| \hat{f}_j - \langle f_j, x \rangle \right|$

Laplace mechanism + composition results

- $error = O(\frac{m \log m}{\varepsilon n}$ or $O(\frac{\sqrt{m \log m \log(1/\delta)}}{\varepsilon n})$
- Time $O(mn)$
- Only useful if $m \ll n^2$.

Linear Queries

Goal: given queries $f_1, .., f_m$, minimize $\mathit{error} = \max_j \left|\hat{f}_j - \langle f_j, x\rangle\right|$

Laplace mechanism + composition results

- $\mathit{error} = O(\frac{m \log m}{\varepsilon n}$ or $O(\frac{\sqrt{m \log m \log(1/\delta)}}{\varepsilon n})$
- Time $O(mn)$
- Only useful if $m \ll n^2$.

Is this the best possible error?

- Yes, when $n \gg m$ [KRSU10,HT10]
- For $m \geq n$, reconstruction attacks rule out error $o(1/\sqrt{n})$.
- Randomly sampling t people from x gives error $O(\frac{\log m}{\sqrt{t}})$...

Linear Queries

Goal: given queries $f_1, .., f_m$, minimize $error = \max_j \left|\hat{f}_j - \langle f_j, x\rangle\right|$

Laplace mechanism + composition results

- $error = O(\frac{m \log m}{\varepsilon n})$ or $O(\frac{\sqrt{m \log m \log(1/\delta)}}{\varepsilon n})$
- Time $O(mn)$
- Only useful if $m \ll n^2$.

Is this the best possible error?

- Yes, when $n \gg m$ [KRSU10,HT10]
- For $m \geq n$, reconstruction attacks rule out error $o(1/\sqrt{n})$.
- Randomly sampling t people from x gives error $O(\frac{\log m}{\sqrt{t}})$...

  ... but shafts t people.

Linear Queries

Goal: given queries $f_1, .., f_m$, minimize $\mathit{error} = \max_j \left|\hat{f}_j - \langle f_j, x\rangle\right|$

Laplace mechanism + composition results

- $\mathit{error} = O(\frac{m \log m}{\varepsilon n}$ or $O(\frac{\sqrt{m \log m \log(1/\delta)}}{\varepsilon n})$
- Time $O(mn)$
- Only useful if $m \ll n^2$.

Is this the best possible error?

- Yes, when $n \gg m$ [KRSU10,HT10]
- For $m \geq n$, reconstruction attacks rule out error $o(1/\sqrt{n})$.
- Randomly sampling t people from x gives error $O(\frac{\log m}{\sqrt{t}})$...
  ... but shafts t people.
- [BLR'08,DNNRV'09,RR'10,HR'10,HLM'11,GRU'11,JT'12]:
  Error $O\left(\frac{\log m \cdot \log|D|}{(\varepsilon n)^{1/3}}\right)$ or $O\left(\frac{\log m \cdot \log|D| \cdot \log(1/\delta)}{(\varepsilon n)^{1/4}}\right)$.
  - Useful even when $m \gg n$:)
- Time $\tilde{O}(|D|m)$
  - Sometimes exponential :(

Linear Queries

Goal: given queries $f_1, .., f_m$, minimize $\mathit{error} = \max_j \left|\hat{f}_j - \langle f_j, x\rangle\right|$

Laplace mechanism + composition results

- $\mathit{error} = O(\frac{m \log m}{\varepsilon n}$ or $O(\frac{\sqrt{m \log m \log(1/\delta)}}{\varepsilon n})$
- Time $O(mn)$
- Only useful if $m \ll n^2$.

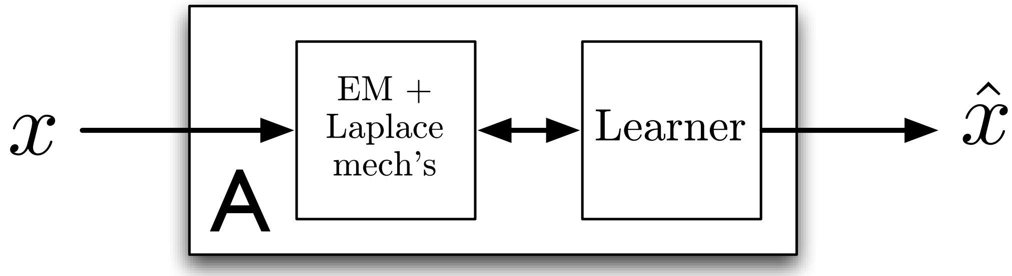
Is this the best possible error?

- Yes, when $n \gg m$ [KRSU10,HT10]
- For $m \geq n$, reconstruction attacks rule out error $o(1/\sqrt{n})$.
- Randomly sampling t people from x gives error $O(\frac{\log m}{\sqrt{t}})$...

  ... but shafts t people.
- [BLR'08,DNNRV'09,RR'10,HR'10,HLM'11,GRU'11,JT'12]:

  Error $O\left(\frac{\log m \cdot \log |D|}{(\varepsilon n)^{1/3}}\right)$ or $O\left(\frac{\log m \cdot \log |D| \cdot \log(1/\delta)}{(\varepsilon n)^{1/4}}\right)$.
  - Useful even when $m \gg n$:)
- Time $\tilde{O}(|D|m)$
  - Sometimes exponential :(

Idea: Learn the Data [DNRRV'09,HR'10,...]

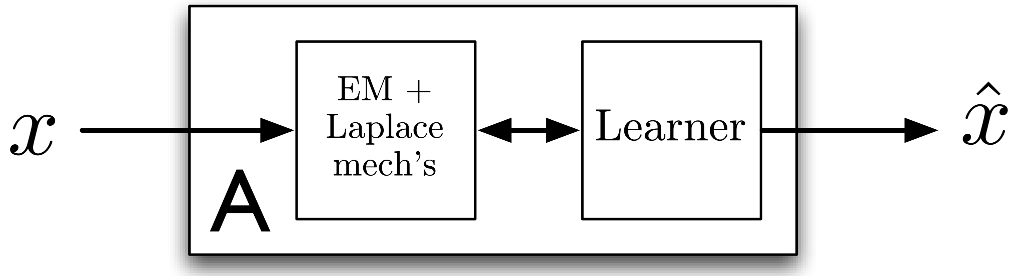


Release mechanism tries to "learn" x through diffe.p. interface

- Output $\hat{x}$ to minimize $\mathit{error}(\hat{x}) = \max_j |\langle f_j, \hat{x}\rangle - \langle f_j, x\rangle|$.
  (Generally do not have $\hat{x} \approx x$.)

Traditional learning	Privacy
Parameters of linear classifier	Data x
Training data	User's Queries f_j
Gradient computations	Actual data access

Idea: Learn the Data *[DNRRV'09,HR'10,...]*



Release mechanism tries to "learn" x through diffe.p. interface

- Output $\hat{x}$ to minimize $error(\hat{x}) = \max_j |\langle f_j, \hat{x}\rangle - \langle f_j, x\rangle|$. (Generally do not have $\hat{x} \approx x$.)

Traditional learning	Privacy
Parameters of linear classifier	Data x
Training data	User's Queries f_j
Gradient computations	Actual data access

- Learner computes a sequence of estimates $x_0, x_1, \ldots x_t, \ldots$
- Gradient: $\nabla error(\hat{x}_t) = \pm f_j$ where f_j maximizes error $|\langle f_j, \hat{x}\rangle - \langle f_j, x\rangle|$.

HLM Algorithm (à la "multiplicative weights")

- Start with $\hat{x}_0 =$ uniform on D.
- Update Step for $t = 0, 1..., T$:
  1. EM to get $j \approx \arg\max_j |\langle f_j, x\rangle - \langle f_j, \hat{x}_t\rangle|$
  2. Use Laplace mechanism to ask $\hat{d}_t \approx d_t = \langle f_j, x\rangle - \langle f_j, \hat{x}_t\rangle$
  3. Update $\hat{x}_{t+1}(i) = \hat{x}_t(i) \cdot e^{d_t f_j(i)/2}$
  4. Normalize $\hat{x}_{t+1}$

HLM Algorithm (à la "multiplicative weights")

- Start with $\hat{x}_0$ = uniform on D.
- Update Step for $t = 0, 1..., T$:
  1. EM to get $j \approx \arg\max_j |\langle f_j, x\rangle - \langle f_j, \hat{x}_t\rangle|$
  2. Use Laplace mechanism to ask $\hat{d}_t \approx d_t = \langle f_j, x\rangle - \langle f_j, \hat{x}_t\rangle$
  3. Update $\hat{x}_{t+1}(i) = \hat{x}_t(i) \cdot e^{d_t f_j(i)/2}$
  4. Normalize $\hat{x}_{t+1}$

Analysis Idea (following [HR'10]):

- Measure convergence of $\hat{x}_t$ to x via $\Psi_t = KL(x \| \hat{x}_t)$.

HLM Algorithm (à la "multiplicative weights")

- Start with $\hat{x}_0 =$ uniform on D.
- Update Step for $t = 0, 1..., T$:
  1. EM to get $j \approx \arg\max_j | \langle f_j, x \rangle - \langle f_j, \hat{x}_t \rangle |$
  2. Use Laplace mechanism to ask $\hat{d}_t \approx d_t = \langle f_j, x \rangle - \langle f_j, \hat{x}_t \rangle$
  3. Update $\hat{x}_{t+1}(i) = \hat{x}_t(i) \cdot e^{d_t f_j(i)/2}$
  4. Normalize $\hat{x}_{t+1}$

Analysis Idea (following [HR'10]):

- Measure convergence of $\hat{x}_t$ to x via $\Psi_t = KL(x \| \hat{x}_t)$.
- **Main utility claim**: $\Psi_t - \Psi_{t+1} \approx error(\hat{x}_t)^2/2$.

HLM Algorithm (à la "multiplicative weights")

- Start with $\hat{x}_0 =$ uniform on D.
- Update Step for $t = 0, 1..., T$:
  1. EM to get $j \approx \arg\max_j |\langle f_j, x\rangle - \langle f_j, \hat{x}_t\rangle|$
  2. Use Laplace mechanism to ask $\hat{d}_t \approx d_t = \langle f_j, x\rangle - \langle f_j, \hat{x}_t\rangle$
  3. Update $\hat{x}_{t+1}(i) = \hat{x}_t(i) \cdot e^{d_t f_j(i)/2}$
  4. Normalize $\hat{x}_{t+1}$

Analysis Idea (following [HR'10]):

- Measure convergence of $\hat{x}_t$ to x via $\Psi_t = KL(x \| \hat{x}_t)$.
- **Main utility claim**: $\Psi_t - \Psi_{t+1} \approx \mathit{error}(\hat{x}_t)^2/2$.
- As long as $\mathit{error} \geq \alpha$, can reduce KL by $\approx \alpha^2/2$

HLM Algorithm (à la "multiplicative weights")

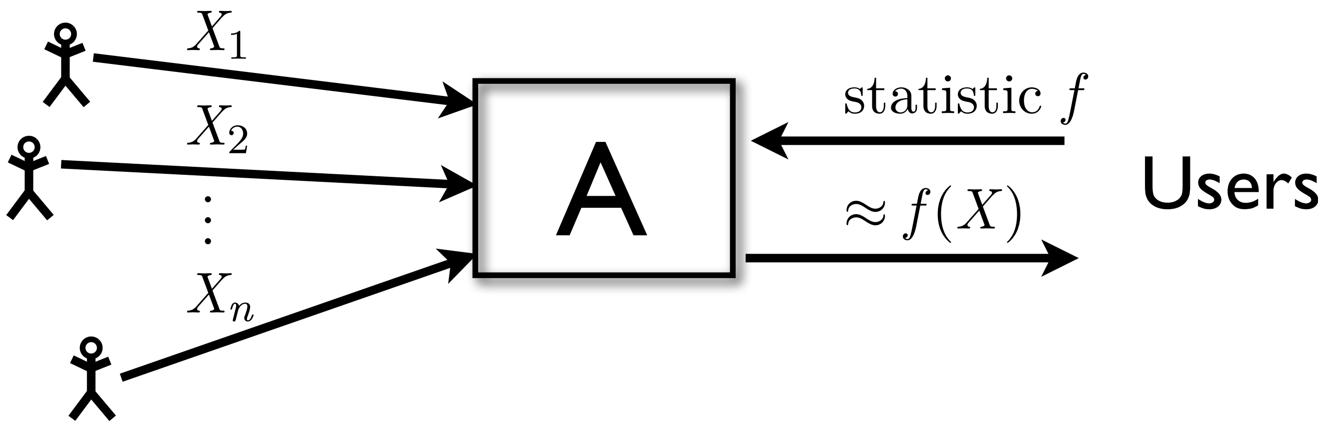
- Start with $\hat{x}_0 =$ uniform on D.
- Update Step for $t = 0, 1..., T$:
  1. EM to get $j \approx \arg\max_j |\langle f_j, x\rangle - \langle f_j, \hat{x}_t\rangle|$
  2. Use Laplace mechanism to ask $\hat{d}_t \approx d_t = \langle f_j, x\rangle - \langle f_j, \hat{x}_t\rangle$
  3. Update $\hat{x}_{t+1}(i) = \hat{x}_t(i) \cdot e^{d_t f_j(i)/2}$
  4. Normalize $\hat{x}_{t+1}$

Analysis Idea (following [HR'10]):

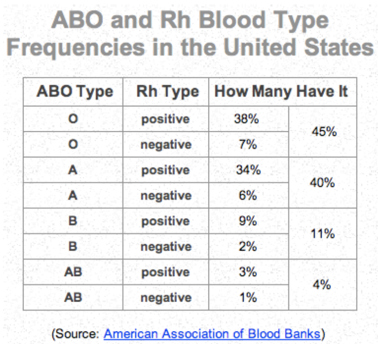
- Measure convergence of $\hat{x}_t$ to x via $\Psi_t = KL(x \| \hat{x}_t)$.
- **Main utility claim**: $\Psi_t - \Psi_{t+1} \approx error(\hat{x}_t)^2/2$.
- As long as $error \geq \alpha$, can reduce KL by $\approx \alpha^2/2$
- Since $KL(x \| \hat{x}_0) \leq \log |D|$, error drops below α after $\frac{\log |D|}{\alpha^2}$ updates.

Local and Smooth Sensitivity

Concrete Problem: Parametric Estimators



A statistic or estimator is a function $f : (\text{data sets}) \to \mathbb{R}^p$, *e.g.*

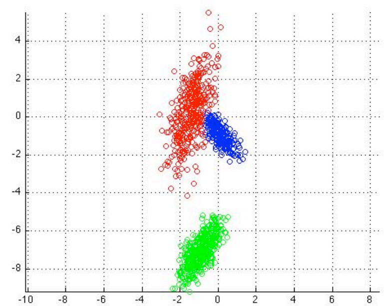


ABO and Rh Blood Type Frequencies in the United States

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%	

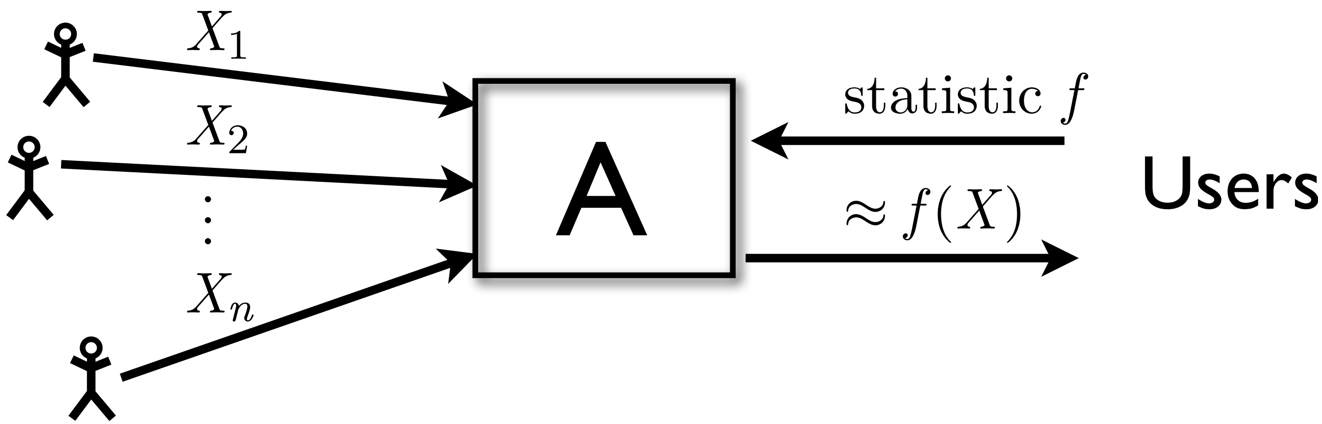
(Source: American Association of Blood Banks)

Contingency table



Fitted parameters of mixture of gaussians

Concrete Problem: Parametric Estimators



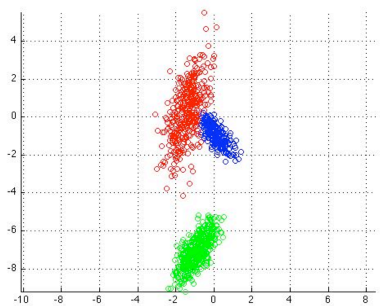
A statistic or estimator is a function f : (data sets) $\rightarrow \mathbb{R}^p$, *e.g.*

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)

Contingency table

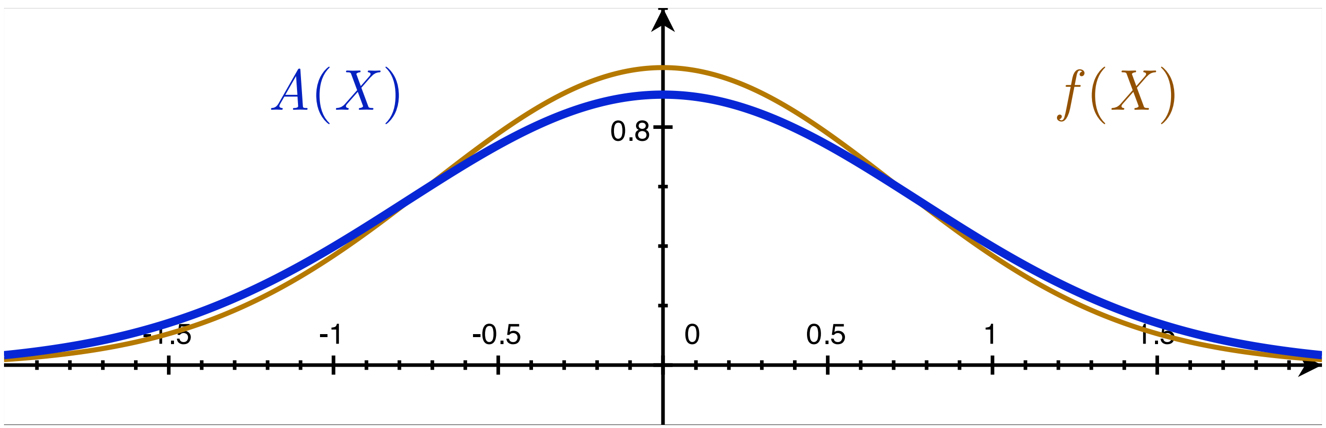


Fitted parameters of mixture of gaussians

Goal: differentially private approximation to f.

Use the Laplace Mechanism?

- Recall: $A(X) = f(X) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$
  - Global sensitivity GS_f measures how much f varies when one data point changes
- Works well for proportions
  - Private statistic has nearly same distribution as true statistic
- For which statistics is this possible?



Asymptotically Normal Statistics

For many statistics f and distributions P, we know:
If $X = X_1, ..., X_n$ is drawn i.i.d. from P, **then**

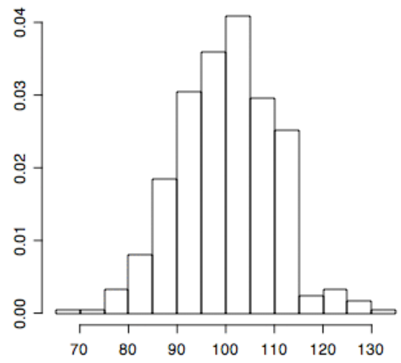
$$f(X) \approx \text{(normal random variable)}$$

Asymptotically Normal Statistics

For many statistics f and distributions P, we know:
If $X = X_1, ..., X_n$ is drawn i.i.d. from P, **then**

$$f(X) \approx \text{(normal random variable)}$$

- Sums & averages (Central Limit Theorem)

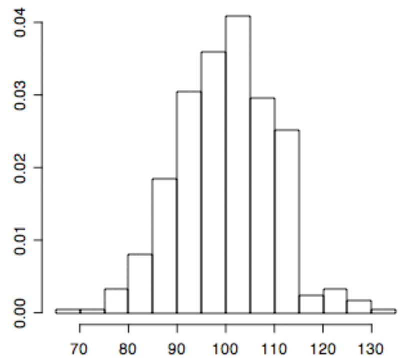


Asymptotically Normal Statistics

For many statistics f and distributions P, we know:
If $X = X_1, ..., X_n$ is drawn i.i.d. from P, **then**

$$f(X) \approx (\text{normal random variable})$$

- Sums & averages (Central Limit Theorem)
- Maximum likelihood estimators
- Regression parameters: linear and logistic regression, SVM

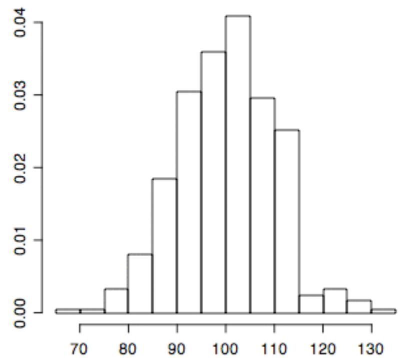


Asymptotically Normal Statistics

For many statistics f and distributions P, we know:
If $X = X_1, ..., X_n$ is drawn i.i.d. from P, **then**

$$f(X) \approx \text{(normal random variable)}$$

- Sums & averages (Central Limit Theorem)
- Maximum likelihood estimators
- Regression parameters: linear and logistic regression, SVM



- "M-estimators"

A General Result

Theorem [S., '11]

For every f : (data sets) $\to \mathbb{R}^p$ and $\varepsilon > 0$,
there exists a ε-diffe.p. algorithm A such that

$$A(X) \approx f(X) \text{ as } n \text{ grows}$$

whenever* $X \sim P^n$ and f is asymptotically normal at P.

A General Result

Theorem [S., '11]

For every f : (data sets) $\to \mathbb{R}^p$ and $\varepsilon > 0$,
there exists a ε-diffe.p. algorithm A such that

$$A(X) \approx f(X) \text{ as } n \text{ grows}$$

whenever* $X \sim P^n$ and f is asymptotically normal at P.

* Some conditions (on bias and third moment) apply.

A General Result

Theorem [S., '11]

For every f : (data sets) $\to \mathbb{R}^p$ and $\varepsilon > 0$,
there exists a ε-diffe.p. algorithm A such that

$$A(X) \approx f(X) \text{ as } n \text{ grows}$$

whenever* $X \sim P^n$ and f is asymptotically normal at P.

* Some conditions (on bias and third moment) apply.

Consequence: estimators with optimal rate $1/\sqrt{n}$ for

- sample mean
- sample median
- maximum likelihood estimator for nice models
- regression coefficients

A General Result

Theorem [S., '11]

For every f : (data sets) $\to \mathbb{R}^p$ and $\varepsilon > 0$,
there exists a ε-diffe.p. algorithm A such that

$$A(X) \approx f(X) \text{ as } n \text{ grows}$$

whenever* $X \sim P^n$ and f is asymptotically normal at P.

- The transformation from f to A is (almost) black box.
  - No need to "understand" structure of f.

A General Result

Theorem [S., '11]

For every f : (data sets) $\to \mathbb{R}^p$ and $\varepsilon > 0$,
there exists a ε-diffe.p. algorithm A such that

$$A(X) \approx f(X) \text{ as } n \text{ grows}$$

whenever* $X \sim P^n$ and f is asymptotically normal at P.

- The transformation from f to A is (almost) black box.
  - No need to "understand" structure of f.

Free lunch!

A General Result

Theorem [S., '11]

For every f : (data sets) $\rightarrow \mathbb{R}^p$ and $\varepsilon > 0$,
there exists a ε-diffe.p. algorithm A such that

$$A(X) \approx f(X) \text{ as } n \text{ grows}$$

whenever* $X \sim P^n$ and f is asymptotically normal at P.

- The transformation from f to A is (almost) black box.
  - No need to "understand" structure of f.

~~Free lunch!~~

- Caveat: Performance degrades with dimension p and privacy parameter ε.
  - Result holds for $p < n^c$ for constant $c \approx 1/6$.
  - Reconstruction attacks imply some degradation is necessary.

Previous Work

Theorem [S., '11]

For every f : (data sets) $\to \mathbb{R}^p$ and $\varepsilon > 0$,
there exists a ε-diffe.p. algorithm A such that

$$A(X) \approx f(X) \text{ as } n \text{ grows}$$

whenever* $X \sim P^n$ and f is asymptotically normal at P.

Relative to previous work, we contribute:

- Generality, simplicity (previous aproaches were problem-specific)
- Improved convergence guarantees for order statistics and linear regression ($O(n^{\frac{1}{2}})$ versus $O(n^{\frac{1}{2}+\gamma})$ [DL'09]).

Technique: Sample and aggregate

Why Not Laplace Mechanism?

Why not release

$$A(X) = f(X) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right) \quad ?$$

Why Not Laplace Mechanism?

Why not release

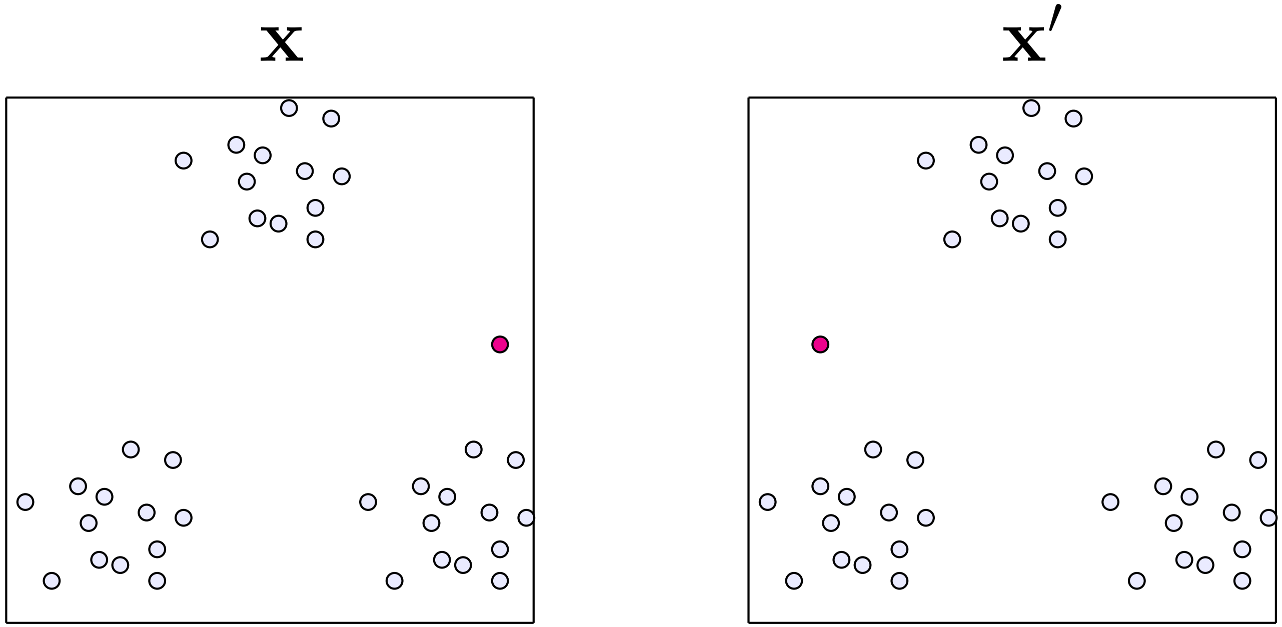
$$A(X) = f(X) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right) \quad ?$$

- Need to **understand** f
  - trusted code?
  - new functions every day...
- Global sensitivity can be too high

High global sensitivity

Example: fitting a mixture of two Gaussians

Database entries: points in a the plane.

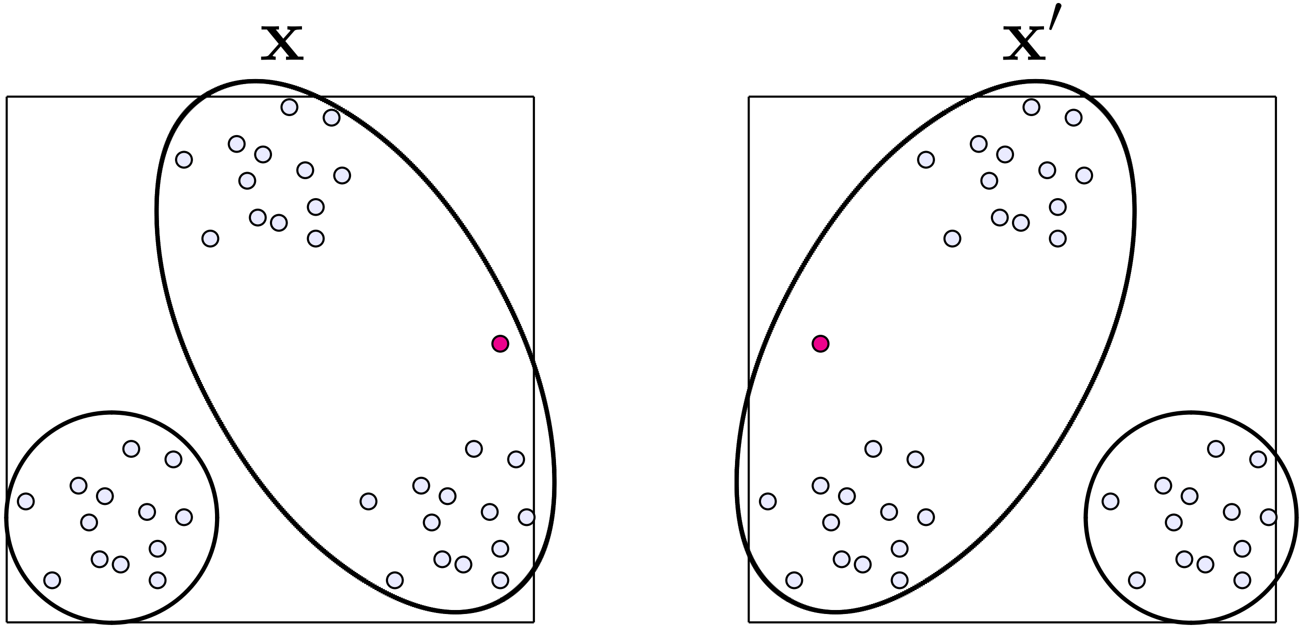


Global sensitivity of component means is roughly the diameter of the space.

High global sensitivity

Example: fitting a mixture of two Gaussians

Database entries: points in a the plane.

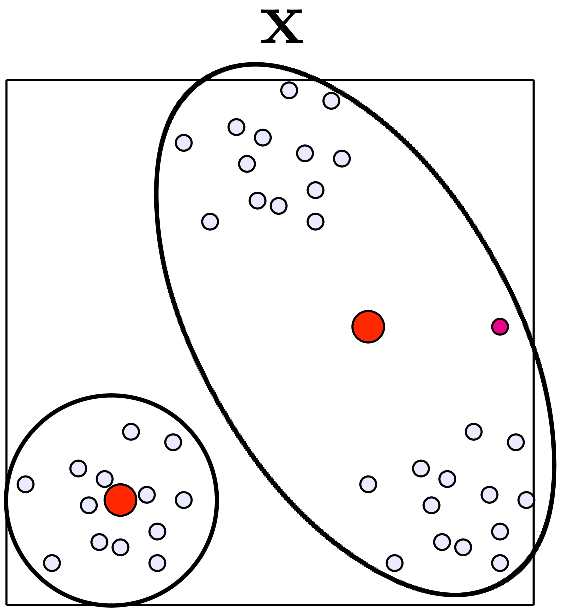


Global sensitivity of component means is roughly the diameter of the space.

High global sensitivity

Example: fitting a mixture of two Gaussians

Database entries: points in a the plane.

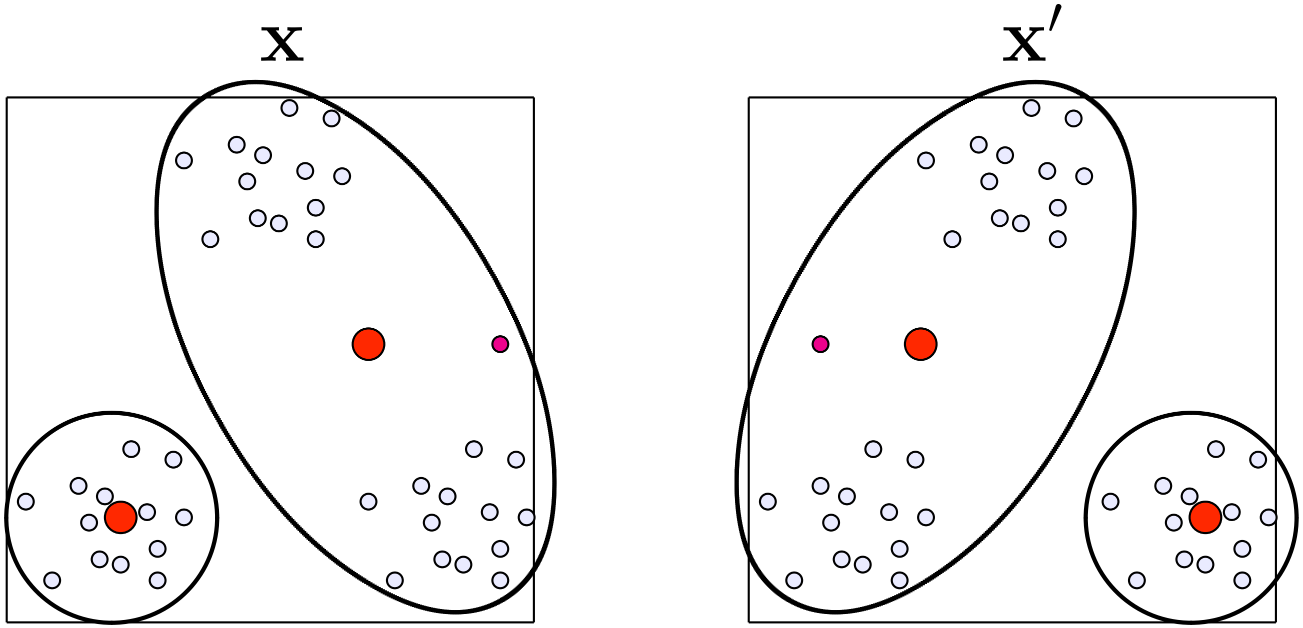


Global sensitivity of component means is roughly the diameter of the space.

High global sensitivity

Example: fitting a mixture of two Gaussians

Database entries: points in a the plane.



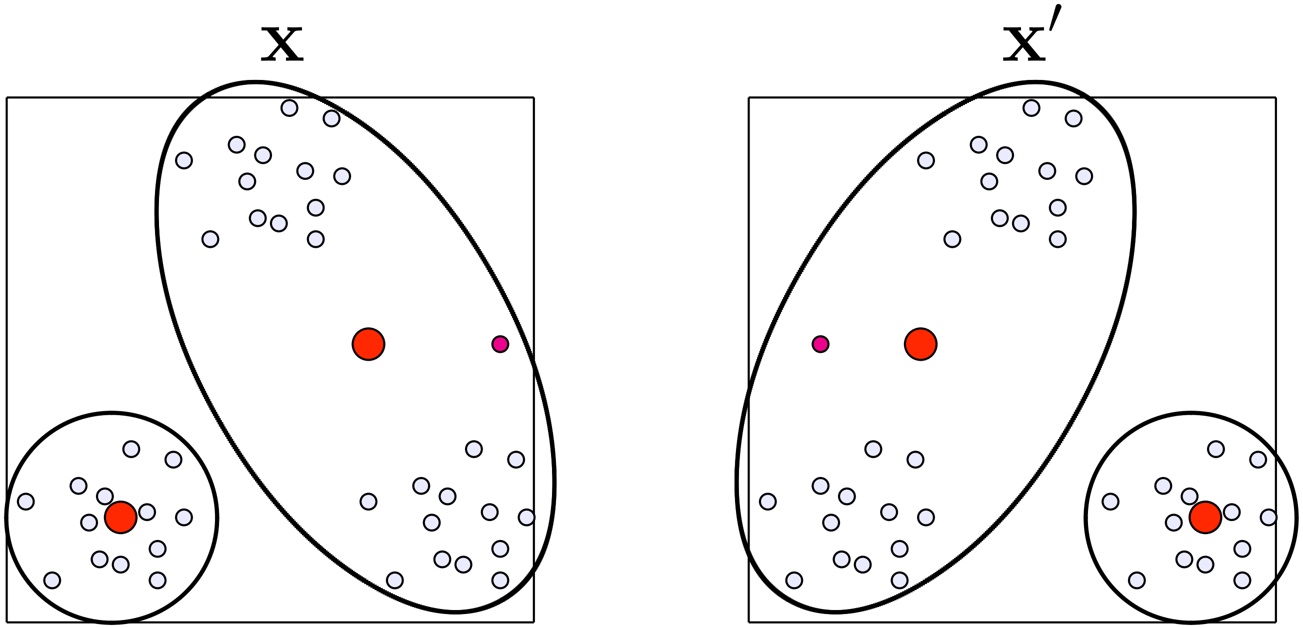
Global sensitivity of component means is roughly the diameter of the space.

- If clustering is "good", means should be insensitive.

High global sensitivity

Example: fitting a mixture of two Gaussians

Database entries: points in a the plane.



Global sensitivity of component means is roughly the diameter of the space.

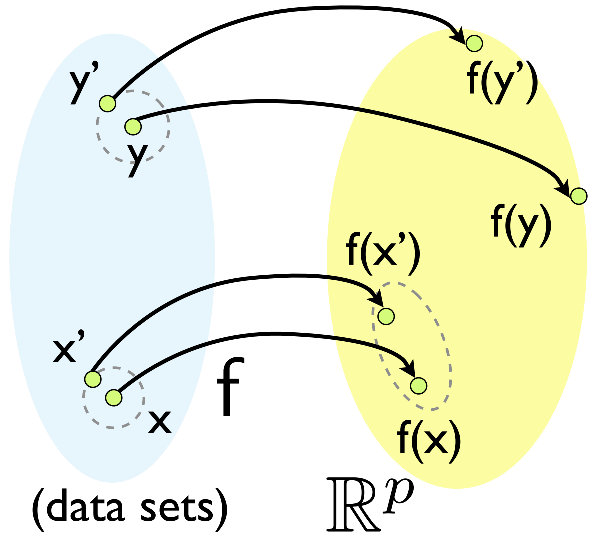
- If clustering is "good", means should be insensitive.
- [Nissim, Raskhodnikova, S'07]: add less noise to "nice" data

Getting Around High Global Sensitivity *[NRS'07]*

Local sensitivity of f at x: how much does f vary among neighbors of x?

$$\mathsf{LS}_f(x) = \max_{x' \text{ neighbor of } x} \|f(x) - f(x')\|_2$$

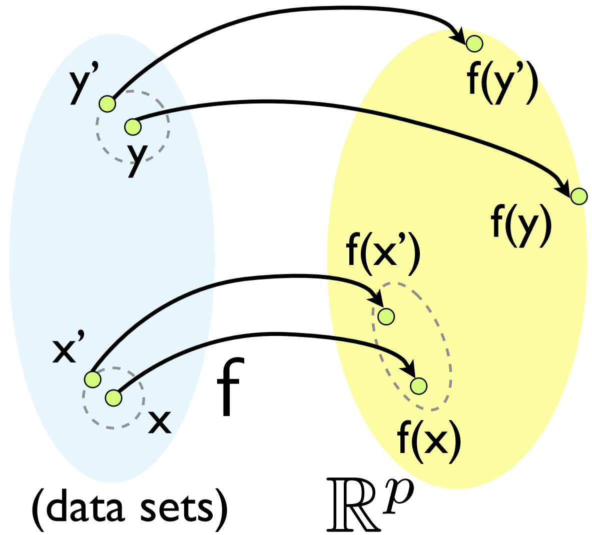
[NRS'07] Goal: add noise proportional to local sensitivity.



Getting Around High Global Sensitivity [NRS'07]

Local sensitivity of f at x: how much does f vary among neighbors of x?

$$\mathsf{LS}_f(x) = \max_{x' \text{ neighbor of } x} \|f(x) - f(x')\|_2$$



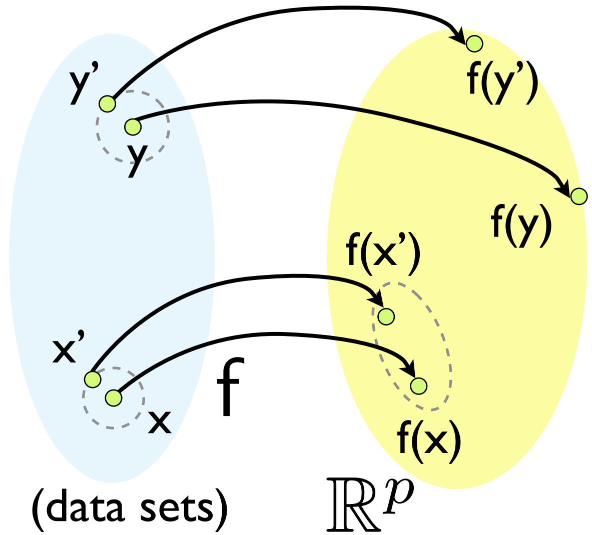
[NRS'07] Goal: add noise proportional to local sensitivity.

- Problem: Using local sensitivity is not private (noise leaks)

Getting Around High Global Sensitivity [NRS'07]

Local sensitivity of f at x: how much does f vary among neighbors of x?

$$\mathsf{LS}_f(x) = \max_{x' \text{ neighbor of } x} \|f(x) - f(x')\|_2$$



[NRS'07] Goal: add noise proportional to local sensitivity.

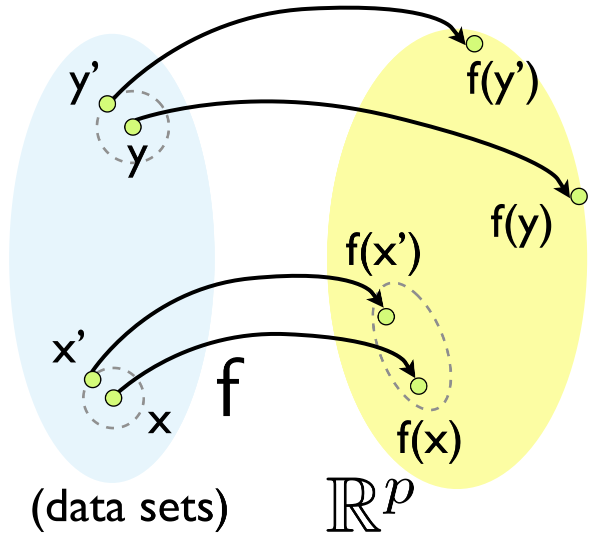
- Problem: Using local sensitivity is not private (noise leaks)
- Solution 1: Use **smoothed** local sensitivity
    - Order statistics (median, quantiles, ...)
    - Stats for social networks (MST cost, subgraph frequencies) [Karwa, Rashodnikova, Yaroslavtsev, S, '11]
    - Problem: often computationally difficult

Getting Around High Global Sensitivity [NRS'07]

Local sensitivity of f at x: how much does f vary among neighbors of x?

$$\mathsf{LS}_f(x) = \max_{x' \text{ neighbor of } x} \|f(x) - f(x')\|_2$$

[NRS'07] Goal: add noise proportional to local sensitivity.



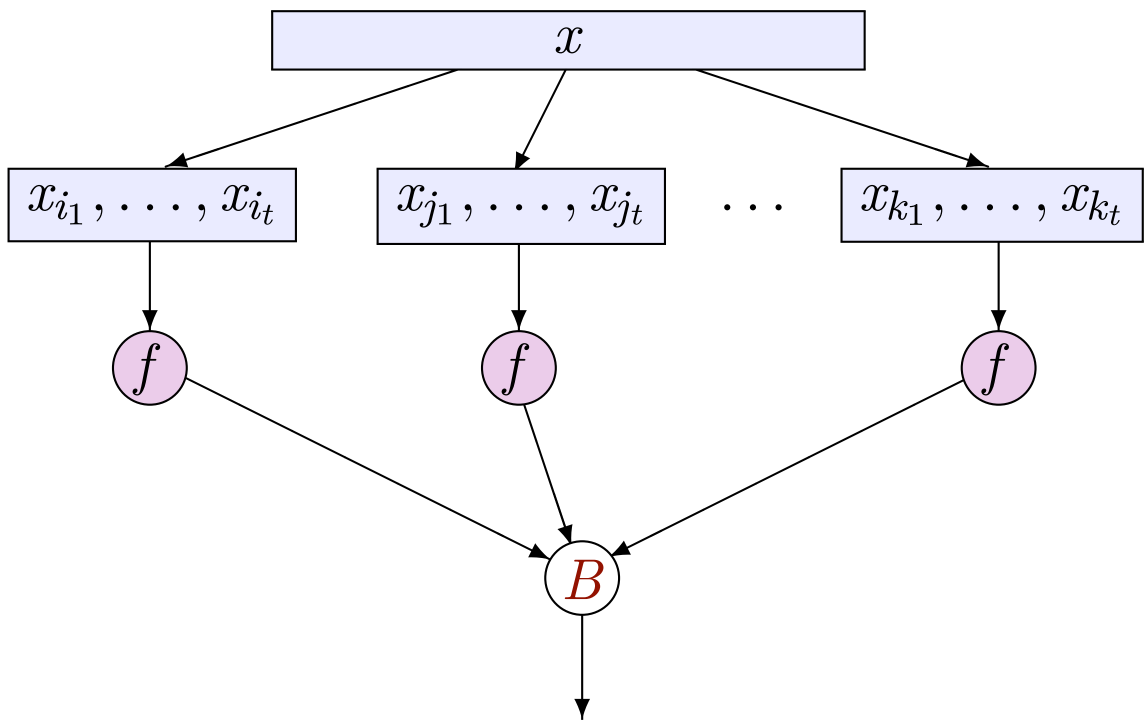
- Problem: Using local sensitivity is not private (noise leaks)
- Solution 1: Use **smoothed** local sensitivity
  - Order statistics (median, quantiles, ...)
  - Stats for social networks (MST cost, subgraph frequencies) [Karwa, Rashodnikova, Yaroslavtsev, S, '11]
  - Problem: often computationally difficult
- Solution 2: "Sample and aggregate"

Sample-and-Aggregate Framework [NRS'07]

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

- Break x into k samples of n/k points
- Compute f on each block
- Run differentially private algorithm B:

$$\tilde{f}(x) = B\big(f(block_1), f(block_2), \ldots, f(block_k)\big)$$

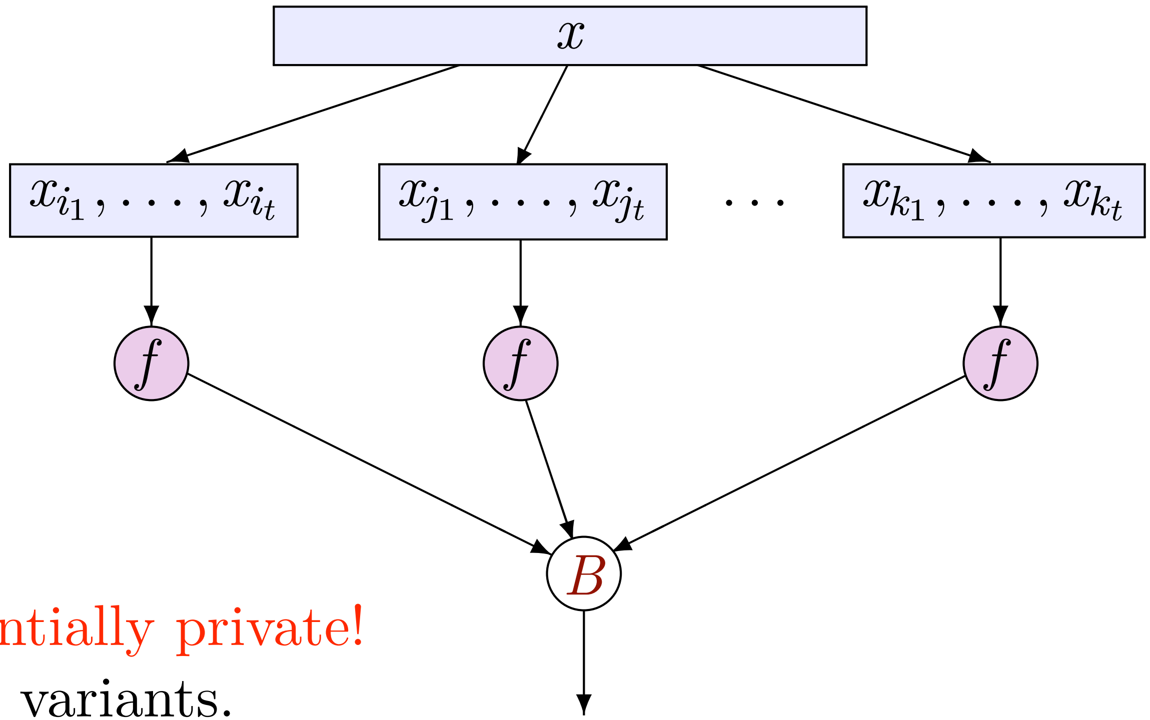


Sample-and-Aggregate Framework [NRS'07]

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

- Break x into k samples of n/k points
- Compute f on each block
- Run differentially private algorithm B:

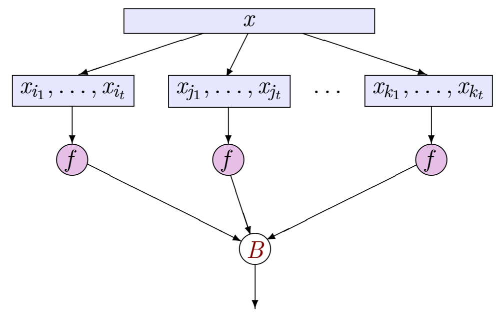
$$\tilde{f}(x) = B(f(block_1), f(block_2), \dots, f(block_k))$$



Always differentially private!

Many possible variants.

Application 1: Normal Statistics



- Suppose f is asymptotically normal at x.
- If block length $\frac{n}{k}$ large enough, then

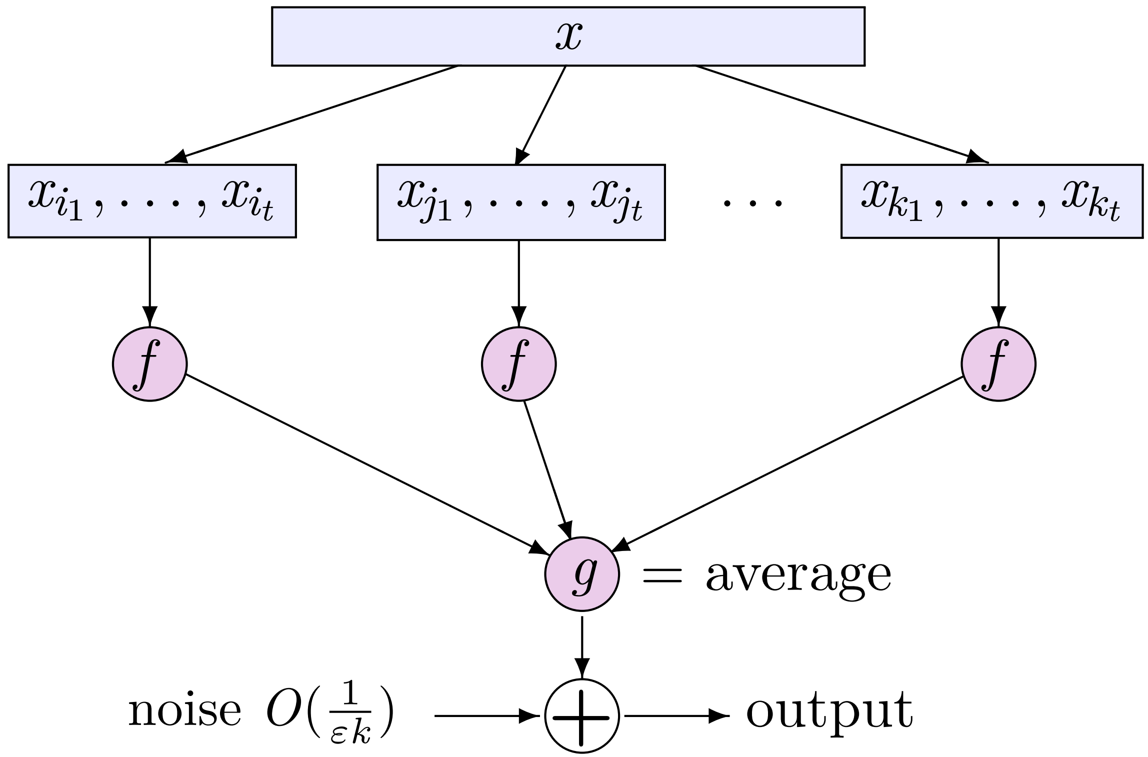
$$f(block_1), f(block_2), \ldots, f(block_k) \approx \text{normal}.$$

- Design aggregation B for estimating mean of approximately normal random variables.
  - One aggregation works for all asymptotically normal random variables.
  - Getting optimal noise requires extra insight into bias/variance tradeoff

Toy variant: Averaging

Suppose $Range(f) \subseteq [0, 1]$

- Randomly break x into k samples of n/k points
- $\tilde{f}(x) = \text{avg}(f(block_1), f(block_2), \ldots, f(block_k))$
- Output $\tilde{f}(x) + \mathsf{Lap}(\frac{1}{k\varepsilon})$.

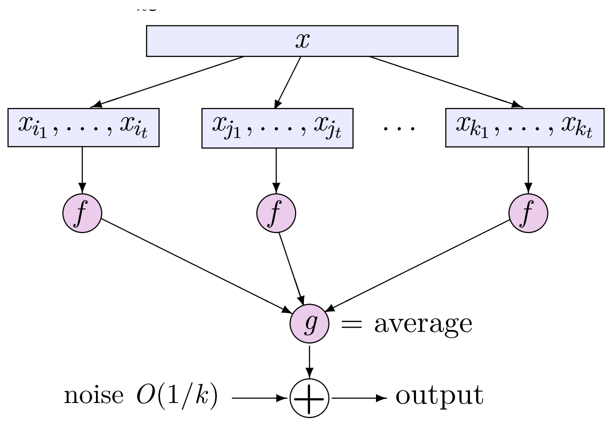


Toy variant: Averaging

Why is this useful?

- If most samples give roughly the same answer, get

$$\text{(that answer)} \pm \underbrace{O\left(\frac{1}{\varepsilon k}\right)}_{\text{added noise}}$$



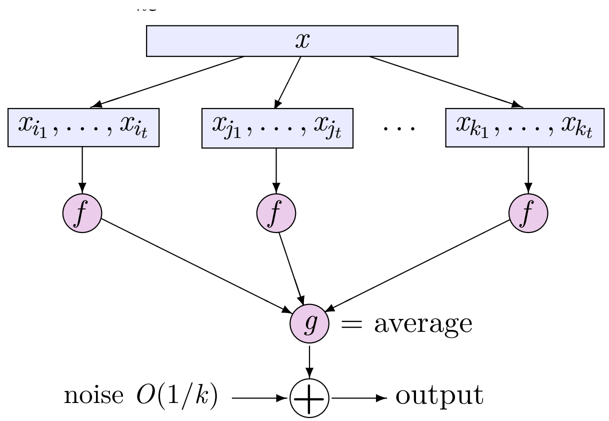
  - Not garbage!
  - But do we only get the "quality" of n/k samples?
  - How to choose k?

Toy variant: Averaging

Why is this useful?

- If most samples give roughly the same answer, get

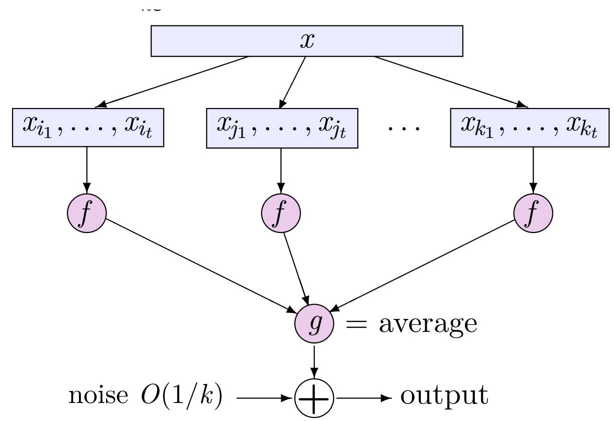
$$(\text{that answer}) \pm \underbrace{O\left(\frac{1}{\varepsilon k}\right)}_{\text{added noise}}$$



  - Not garbage!
  - But do we only get the "quality" of n/k samples?
  - How to choose k?
- [NRS'07] Generic aggregator, works for many types of data

Toy variant: Averaging

Why is this useful?



- If most samples give roughly the same answer, get

$$\text{(that answer)} \pm \underbrace{O\left(\frac{1}{\varepsilon k}\right)}_{\text{added noise}}$$

  - Not garbage!
  - But do we only get the "quality" of n/k samples?
  - How to choose k?
- [NRS'07] Generic aggregator, works for many types of data
- [S. '11] Tighter results for normal statistics
  - Take advantage of low bias of typical estimators
  - Roughly: get the "quality" of all n points

Application 2: Sparse Regression *[Kifer,S,Thakurta '12]*

$$\text{Given: } \mathbf{X} = \underbrace{\begin{pmatrix} ----- & x_1 & ----- \\ & \vdots & \\ ----- & x_i & ----- \\ & \vdots & \\ ----- & x_n & ----- \end{pmatrix}}_{p \text{ ``features''}} \text{ and } \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}$$

Application 2: Sparse Regression [Kifer,S,Thakurta '12]

$$\text{Given: } \mathbf{X} = \underbrace{\begin{pmatrix} ------ & x_1 & ------ \\ & \vdots & \\ ------ & \color{red}{x_i} & ------ \\ & \vdots & \\ ------ & x_n & ------ \end{pmatrix}}_{p \text{ ``features''}} \text{ and } \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ \color{red}{y_i} \\ \vdots \\ y_n \end{pmatrix}$$

Application 2: Sparse Regression *[Kifer,S,Thakurta '12]*

$$\text{Given: } \mathbf{X} = \underbrace{\begin{pmatrix} ------ & x_1 & ------ \\ & \vdots & \\ ------ & x_i & ------ \\ & \vdots & \\ ------ & x_n & ------ \end{pmatrix}}_{p \text{ "features"}} \text{ and } \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}$$

Linear Regression: find $\vec{\theta}$ such that $\mathbf{X}\vec{\theta} \approx \vec{y}$

Application 2: Sparse Regression [Kifer,S,Thakurta '12]

$$\text{Given: } \mathbf{X} = \underbrace{\begin{pmatrix} ----- & x_1 & ----- \\ & \vdots & \\ ----- & x_i & ----- \\ & \vdots & \\ ----- & x_n & ----- \end{pmatrix}}_{p \text{ "features"}} \text{ and } \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}$$

Sparse Linear Regression: find $\vec{\theta}$ such that $\mathbf{X}\vec{\theta} \approx \vec{y}$
and $\vec{\theta}$ has at most s nonzero entries.

Application 2: Sparse Regression *[Kifer,S,Thakurta '12]*

$$\text{Given: } \mathbf{X} = \underbrace{\begin{pmatrix} ------ & x_1 & ------ \\ & \vdots & \\ ------ & x_i & ------ \\ & \vdots & \\ ------ & x_n & ------ \end{pmatrix}}_{p \text{ ``features''}} \text{ and } \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}$$

Sparse Linear Regression: find $\vec{\theta}$ such that $\mathbf{X}\vec{\theta} \approx \vec{y}$
and $\vec{\theta}$ has at most s nonzero entries.

Typical setting: $p \gg n$.

- Solvable nonprivately roughly when $n \gg s \log p$

Application 2: Sparse Regression *[Kifer,S,Thakurta '12]*

$$\text{Given: } \mathbf{X} = \underbrace{\begin{pmatrix} ------ & x_1 & ------ \\ & \vdots & \\ ------ & x_i & ------ \\ & \vdots & \\ ------ & x_n & ------ \end{pmatrix}}_{p \text{ ``features''}} \text{ and } \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}$$

Sparse Linear Regression: find $\vec{\theta}$ such that $\mathbf{X}\vec{\theta} \approx \vec{y}$
and $\vec{\theta}$ has at most s nonzero entries.

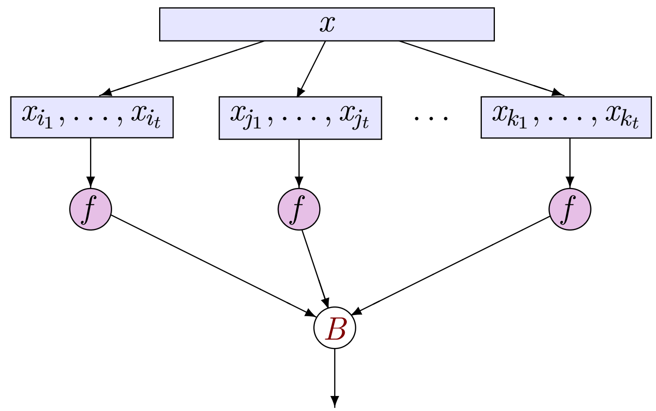
Typical setting: $p \gg n$.

- Solvable nonprivately roughly when $n \gg s \log p$
- Private algorithm?
  - Noise addition fails because of high dimension (noise p/n per coefficient)

Application 2: Sparse Regression [Kifer,S,Thakurta '12]

[KST'12]

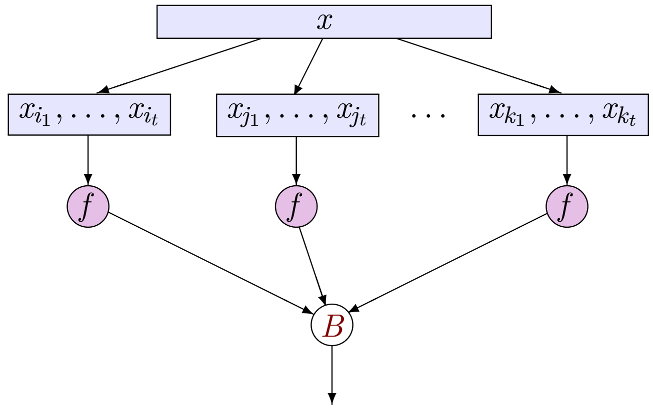
- Use sample and aggregate to find relevant features.
- Apply previous algorithms on those features



Application 2: Sparse Regression *[Kifer,S,Thakurta '12]*

[KST'12]

- Use sample and aggregate to find relevant features.
- Apply previous algorithms on those features



In each block:

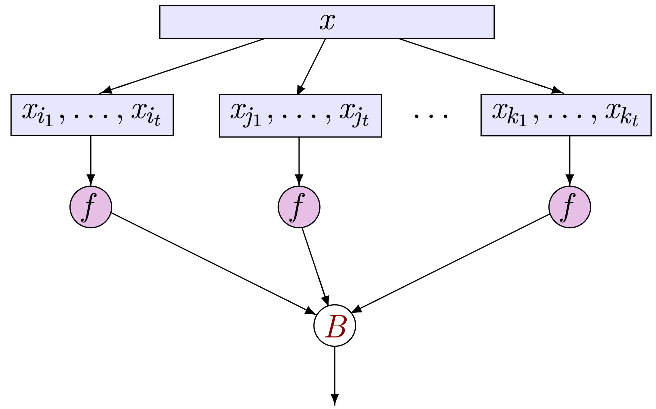
- Run nonprivate algorithm to get candidate list of s features

Aggregation: Privately choose features selected most often

Application 2: Sparse Regression [Kifer, S, Thakurta '12]

[KST'12]

- Use sample and aggregate to find relevant features.
- Apply previous algorithms on those features



In each block:

- Run nonprivate algorithm to get candidate list of s features

Aggregation: Privately choose features selected most often

- Use "exponential sampling" [McSherry, Talwar '07, Bhaskar, Laxman, S, Thakurta '10].

  Sample s features randomly, where

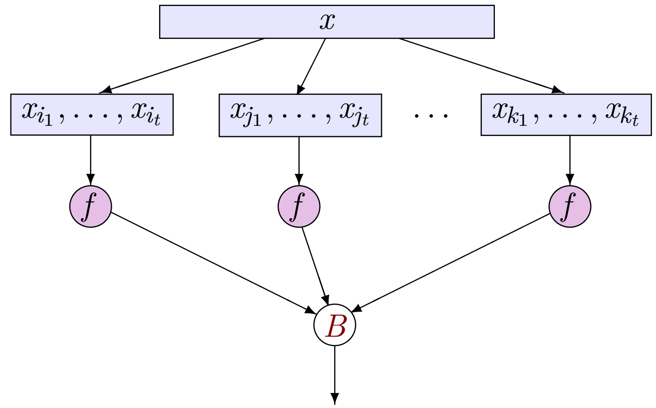
$$\Pr(i) \propto \exp(\varepsilon \cdot (\#\text{ blocks where } i \text{ was selected})).$$

- Produces good estimates when $n \gg s^2 \log p$.
- Open question: match nonprivate bound

Sample-and-aggregate

Two applications:

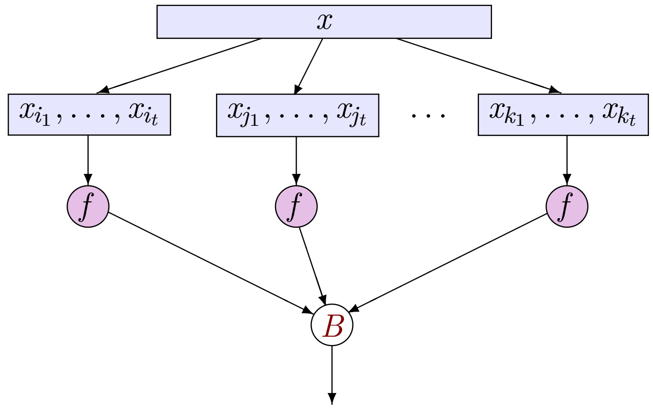
- Asymptotically normal statistics
- Sparse regression



Sample-and-aggregate

Two applications:

- Asymptotically normal statistics
- Sparse regression



Produces algorithms with interesting properties, regardless of privacy

- Stability: robust to small changes in input
  - Guarantees good generalization error
  - Deterministic stable sparse learning impossible [Xu *et al.*,'11]
- Streaming: algorithms require little space ($\approx \sqrt{n}$)
  - Useful for very large data sets

Implemented by [Moharan et al., *SIGMOD* 2012]

Postscript:
Systems and Implementation

Differential Privacy in "Practice"

- Currently, differential private algorithms hard to use
  - ➢ noise
  - ➢ can't use out-of-the-box software
  - ➢ requires fresh thinking for each new problem, etc
- Several systems to make use easier
  - ➢ [McSherry'09] PINQ: variation on LINQ with differential privacy enforced by query mechanism
  - ➢ [Haeberlen et al. '11] Programming language with privacy enforced by type system
  - ➢ [Roy et al. '10, Moharan et al. '12] Systems for restricted classes of queries, focus usability with legacy code
- Hard to get right!
  - ➢ [Haeberlen et al. '11] Timing attacks
  - ➢ [Mironov '12] Leakage via numerical errors

A play in three acts

A play in three acts

- **Act I: Attacks**
    - (Why is privacy hard?)
    - Reconstruction attacks

A play in three acts

- **Act I: Attacks**
  - (Why is privacy hard?)
  - Reconstruction attacks
- **Act II: Definitions**
  - One approach: "differential" privacy
  - Variations on the theme

A play in three acts

- **Act I: Attacks**
  - (Why is privacy hard?)
  - Reconstruction attacks
- **Act II: Definitions**
  - One approach: "differential" privacy
  - Variations on the theme
- **Act III: Algorithms**
  - Basic techniques: noise addition, exponential sampling
  - Exploiting "local" sensitivity
  - Answering many queries

Things I did not cover

- Multiparty models
  - ➢ What if data are distributed?
- Computational considerations
  - ➢ "Require" distributed models to exploit
- Graph data
  - ➢ Hard to pin down which data are "mine"
- Information-theoretic definitions
- Lower bounds specific to differential privacy
- And More!

Conclusions

Conclusions

- Define privacy in terms of my effect on output
  - Meaningful despite arbitrary external information
  - I should participate if I get benefit

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 with rigorous guarantees?
  - ➢ Basic Tools
  - ➢ More advanced examples

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 with rigorous guarantees?
  - Basic Tools
  - More advanced examples
- Future work
  - Other definitions: How can we exploit uncertainty?
  - Applications: genetics, finance, ...
  - How can we reason about privacy, more broadly?

Further resources

- Aaron Roth's lecture notes
  - http://www.cis.upenn.edu/~aaroth/courses/privacyF11.html
- 2010 course by Sofya Raskhodnikova and me
  - http://www.cse.psu.edu/~asmith/privacy598
- DIMACS Workshop on Data Privacy
  - October 24-26, 2012 (immediately after FOCS)
  - http://dimacs.rutgers.edu/Workshops/DifferentialPrivacy/