Privacy Preserving Data Mining

Cynthia Dwork and Frank McSherry

In collaboration with: Ilya Mironov and Kunal Talwar

Interns: S. Chawla, K. Kenthapadi, A. Smith, H. Wee

Visitors: A. Blum, P. Harsha, M. Naor, K. Nissim, M. Sudan

Intentionally Blank Slide

Data Mining: Privacy v. Utility

Motivation: Inherent tension in mining sensitive databases:

We want to release **aggregate** information about the data, without leaking **individual** information about participants.

- Aggregate info: Number of A students in a school district.
- Individual info: If a particular student is an A student.

Data Mining: Privacy v. Utility

Motivation: Inherent tension in mining sensitive databases:

We want to release **aggregate** information about the data, without leaking **individual** information about participants.

- Aggregate info: Number of A students in a school district.
- Individual info: If a particular student is an A student.

Problem: Exact aggregate info may leak individual info. Eg:

Number of A students in district, and
Number of A students in district not named Frank McSherry.

Data Mining: Privacy v. Utility

Motivation: Inherent tension in mining sensitive databases:

We want to release **aggregate** information about the data, without leaking **individual** information about participants.

- Aggregate info: Number of A students in a school district.
- Individual info: If a particular student is an A student.

Problem: Exact aggregate info may leak individual info. Eg:

Number of A students in district, and
Number of A students in district not named Frank McSherry.

Goal: Method to protect individual info, release aggregate info.

What's New Here?

Common Question: Hasn't this problem been studied before?

1. Census Bureau has privacy methods. Ad hoc, ill-understood.

2. DB interest recently rekindled, but weak results / definitions.

3. Standard cryptography does not solve the problem either. Information is leaked through correct answers.

What's New Here?

Common Question: Hasn't this problem been studied before?

1. Census Bureau has privacy methods. Ad hoc, ill-understood.

2. DB interest recently rekindled, but weak results / definitions.

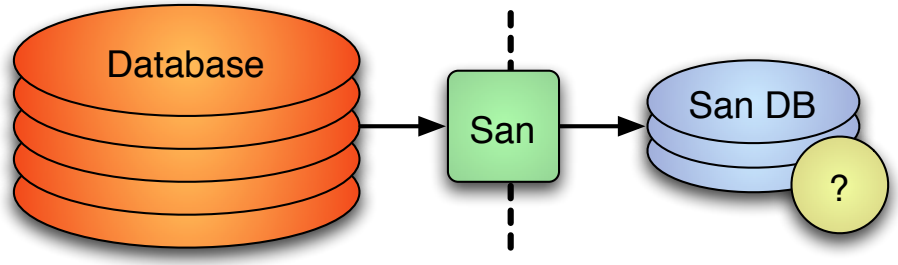
3. Standard cryptography does not solve the problem either. Information is leaked through correct answers.

This Work: Cryptographic rigor applied to private data mining.

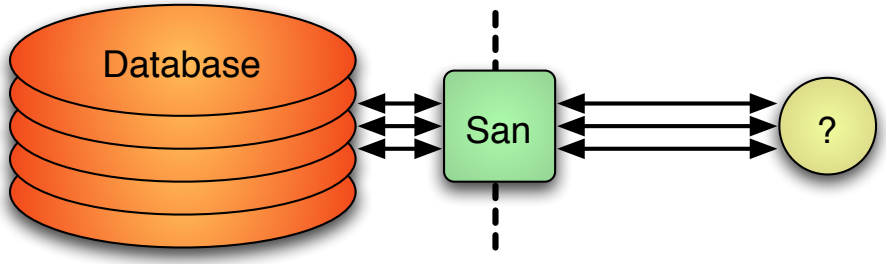
1. **Provably** strong protection of individual information.
2. Release of very accurate aggregate information.

Two Privacy Models

1. **Non-interactive**: Database is sanitized and released.



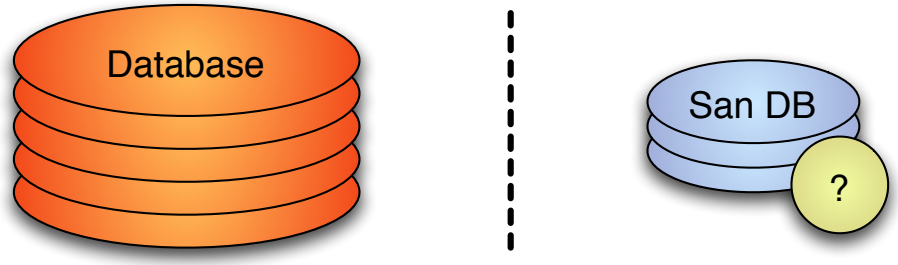
2. **Interactive**: Multiple questions asked / answered adaptively.



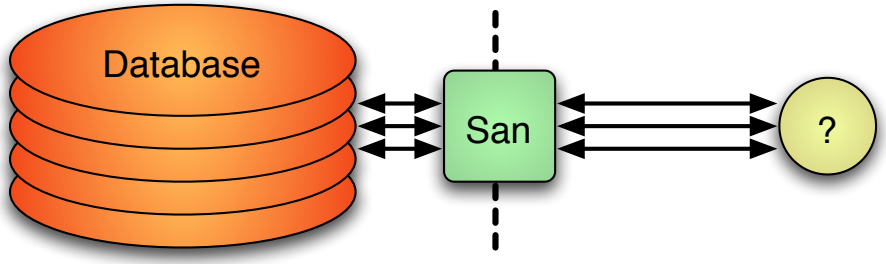
We will focus on the interactive model in this talk.

Two Privacy Models

1. **Non-interactive**: Database is sanitized and released.



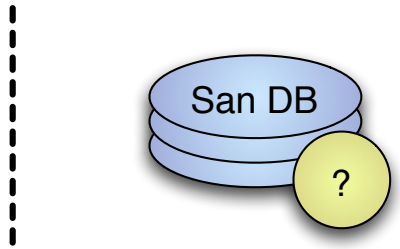
2. **Interactive**: Multiple questions asked / answered adaptively.



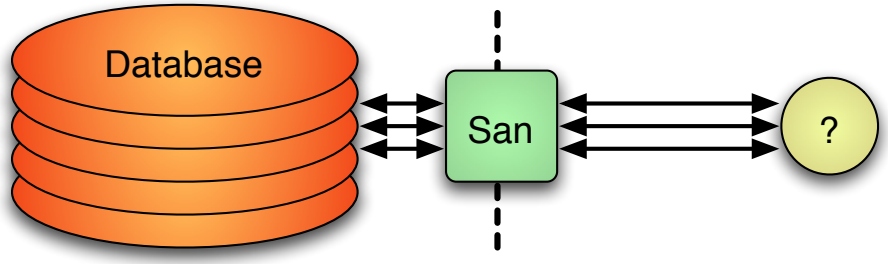
We will focus on the interactive model in this talk.

Two Privacy Models

1. **Non-interactive**: Database is sanitized and released.



2. **Interactive**: Multiple questions asked / answered adaptively.

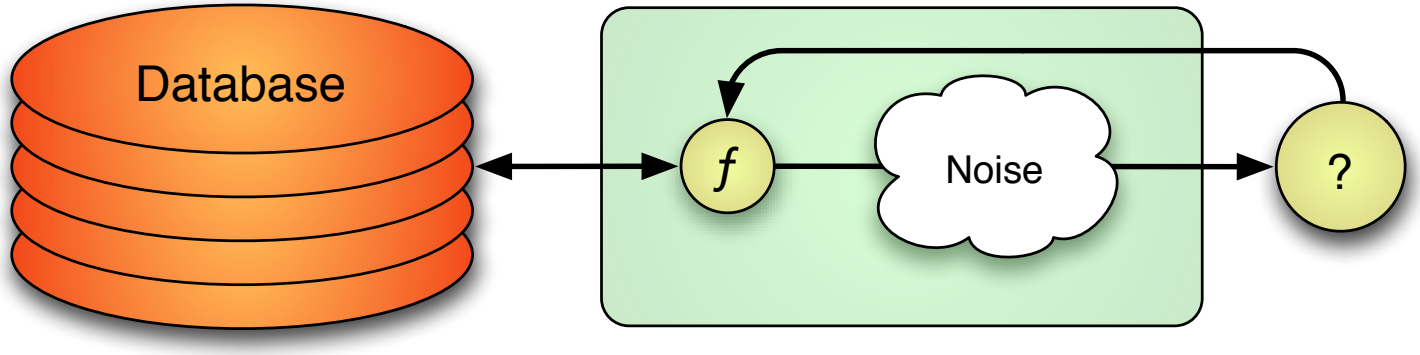


We will focus on the interactive model in this talk.

An Interactive Sanitizer: $\mathcal{K}_f$

$\mathcal{K}_f$ applies query function f to database, and returns noisy result.

$$\mathcal{K}_f(\text{DB}) \equiv f(\text{DB}) + \text{Noise}$$



Adding **random** noise introduces uncertainty, and thus privacy.

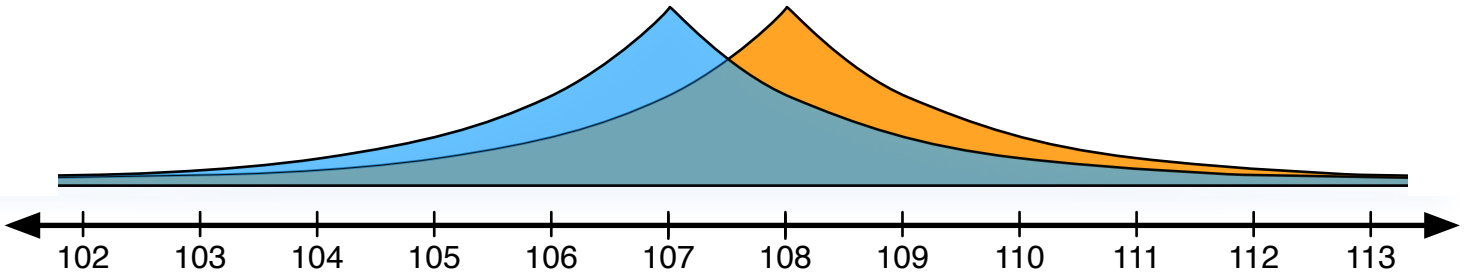
Important: The amount of noise, and privacy, is configurable. Determined by a privacy parameter ϵ and the query function f.

Differential Privacy

Privacy Concern: Joining the database leads to a bad event.

Strong Privacy Goal: Joining the database should not substantially increase or decrease the probability of *any* event happening.

Consider the distributions $\mathcal{K}_f(\text{DB} - \text{Me})$ and $\mathcal{K}_f(\text{DB} + \text{Me})$:



Q: Is any response much more likely under one than the other?

If not, then all events are just as likely now as they were before. Any behavior based on the output is just as likely now as before.

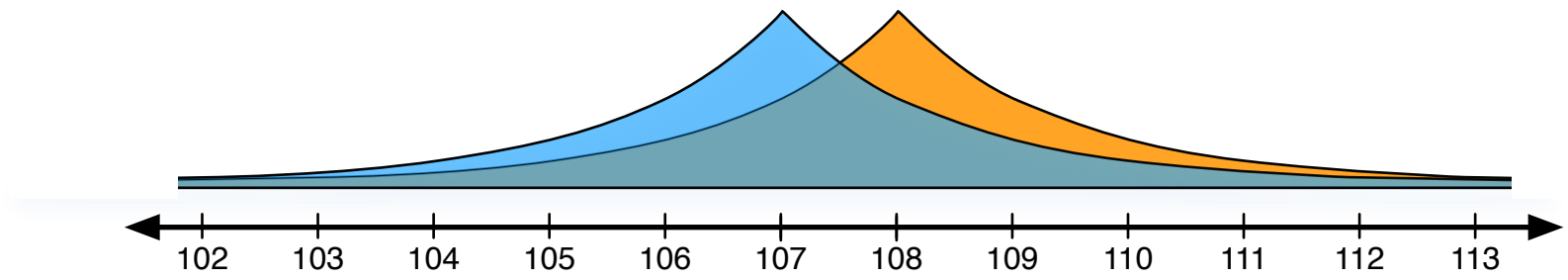
Differential Privacy

Definition

We say $\mathcal{K}_f$ gives **ϵ-differential privacy** if for all possible values of DB and Me, and all possible outputs a,

$$\Pr[\mathcal{K}_f(\text{DB} + \text{Me}) = a] \leq \Pr[\mathcal{K}_f(\text{DB} - \text{Me}) = a] \times \exp(\epsilon)$$

Theorem: Probability of any event increases by at most $\exp(\epsilon)$.



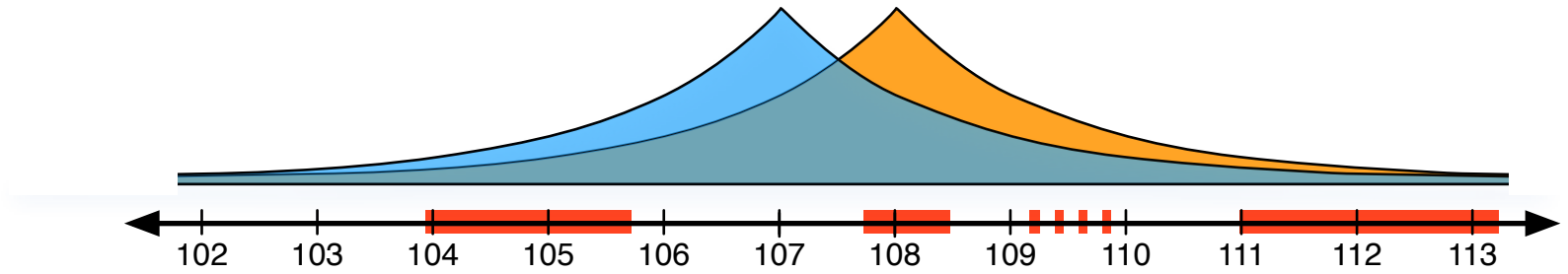
Differential Privacy

Definition

We say $\mathcal{K}_f$ gives **ϵ-differential privacy** if for all possible values of DB and Me, and all possible outputs a,

$$\Pr[\mathcal{K}_f(\text{DB} + \text{Me}) = a] \leq \Pr[\mathcal{K}_f(\text{DB} - \text{Me}) = a] \times \exp(\epsilon)$$

Theorem: Probability of any event increases by at most $\exp(\epsilon)$.



Values leading to bad event

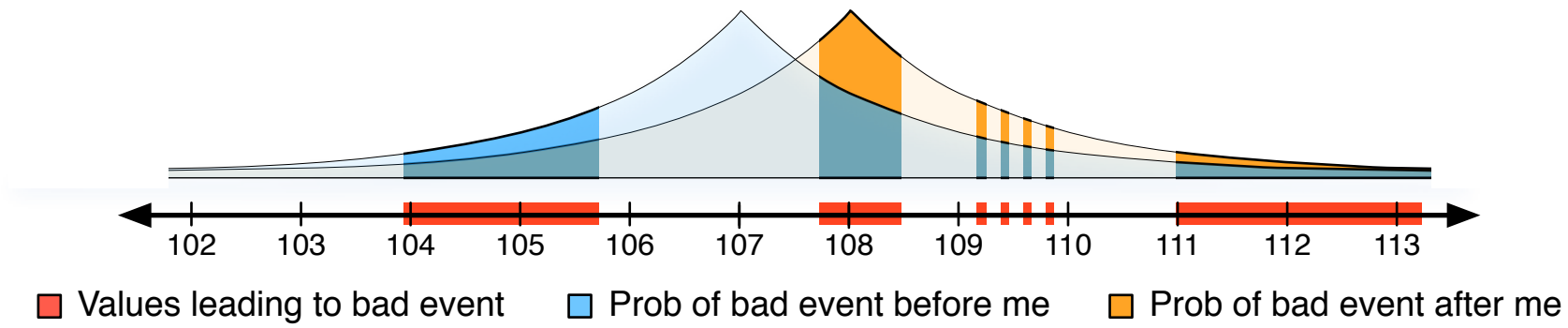
Differential Privacy

Definition

We say $\mathcal{K}_f$ gives **ϵ-differential privacy** if for all possible values of DB and Me, and all possible outputs a,

$$\Pr[\mathcal{K}_f(\text{DB} + \text{Me}) = a] \leq \Pr[\mathcal{K}_f(\text{DB} - \text{Me}) = a] \times \exp(\epsilon)$$

Theorem: Probability of any event increases by at most $\exp(\epsilon)$.



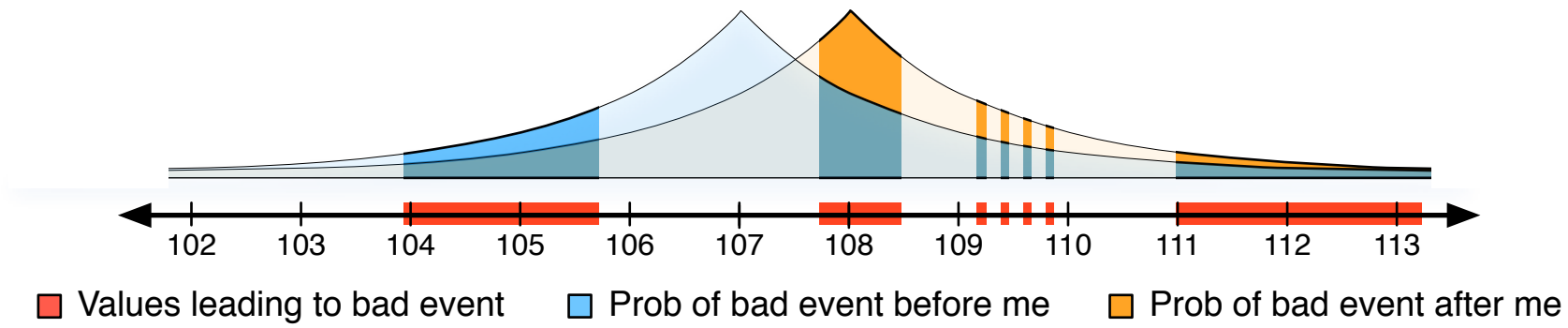
Differential Privacy

Definition

We say $\mathcal{K}_f$ gives **ϵ-differential privacy** if for all possible values of DB and Me, and all possible outputs a,

$$\Pr[\mathcal{K}_f(\text{DB} + \text{Me}) = a] \leq \Pr[\mathcal{K}_f(\text{DB} - \text{Me}) = a] \times \exp(\epsilon)$$

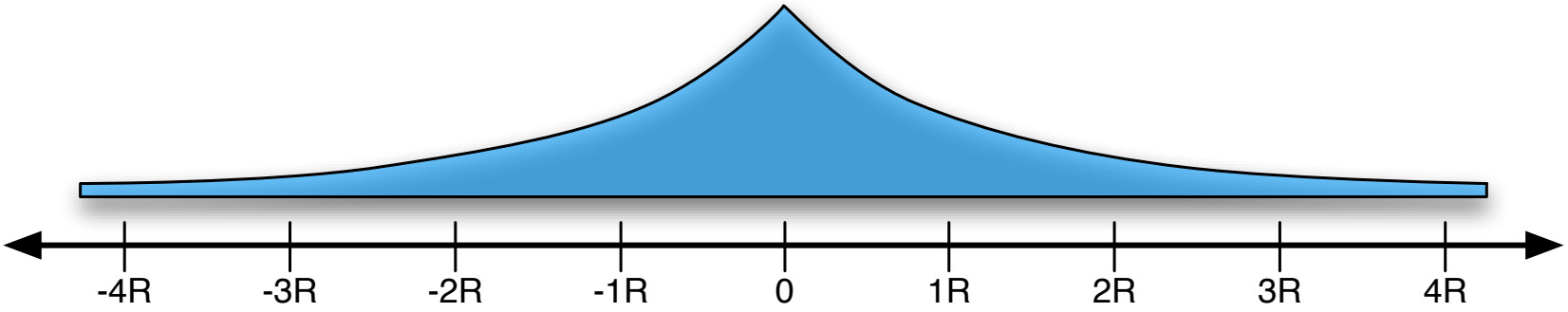
Theorem: Probability of any event increases by at most $\exp(\epsilon)$.



Important: No assumption on adversary's knowledge / power.

Exponential Noise

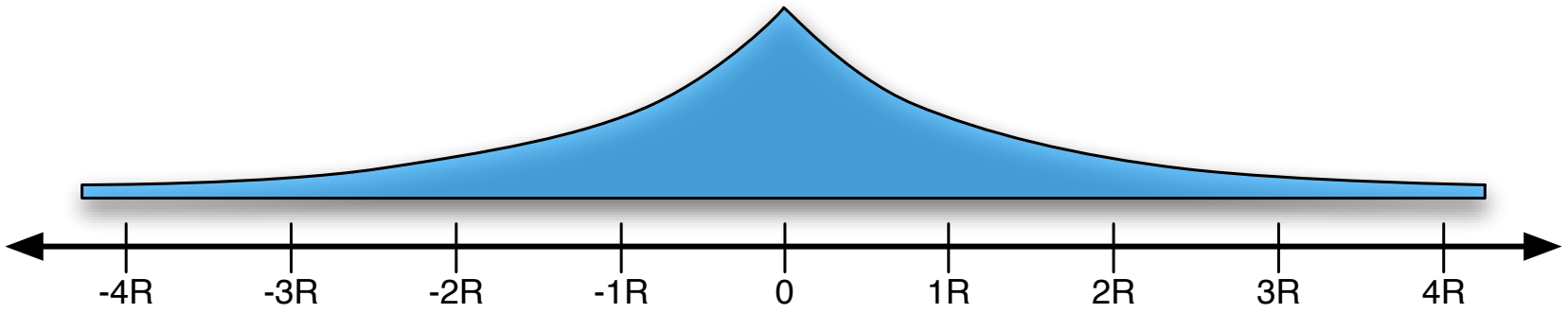
The noise distribution we use is a *scaled symmetric exponential*:



Probability of x proportional to $\exp(-|x|/R)$. Scale based on R.

Exponential Noise

The noise distribution we use is a *scaled symmetric exponential*:



Probability of x proportional to $\exp(-|x|/R)$. Scale based on R.

Definition: Let $\Delta f = \max_{\text{DB}} \max_{\text{Me}} |f(\text{DB} + \text{Me}) - f(\text{DB} - \text{Me})|$.

Theorem: For all f, $\mathcal{K}_f$ gives $(\Delta f/R)$-differential privacy.

Noise level R is determined by Δf, independent of DB, $f(DB)$.

Returning to Utility

$\mathcal{K}_f$ answers queries f with small values of Δf very accurately:

1. Counting: "How many rows have property X?"
2. Distance: "How few rows must change to give property X?"
3. Statistics: A number that a random sample estimates well.

Note: *Most* analyses are inherently robust to noise. Small Δf.

Returning to Utility

$\mathcal{K}_f$ answers queries f with small values of Δf very accurately:

1. Counting: "How many rows have property X?"
2. Distance: "How few rows must change to give property X?"
3. Statistics: A number that a random sample estimates well.

Note: *Most* analyses are inherently robust to noise. Small Δf.

$\mathcal{K}$ can also be used interactively, acting as interface to data. Programs that only interact with data through $\mathcal{K}$ are private.

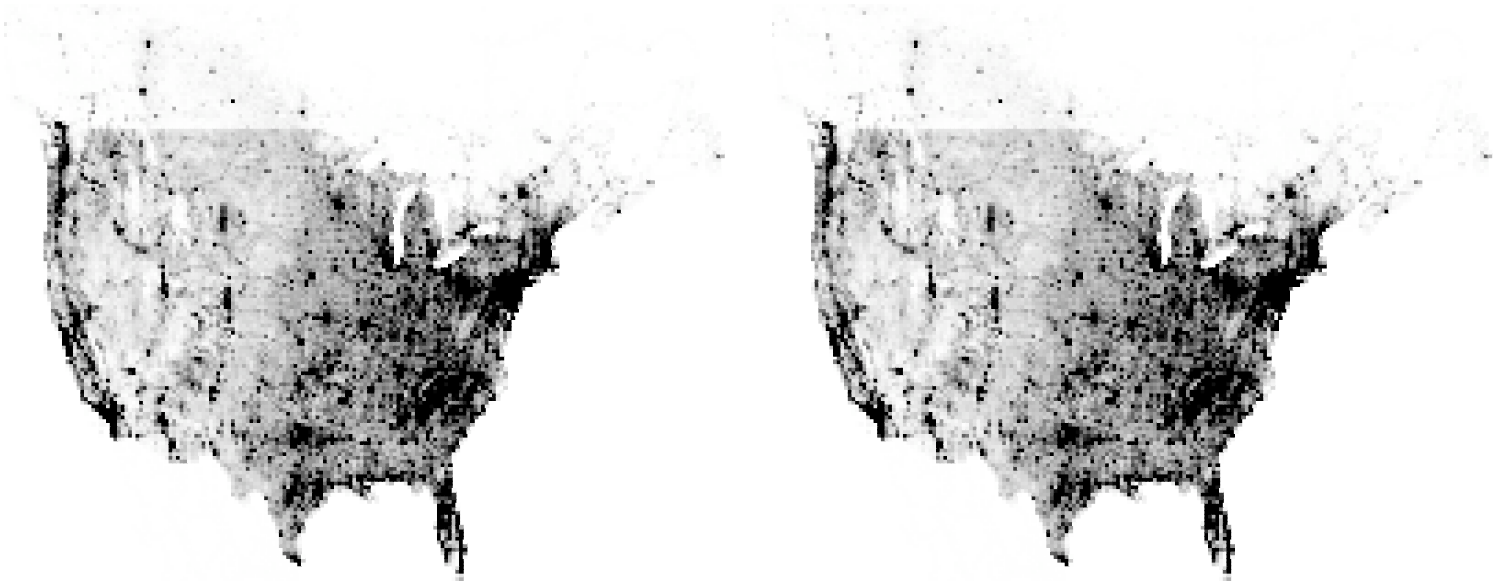
Examples: PCA, k-means, perceptron, association rules, ...

Challenging and fun part is re-framing the algorithms to use $\mathcal{K}$. Queries have cost! Every query can degrade privacy by up to ϵ.

Example: Traffic Histogram

Database of traffic intersections. Each row is a (x, y) pair.
Histogram counts intersections in each of $64,909$ grid cells.

Counting performed using $\mathcal{K}$, with 1.000-differential privacy.



Maximum counting error: 13. Average counting error: 1.02.

Example: Traffic Histogram

Database of traffic intersections. Each row is a (x, y) pair.
Histogram counts intersections in each of $64,909$ grid cells.

Counting performed using $\mathcal{K}$, with 1.000-differential privacy.

Maximum counting error: 13. Average counting error: 1.02.

Example: Traffic Histogram

Database of traffic intersections. Each row is a (x, y) pair. Histogram counts intersections in each of $64,909$ grid cells.

Counting performed using $\mathcal{K}$, with 0.100-differential privacy.

Maximum counting error: 109. Average counting error: 9.12.

Example: Traffic Histogram

Database of traffic intersections. Each row is a (x, y) pair.
Histogram counts intersections in each of $64,909$ grid cells.

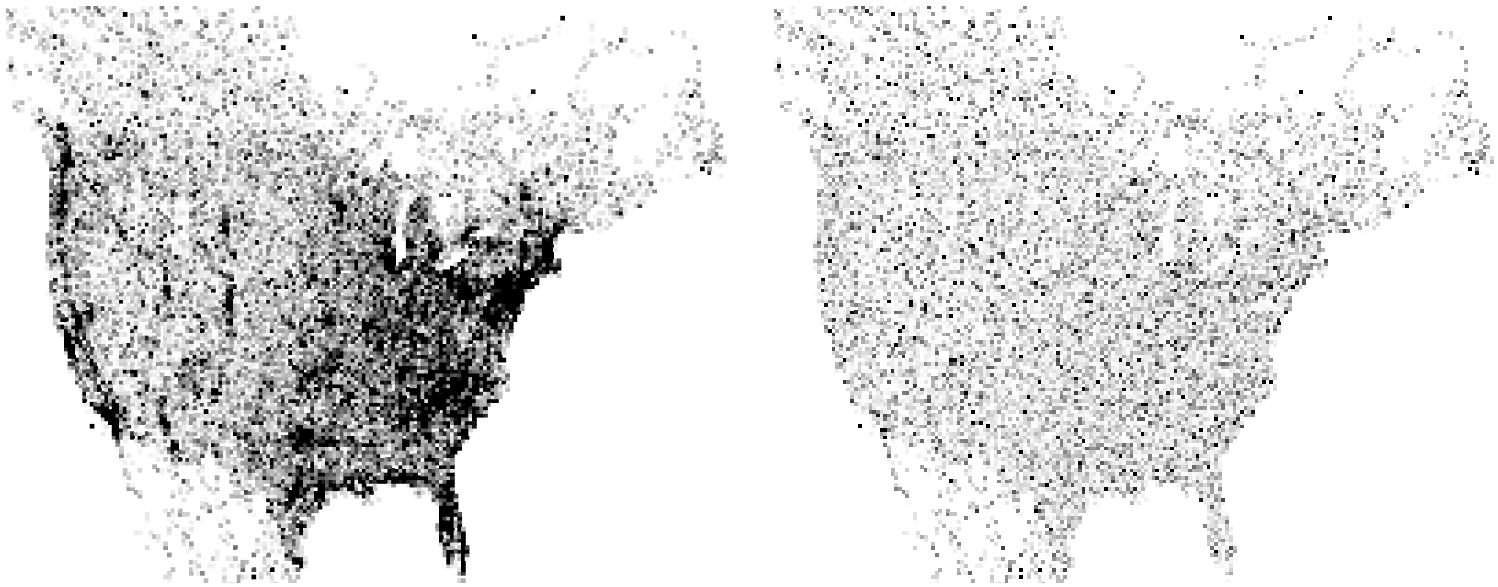
Counting performed using $\mathcal{K}$, with 0.010-differential privacy.

Maximum counting error: 1041. Average counting error: 98.56.

Example: Traffic Histogram

Database of traffic intersections. Each row is a (x, y) pair.
Histogram counts intersections in each of $64,909$ grid cells.

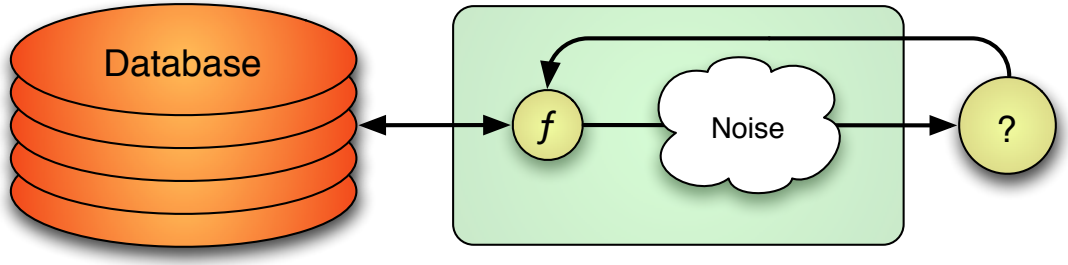
Counting performed using $\mathcal{K}$, with 0.001-differential privacy.



Maximum counting error: 9663. Average counting error: 1003.23.

Wrapping Up

Interactive output perturbation based sanitization mechanism: $\mathcal{K}$



Using appropriately scaled exponential noise gives:

1. Provable privacy guarantees about participation in DB.
2. Very accurate answers to queries with small Δf.

Protects individual info and releases aggregate info at same time.

Configurable: Boundary between individual/aggregate set by R.

Other Work in MSR

Web Page URL:

`http://research.microsoft.com/research/sv/DatabasePrivacy/`

Other work:

- Impossibility results: What can and can not be done.
- Weaker positive results in the non-interactive setting.
- Connections to: Game theory, Online learning, etc...
- Enforcing privacy using cryptography.