Privacy Preserving Data Analysis

Frank McSherry

What is "Privacy"?

Lots of useful data out there, containing valuable information.

Substantial, and reasonable, concern about sensitive data.

Access control alone isn't an answer; we want to understand sensitive parts of a dataset and publish our conclusions.

In this talk "privacy" will be about releasing restricted but useful information about sensitive data.

1. Early privacy definitions: k-anonymity, l-diversity, m-invariance, t-
2. A more recent definition: Differential Privacy.
3. Some applications thereof.

Syntactic Privacy Definitions

Prevailing wisdom has been that privacy relates to databases. Record attributes are either "public", or "sensitive".

Age	Gender	Diagnosis
37	Male	Flu
35	Male	Flu
32	Female	Flu
23	Male	STD
37	Male	HIV
63	Male	Dead

Releasing this information reveals information about patients.

k-Anonymity

The first approach, **k-Anonymity**, requires suppressing public attributes so that each equivalence class has size at least k:

Age	Gender	Diagnosis
37	Male	Flu
35	Male	Flu
32	Female	Flu
23	Male	STD
37	Male	HIV
63	Male	Dead

→

Age	Gender	Diagnosis
30s	*	Flu
30s	*	Flu
30s	*	Flu
*	Male	STD
*	Male	HIV
*	Male	Dead

Does the table on the right actually protect everyone's privacy?

l-Diversity

The next approach, **l-Diversity** requires equivalence classes to have sufficiently high entropy secret attributes:

Age	Gender	Diagnosis
30s	*	Flu
30s	*	Flu
30s	*	Flu
*	Male	STD
*	Male	HIV
*	Male	Dead

→

Age	Gender	Diagnosis
30s	*	*
30s	*	*
30s	*	*
*	Male	STD
*	Male	HIV
*	Male	Dead

Does the table on the right actually protect everyone's privacy?

Syntactic Definitions

There are fundamental problems with these sorts of approaches:

Age	Gender	Diagnosis
30s	*	Flu
30s	*	Flu
30s	*	Flu
*	Male	STD
*	Male	HIV
*	Male	Dead

and

Age	Gender	Diagnosis
20s	Male	Flu
20s	Male	Flu
20s	Male	HIV
*	*	Flu
*	*	Lupus
*	*	Gout

Imagine you are the 20 year-old male who went to both hospitals.

Problem: These guarantees are syntactic, rather than semantic. No bounds on how much I can learn about the actual input data.

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.

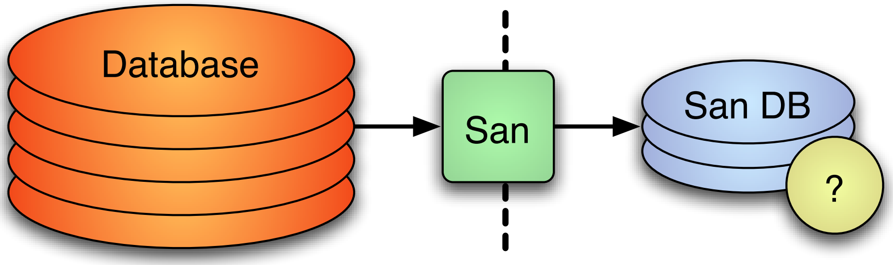
Problem: Exact aggregate info may leak individual info. Eg:

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

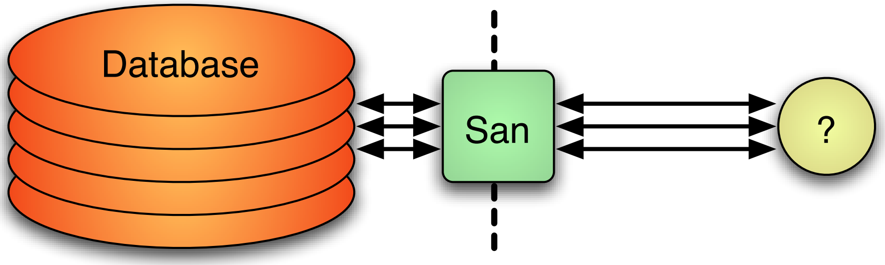
Goal: Method to protect individual info, release aggregate info.

Two Privacy Models

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



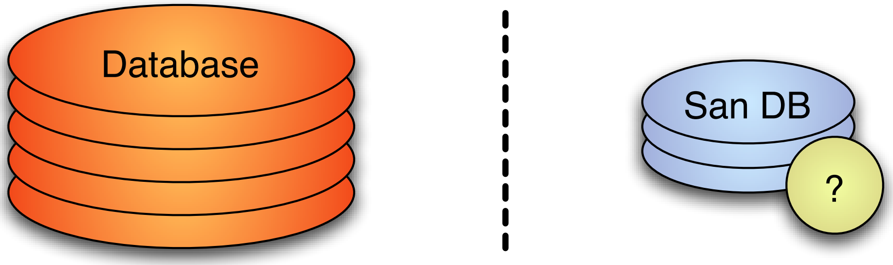
2. **Interactive**: Multiple questions asked / answered adaptively.



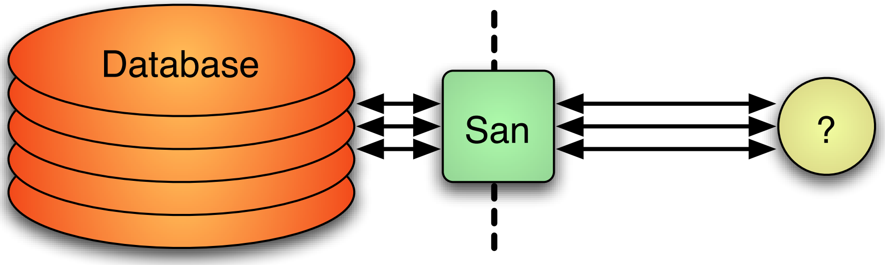
We will focus on the interactive model in this talk.

Two Privacy Models

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



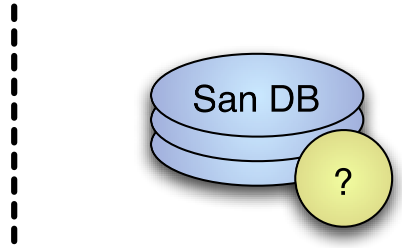
2. **Interactive**: Multiple questions asked / answered adaptively.



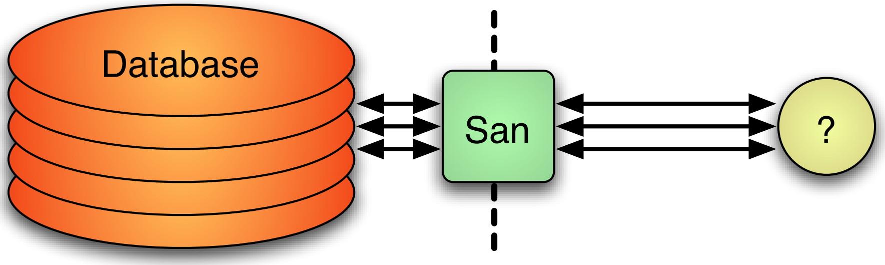
We will focus on the interactive model in this talk.

Two Privacy Models

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



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

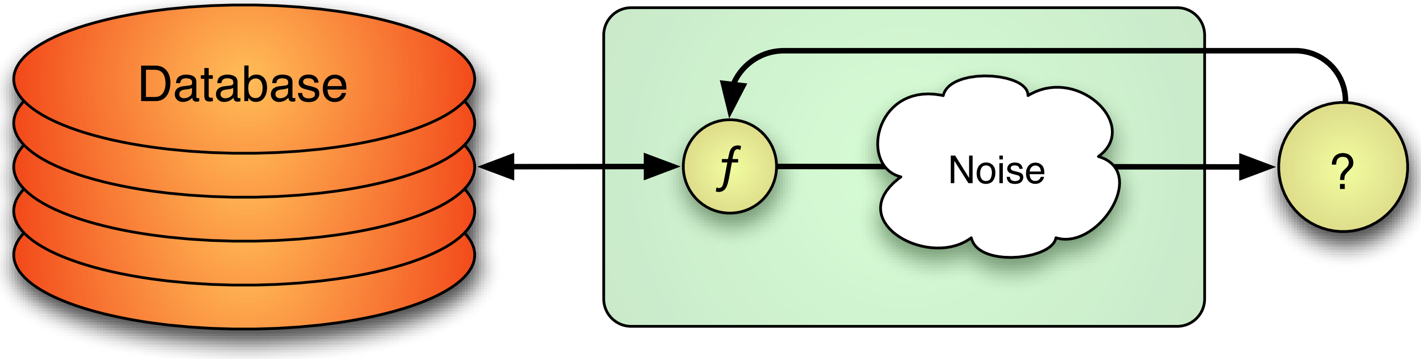


We will focus on the interactive model in this talk.

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

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

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



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

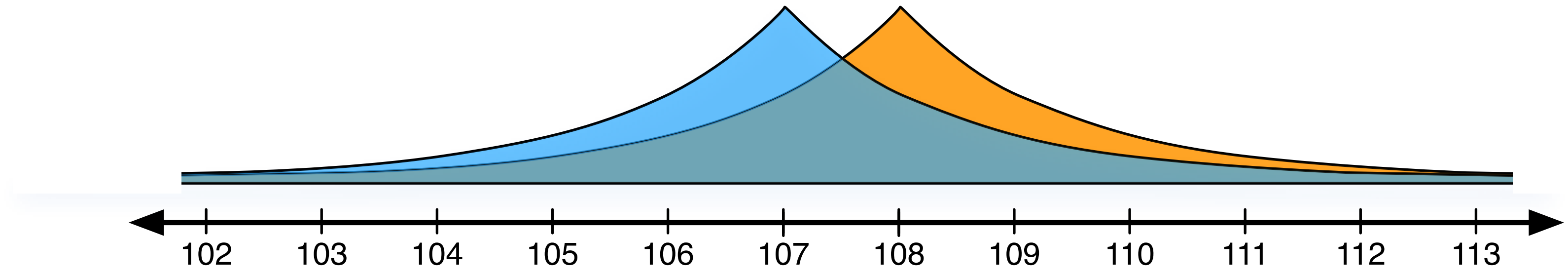
Important: The amount of noise, and privacy, is configurable. Determined by a privacy parameter ϵ and the query function f.

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 $\mathcal{K}_f(\mathrm{DB} - \mathrm{Me})$ and $\mathcal{K}_f(\mathrm{DB} + \mathrm{Me})$:



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

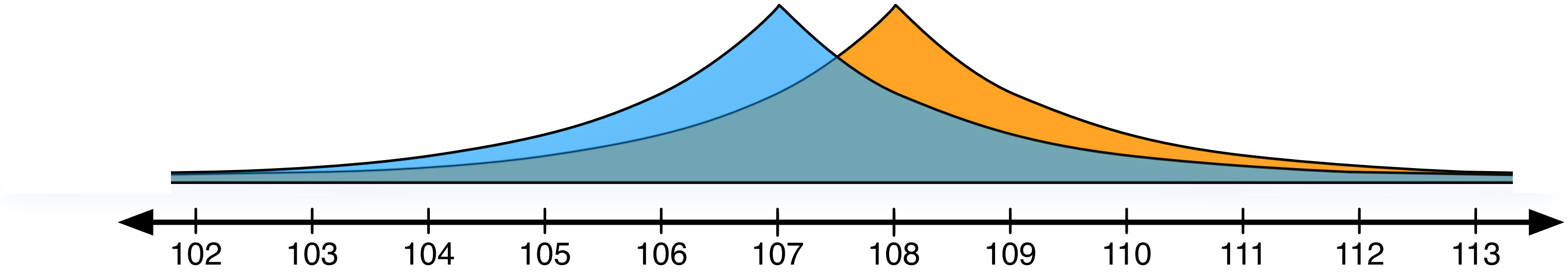
If not, then all events are just as likely now as they were before. Any behavior based on the output is just as likely now as before.

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,

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

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

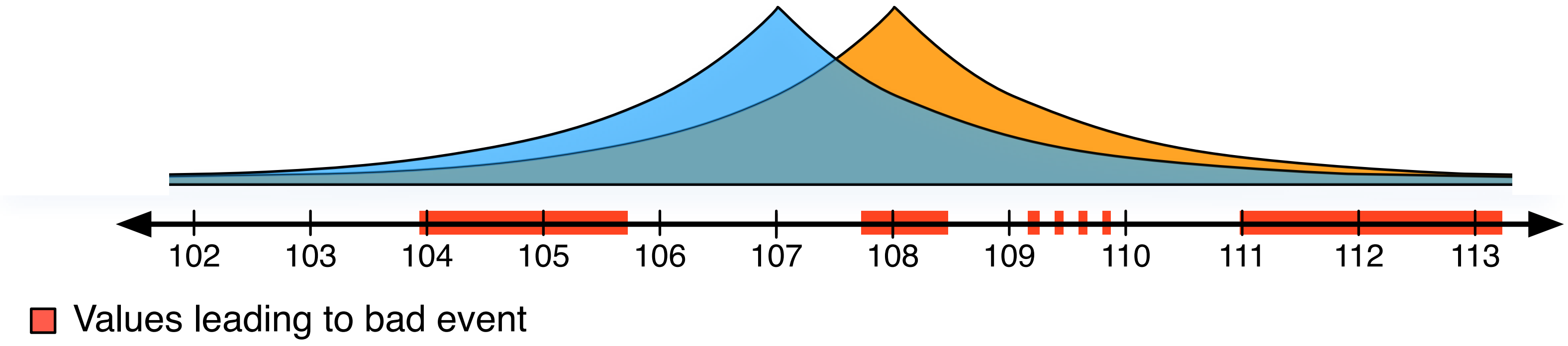


Differential Privacy

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

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

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

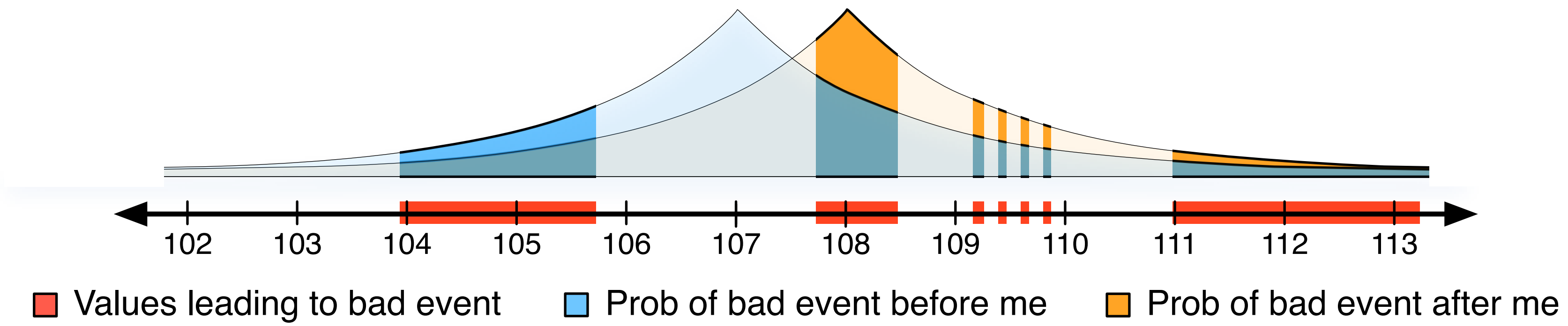


Differential Privacy

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 $\exp(\epsilon)$.



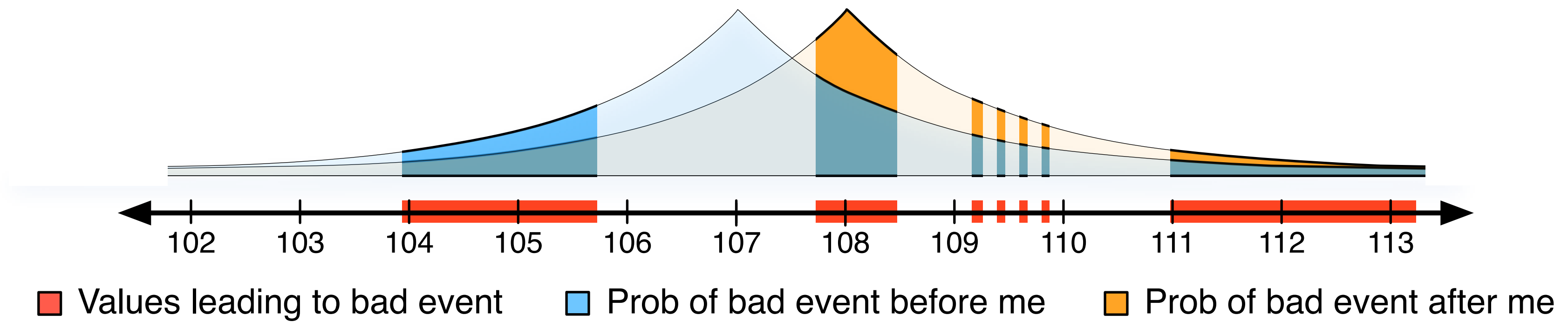
Differential Privacy

Definition

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

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

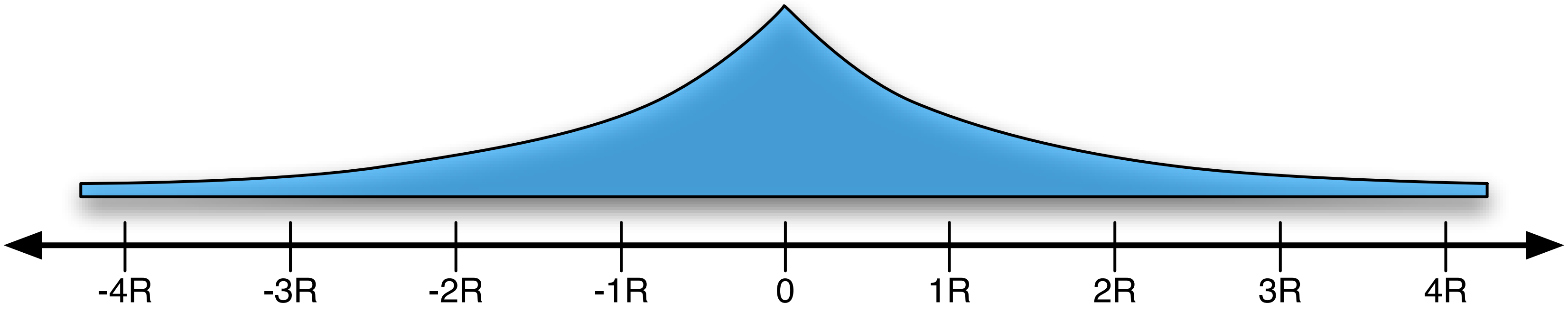
Theorem: Probability of any event increases by at most $\exp(\epsilon)$.



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

Exponential Noise

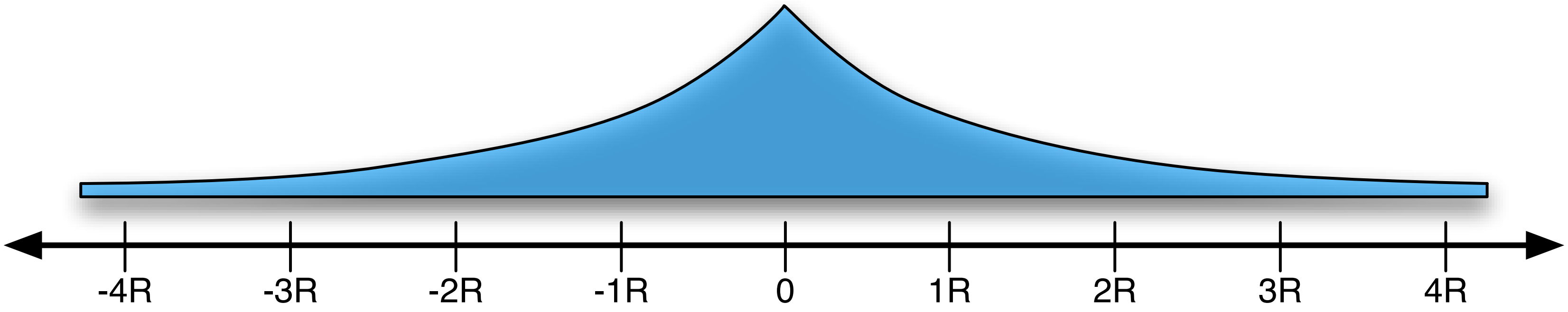
The noise distribution we use is a *scaled symmetric exponential*:



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.

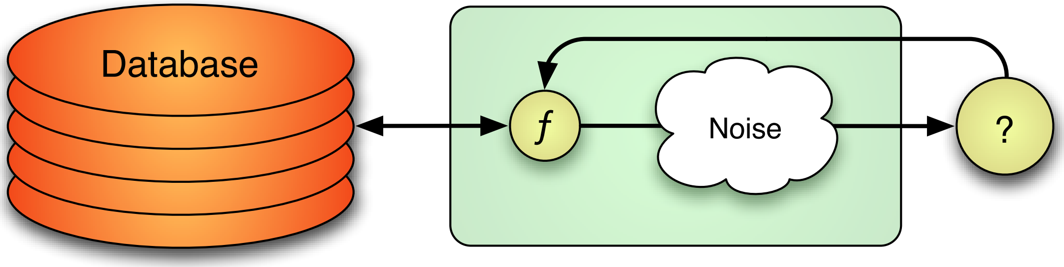
Definition: Let $\Delta f \;=\; \max_{\mathrm{DB}} \max_{\mathrm{Me}} \left| f(\mathrm{DB} + \mathrm{Me}) - f(\mathrm{DB} - \mathrm{Me}) \right| .$

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

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

Summing 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.

Doing things with DP

Let's start to look at more some interesting things we can do.

First, we'll generalize our noise mechanism to many dimensions.

1. k-Means clustering algorithms.
2. Histograms and visualization.

Second, we'll look at a mechanism which bypasses additive noise.

1. Motivated by problems and applications to game theory.
2. Results in a fully general mechanism for ϵ-DP.

Data Mining (k-Means)

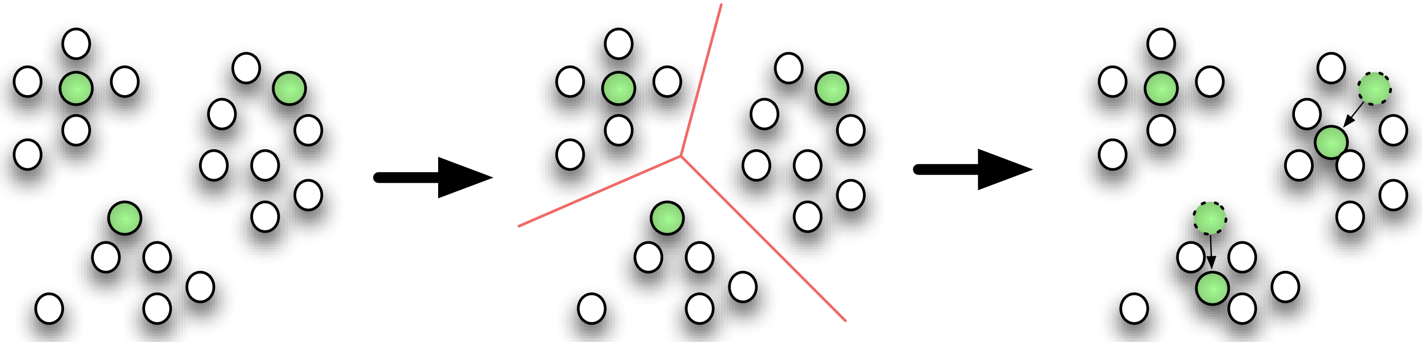
Data: points $x_1, \ldots, x_n$ in R^d. **Input**: candidate means $\mu_1, \ldots, \mu_k$.

K-Means($\mu_1, \ldots, \mu_k$)

1. For each $1 \leq i \leq k$, compute

$$S_i \quad = \quad \{x : i = \arg\min_j \|x - \mu_j\|\}$$

2. For each $1 \leq i \leq k$, return $\mu'_i = \mathrm{avg}_{S_i} x$ as the new mean.



Data mining with $\mathcal{K}$

By using sneaky functions f, we can emulate k-means:

```
$\mathcal{K}$-Means($\mu_1, \ldots, \mu_k$)
  1. For each $1 \leq i \leq k$, compute both
         $s_i = \mathcal{K}(f(x) := 1 \text{ iff } i = \arg\min_j \|x - \mu_j\|)$
         $m_i = \mathcal{K}(f(x) := x \text{ iff } i = \arg\min_j \|x - \mu_j\|)$
  2. For each $1 \leq i \leq k$, return $m_i/s_i$ as the new mean.
```

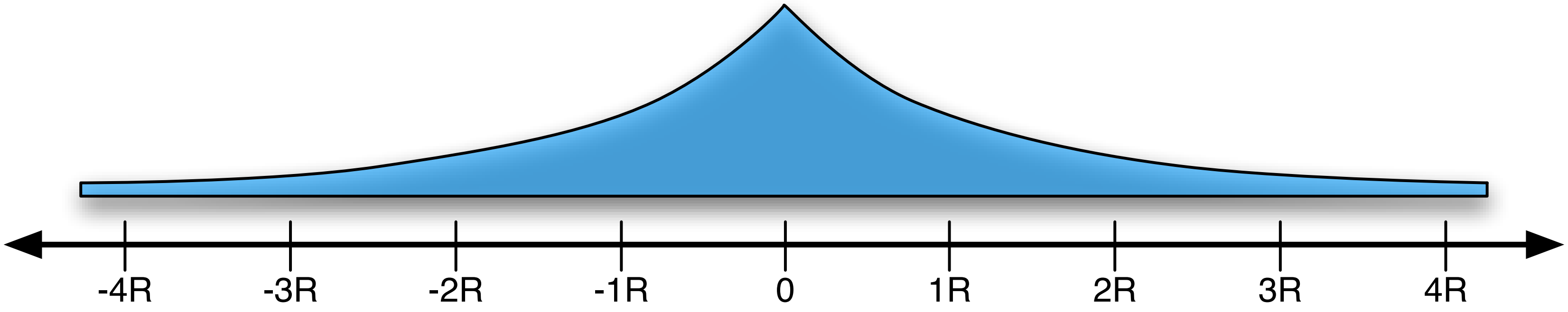
Obs 1: If $|S_i|$ is sufficiently large, then $m_i/s_i \approx \mu'_i$.

Obs 2: In t iterations, $\mathcal{K}$-means poses $(d+1)kt$ questions.

Obs 3: Only access to data is through $\mathcal{K}$. Privacy automatic.

Tightening Privacy Guarantees

The standard composition rules for DP have us add all ϵ.
This can be pessimistic for some analyses, like k-means.



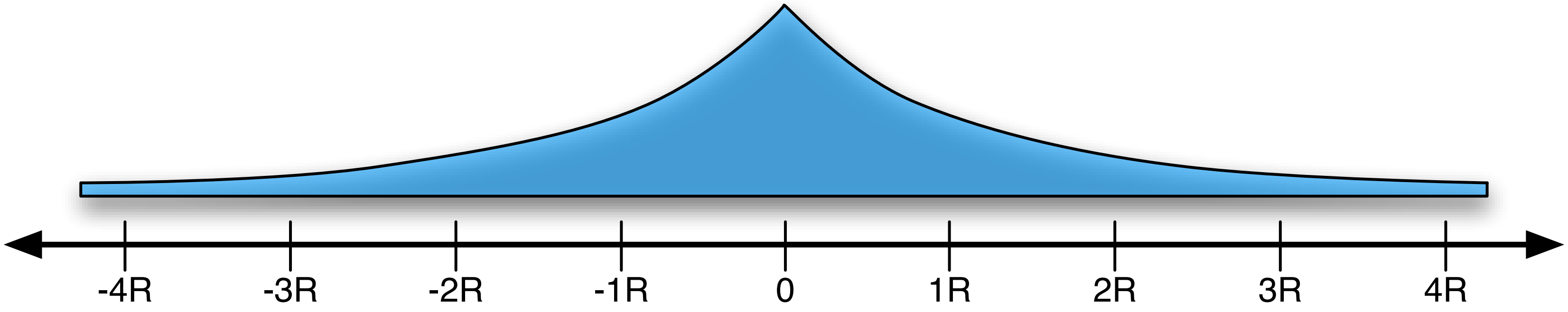
Probability of x proportional to $\exp(-|x|/R)$. Scale based on R.

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

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

Tightening Privacy Guarantees

The standard composition rules for DP have us add all ϵ.
This can be pessimistic for some analyses, like k-means.



Probability of x proportional to exp$(-\|x\|_1/R)$. Scale based on R.

Definition: Let $\Delta f \;=\; \max_{\text{DB}} \max_{\text{Me}} \|f(\text{DB} + \text{Me}) - f(\text{DB} - \text{Me})\|_1$.

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

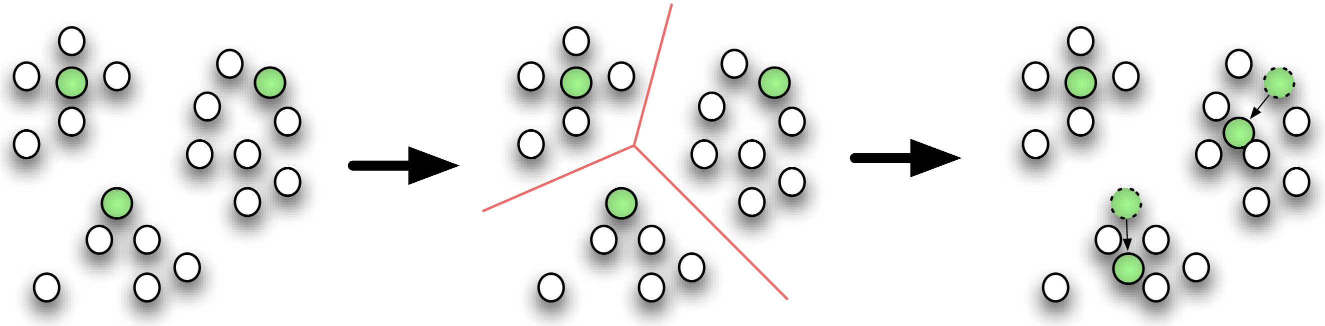
Data Mining (k-Means)

$\mathcal{K}$-**Means**$(\mu_1, \ldots, \mu_k)$

1. For each $1 \leq i \leq k$, compute both

$$\begin{aligned} s_i &= \mathcal{K}(f(x) := 1 \text{ iff } i = \arg\min_j \|x - \mu_j\|) \\ m_i &= \mathcal{K}(f(x) := x \text{ iff } i = \arg\min_j \|x - \mu_j\|) \end{aligned}$$

2. For each $1 \leq i \leq k$, return m_i / s_i as the new mean.

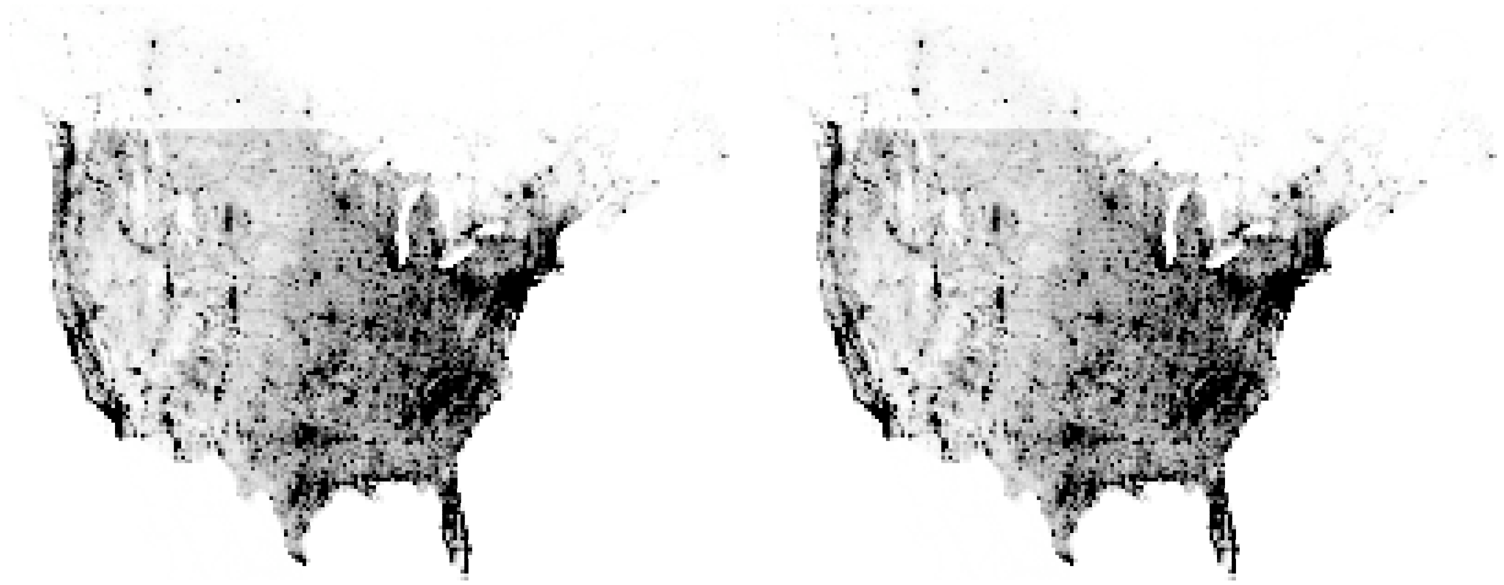


$$\Delta f = \max_{\mathrm{DB}} \max_{\mathrm{Me}} \|f(\mathrm{DB} + \mathrm{Me}) - f(\mathrm{DB} - \mathrm{Me})\|_1 \leq d + 1$$

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.

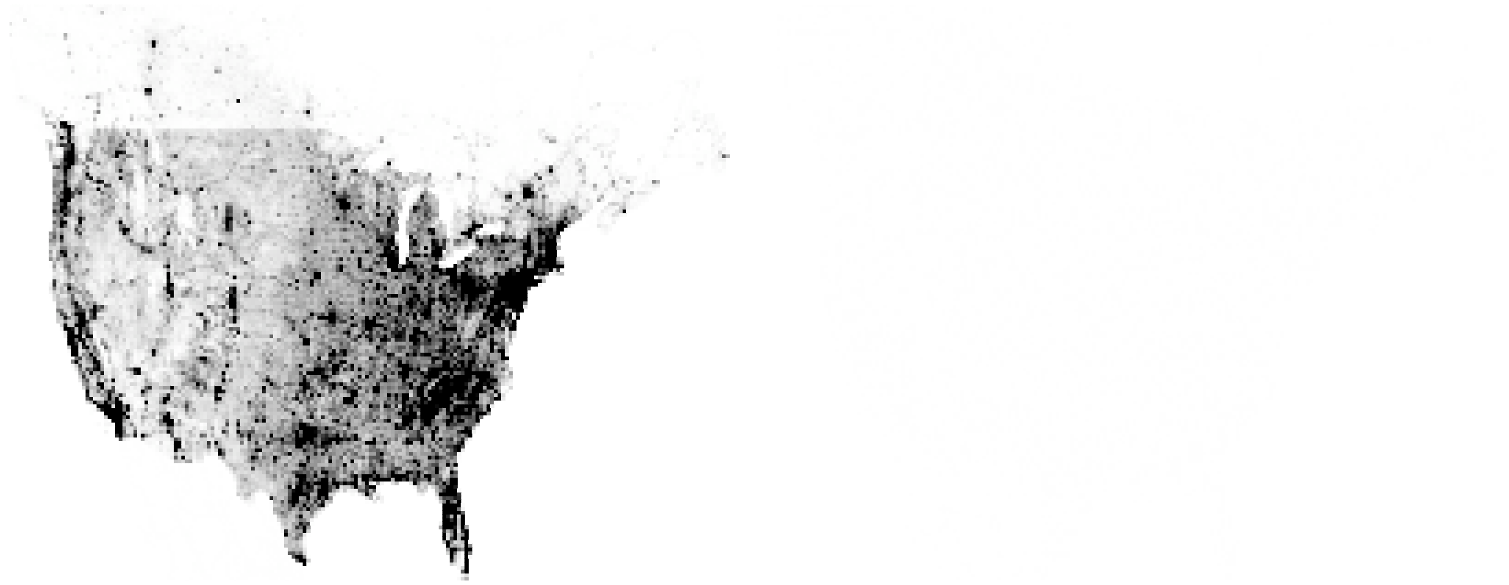


Maximum counting error: 13. Average counting error: 1.02.

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.

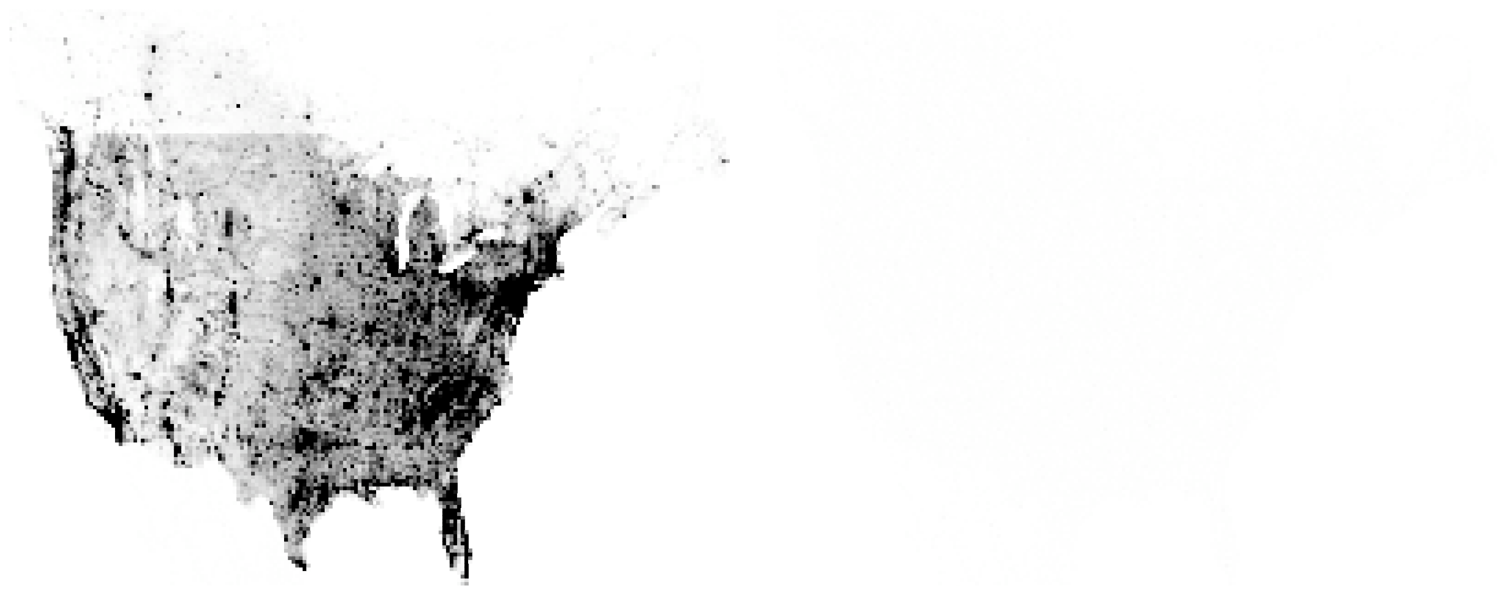


Maximum counting error: 13. Average counting error: 1.02.

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.

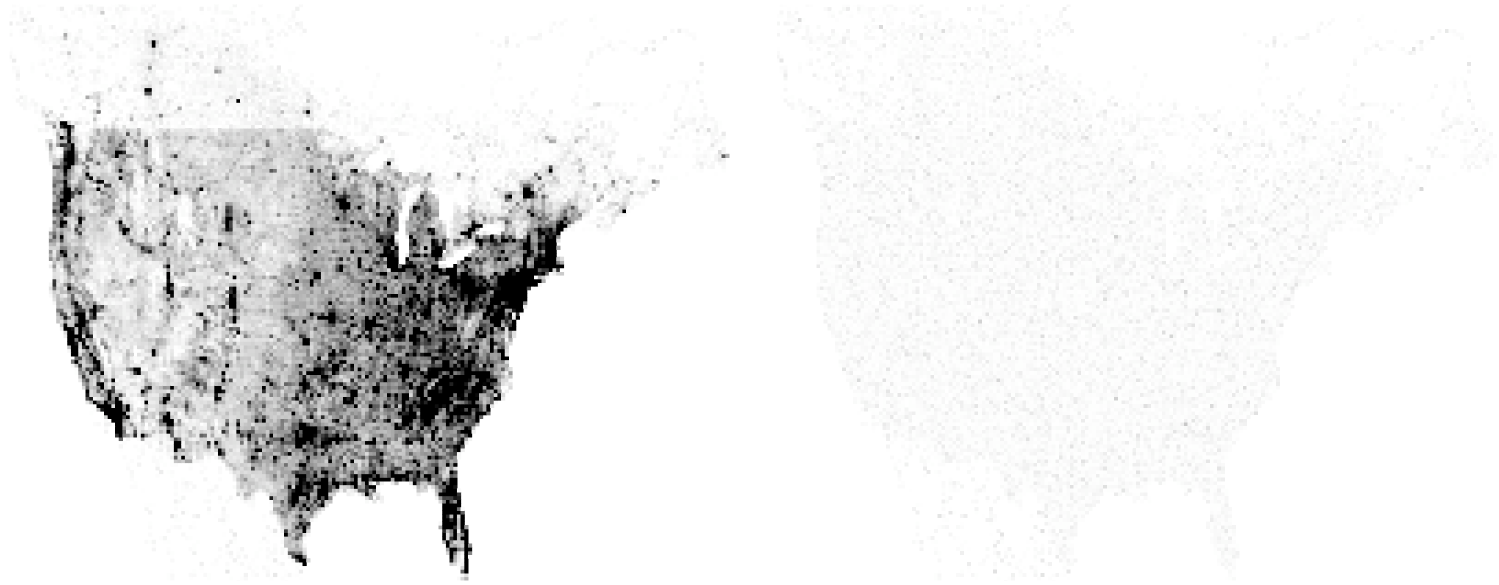


Maximum counting error: 109. Average counting error: 9.12.

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.

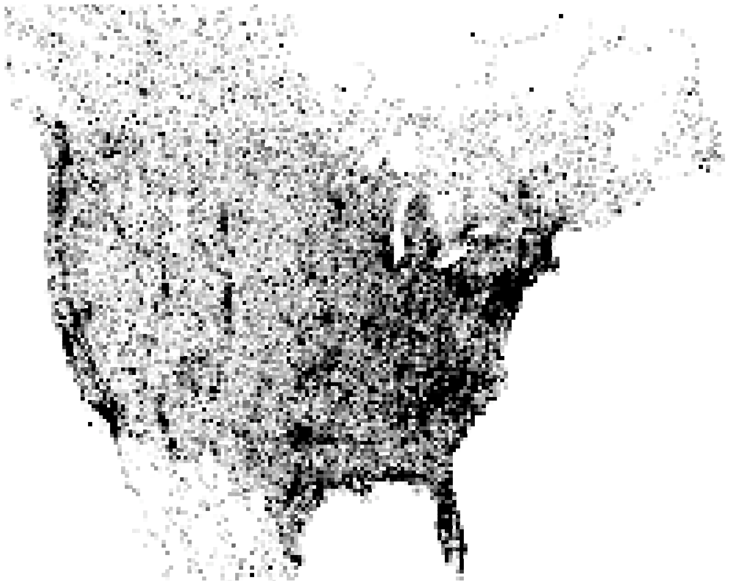


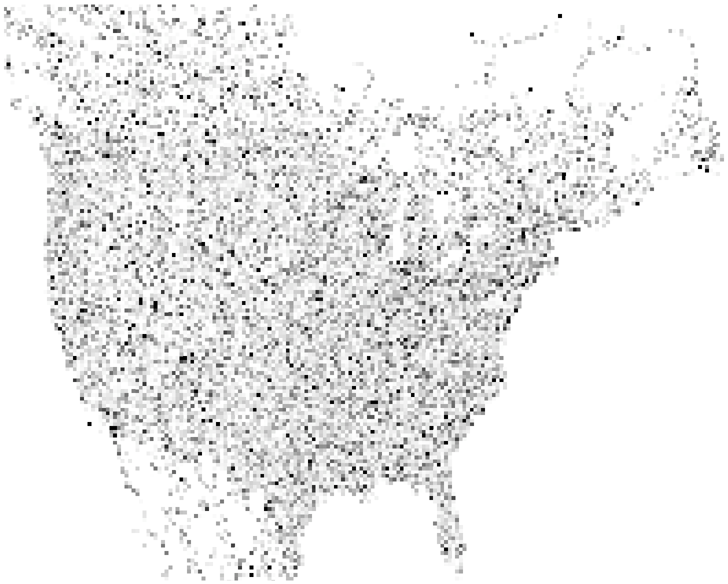
Maximum counting error: 1041. Average counting error: 98.56.

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.

Problems with Perturbation

Consider trying to price some good with a fixed production cost. Picking the price high or low influences your revenue.

Pricing: Inputs are n bids in $[0, 1]$. Output is a price $p \in [0, 1]$.

Problem: Perturbing the true answer by some noise may fail.

1. The function may have high sensitivity. (eg: Pricing)
2. Perturbations may not actually be useful. (eg: Pricing)

Moreover: Additive perturbations also fail when

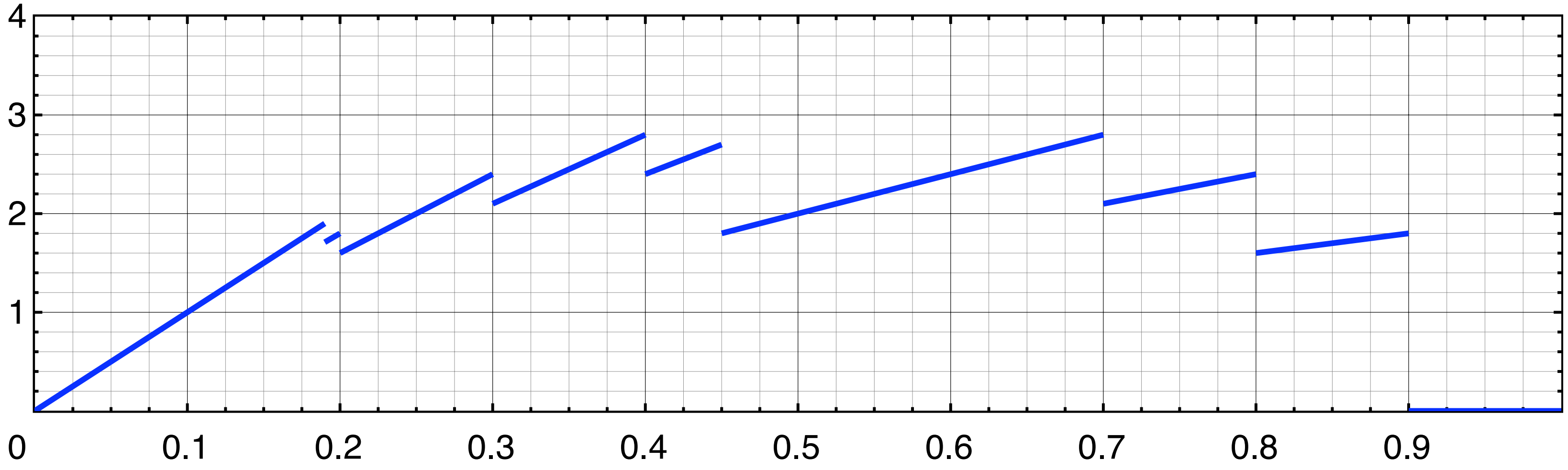
3. Outputs are not numbers. (eg: strings, trees, etc...)

A General Mechanism

Previously a "query" was $f : \mathcal{D}^N \rightarrow \mathbb{R}^d$, mapping data to result. Implicit assumption that results r near $f(d)$ are nearly as good.

Now, a query is $q : (\mathcal{D}^N \times \mathcal{R}) \rightarrow \mathbb{R}$. Score of result r for data d.

Eg: Given bids and a price, revenue is $q(d, r) = r \times \#(i : d_i > r)$.

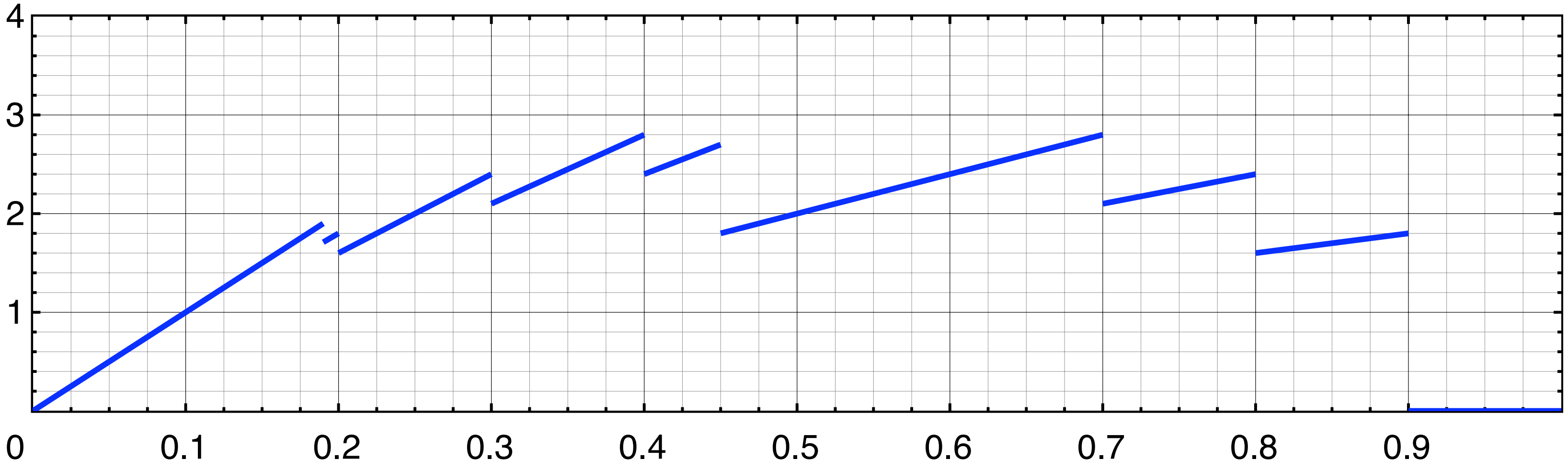


A General Mechanism

Previously a "query" was $f : \mathcal{D}^N \to \mathbb{R}^d$, mapping data to result. Implicit assumption that results r near $f(d)$ are nearly as good.

Now, a query is $q : (\mathcal{D}^N \times \mathcal{R}) \to \mathbb{R}$. Score of result r for data d.

Eg: Given bids and a price, revenue is $q(d, r) = r \times \#(i : d_i > r)$.



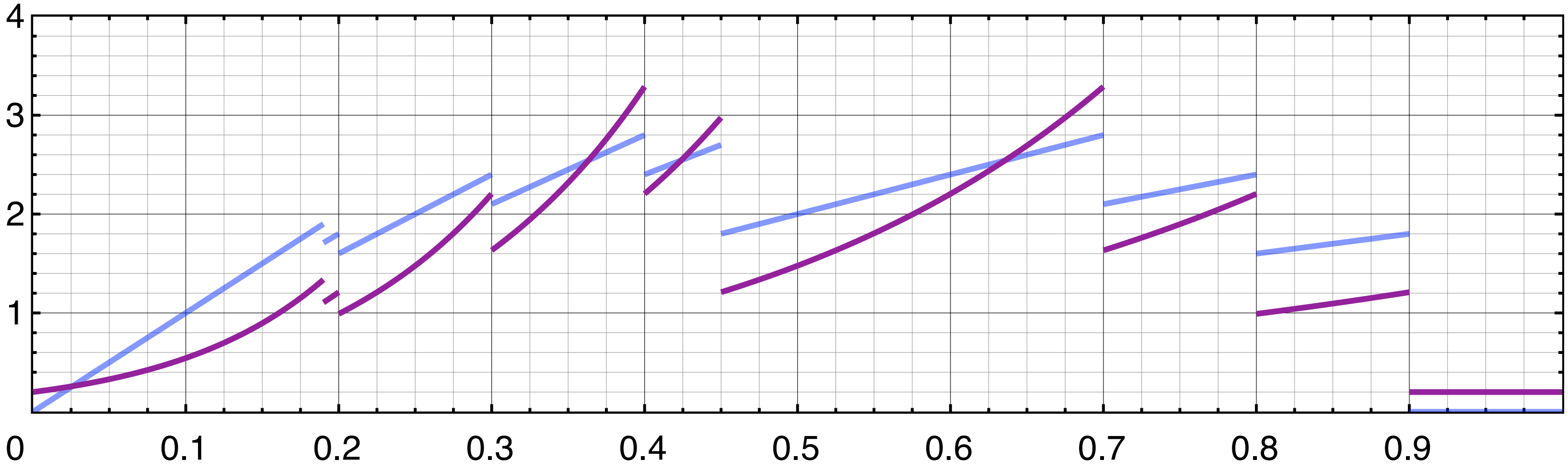
Definition: Let $\mathcal{E}_q^\epsilon(d)$ output r with probability $\propto$ $\exp(\epsilon q(d, r))$.

A General Mechanism

Previously a "query" was $f : \mathcal{D}^N \to \mathbb{R}^d$, mapping data to result. Implicit assumption that results r near $f(d)$ are nearly as good.

Now, a query is $q : (\mathcal{D}^N \times \mathcal{R}) \to \mathbb{R}$. Score of result r for data d.

Eg: Given bids and a price, revenue is $q(d, r) = r \times \#(i : d_i > r)$.



Definition: Let $\mathcal{E}_q^\epsilon(d)$ output r with probability $\propto$ exp$(\epsilon q(d, r))$.

The Exponential Mechanism

Definition: Let $\mathcal{E}_q^\epsilon(d)$ output r with probability $\propto$ exp($\epsilon q(d, r)$).

This mechanism is referred to as "the Exponential Mechanism". It magically evaluates and pulls results from an arbitrary set $\mathcal{R}$, without incurring privacy cost proportional to $|\mathcal{R}|$.

Privacy: $\mathcal{E}_q^\epsilon$ gives ($2\epsilon\Delta q$)-differential privacy, where we define

$$\Delta q \quad = \quad \max_{r \in \mathcal{R}} \max_{d \approx d'} |q(d, r) - q(d', r)| \ .$$

At the same time, it selects great results from discrete sets.

Utility: $Pr[q(d, \mathcal{E}_q^\epsilon(d)) < OPT - t/\epsilon]$ is at most $|R|$ exp($-t$).

Also "complete" for DP. Any DP computation has a q function.

Applications to Pricing

Every bidder gives a demand curve: $d_i : [0,1] \to \mathbb{R}^+$. $(rd_i(r) \leq 1)$

Theorem: Taking $q(d,r) = r\sum_i d_i(r)$, then the mechanism $\mathcal{E}_q^\epsilon$ gives (2ϵ)-differential privacy, and has expected revenue at least

$$OPT - 3\ln(e + \epsilon^2 OPTm)/\epsilon\ ,$$

where m is the number of items sold at the optimal price.

Proof: Grind $t = \ln(e + \epsilon^2 OPTm)$ through continuous theorem. Argue that $\mu(S_t)$ is not small. (near-opt r gives near-opt $q(d,r)$).