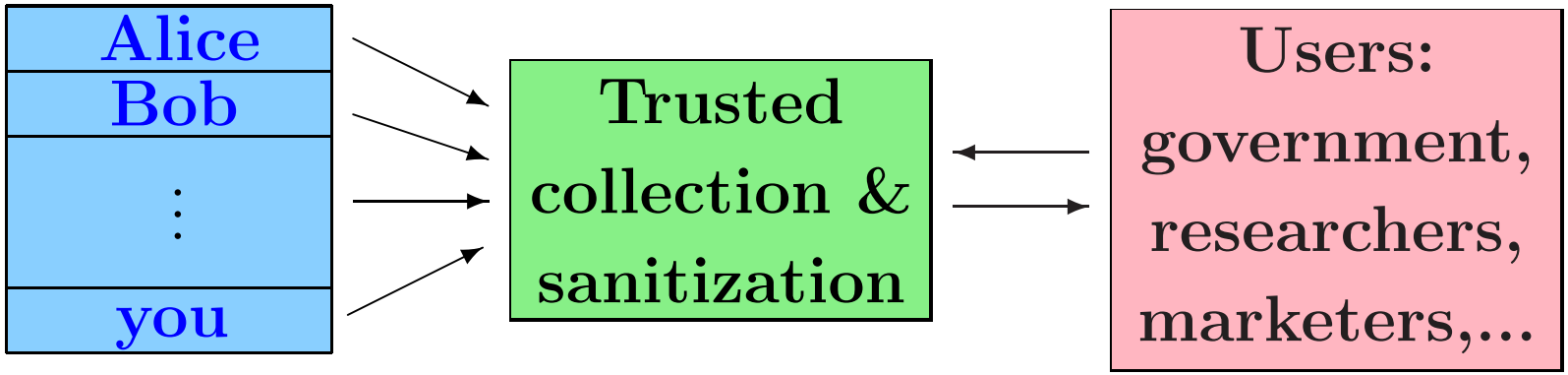
# *Smooth Sensitivity and Sampling in Private Data Analysis*

**Sofya Raskhodnikova**
Penn State University

Joint work with **Kobbi Nissim** (Ben Gurion University)
and **Adam Smith** (Penn State University)

# *Private data analysis*




Collections of personal and sensitive data

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

# *Meta Question*

## What information can be released?

- Two conflicting goals
  - utility: users can extract "global" statistics
  - privacy: individual information stays hidden
- **Not** the usual secure computation problem:
  not "how"...
  - We trust census bureau (in this work): no need for SFE

  ... but "what"
  - What functions can we compute without breaching privacy?

## *Related work*

Other fields: huge amount of work

- in statistics (statistical disclosure limitation)
- in data mining (privacy-preserving data mining)
- largely: no precise privacy definition
  (only security against specific attacks)

In cryptography (private data analysis)

- [Dinur Nissim 03, Dwork Nissim 04,
  Chawla Dwork McSherry Smith Wee 05,
  Blum Dwork McSherry Nissim 05, Chawla Dwork McSherry Talwar 05,
  Dwork McSherry Nissim Smith 06, Dwork Naor 06,
  Dwork McSherry Talwar 07, ...]
- rigorous privacy guarantees

## *Our main contributions*

- New tools for private data analysis
- Greatly expand the types of information that can be released

## *Road map*

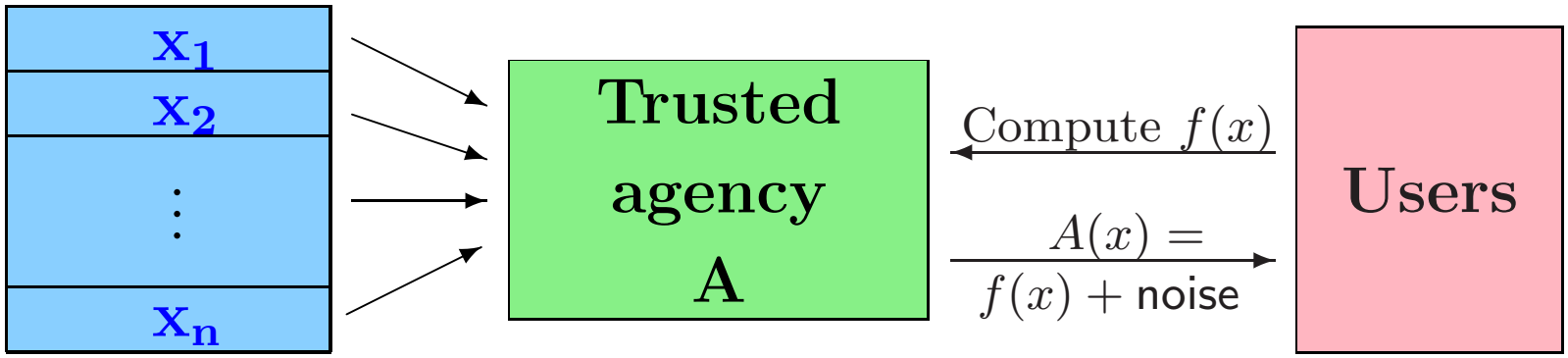
I. Introduction

- Definitions
- Global sensitivity framework [DMNS06]

II. Smooth sensitivity framework

III. Sample-and-Aggregate framework

# *Output perturbation*




Each row is arbitrarily complex data supplied by 1 person.

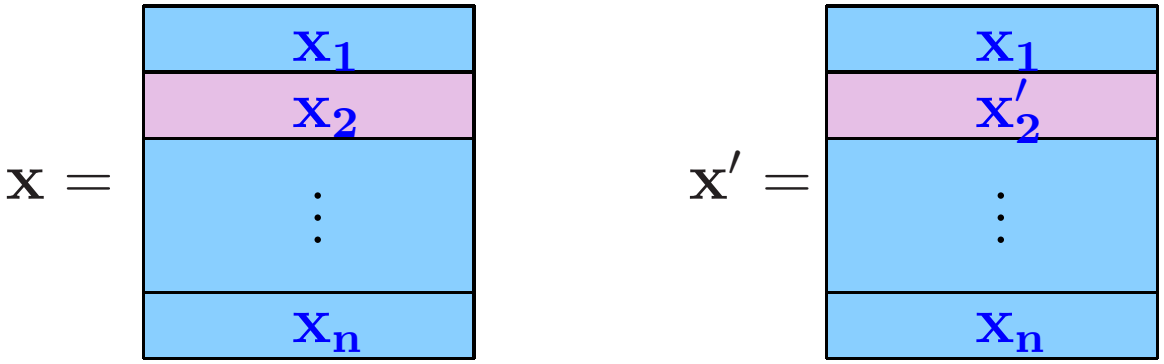
For which functions $f$ can we have:

- utility: little noise
- privacy: differential privacy [DMNS06, D06]

# *Differential privacy* [DMNS06]

*Intuition:* Users learn the same thing about me
whether or not I participate in the census

Two databases are *neighbors* if they differ in one row.




**Privacy definition**

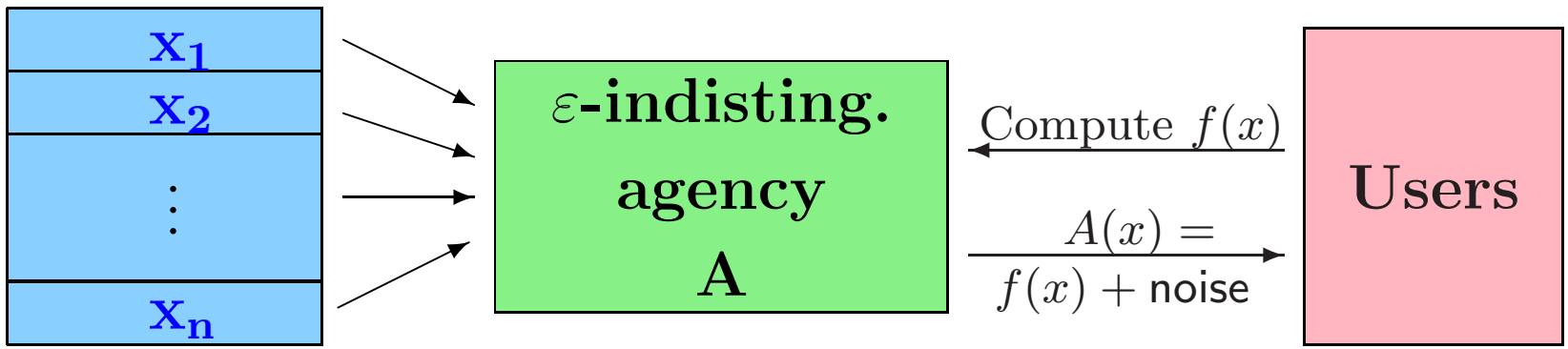
*Algorithm $A$ is $\varepsilon$-differentially private if*

- *for all neighbor databases $x, x'$*
- *for all sets of answers $S$*

$$\Pr[A(x) \in S] \leq (1+\varepsilon) \cdot \Pr[A(x') \in S]$$

# *Properties of differential privacy*

- robust in the presence of arbitrary side information
- $\varepsilon$ is non-negligible (at least $\frac{1}{n}$).
- *Composition:* If $A$ is $\varepsilon$-differentially private on each query, it is $\varepsilon q$-differentially private on $q$ queries.



## *Global sensitivity framework* [DMNS06]

*Intuition:* $f$ can be released accurately when it is insensitive to individual entries $x_1, \ldots, x_n$.

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

*Example:* $\mathsf{GS}_{\text{average}} = \frac{1}{n}$ if $x \in [0,1]^n$.

**Theorem**

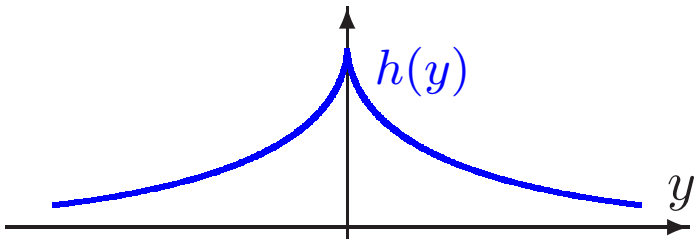
*If* $A(x) = f(x) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$ *then* $A$ *is* $\varepsilon$*-differentially private.*

# *Global sensitivity: noise distribution*

**Theorem**

*If* $A(x) = f(x) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$ *then* $A$ *is* $\varepsilon$*-diff. private.*

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

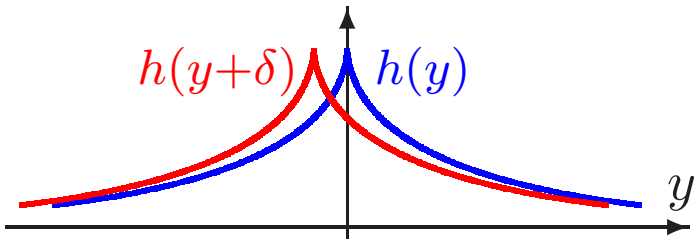


# *Global sensitivity: noise distribution*

**Theorem**

*If* $A(x) = f(x) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$ *then* $A$ *is* $\varepsilon$*-diff. private.*

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



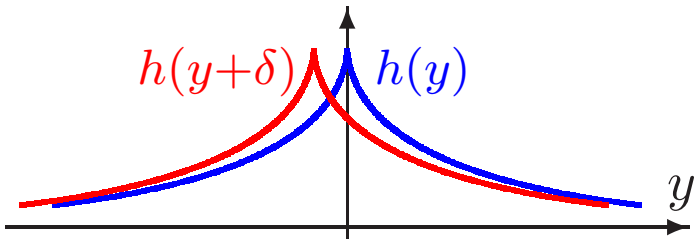

Sliding property of $\mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$: $\frac{h(y)}{h(y+\delta)} \leq e^{\varepsilon \cdot \frac{\|\delta\|}{\mathsf{GS}_f}}$ for all $y, \delta$

# *Global sensitivity: noise distribution*

**Theorem**

*If* $A(x) = f(x) + \mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$ *then* $A$ *is* $\varepsilon$*-diff. private.*

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




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

*Proof idea:*

$A(x)$: blue curve

$A(x')$: red curve

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

## *Functions with low* GS [DMNS06]

*Example:* $\mathsf{GS}_{\text{average}} = \frac{1}{n}$ if $x \in [0,1]^n$. Noise $= \mathsf{Lap}\left(\frac{1}{\varepsilon n}\right)$.
*Compare to:* Estimating frequencies (e.g., proportion of people with blue eyes) from $n$ samples: sampling error $\frac{1}{\sqrt{n}}$.

- Means, variances
  - for data in a bounded interval
- histograms, contingency tables
- singular value decomposition
- distance to a property
- functions that can be approximated well on any $x$ by a small random sample of $x_i$'s

## *Road map*

I. Introduction

- Definitions
- Global sensitivity framework [DMNS06]

II. Smooth sensitivity framework

- Motivating examples
- The framework
- Computing and approximating smooth sensitivity

III. Sample-and-Aggregate framework

# *High global sensitivity: example 1*

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

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

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

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

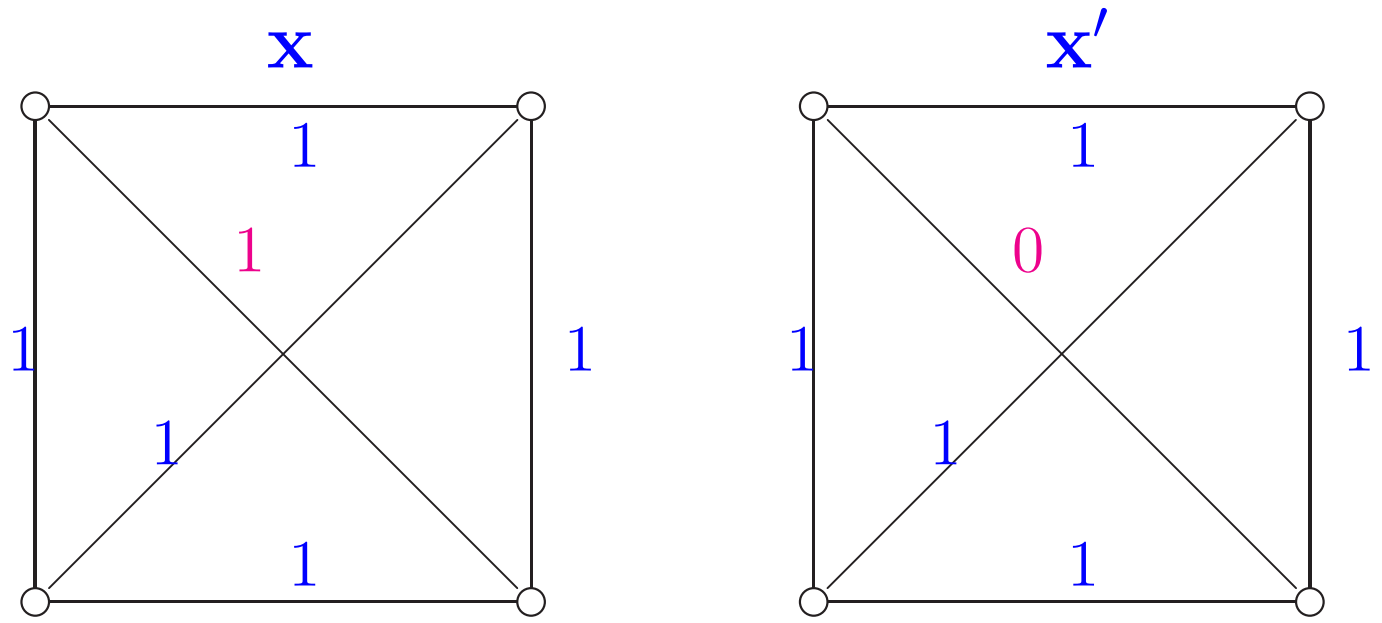
- Noise magnitude: $\frac{1}{\varepsilon}$. Too much noise!
- But for most neighbor databases $x, x'$,

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

# *High global sensitivity: example 2*

*Example 2: the weight of a minimum spanning tree*

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

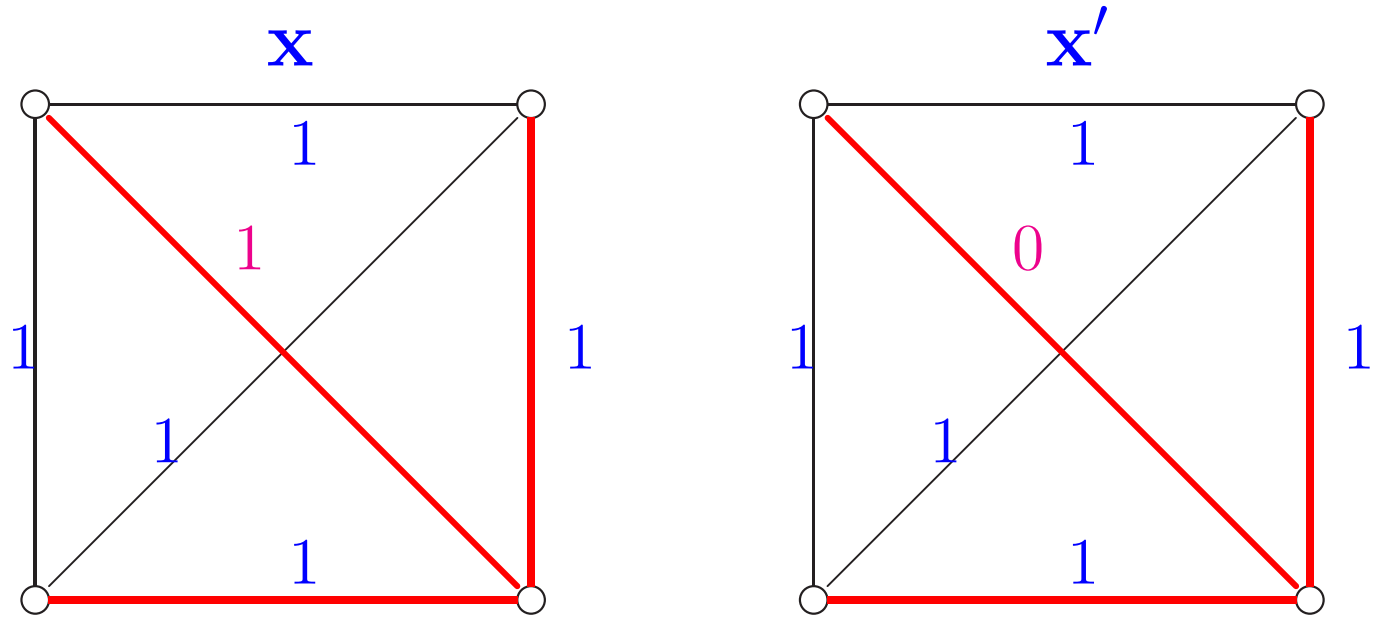



$$\mathsf{GS}_{\text{MST-weight}} = 1$$

# *High global sensitivity: example 2*

*Example 2: the weight of a minimum spanning tree*

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

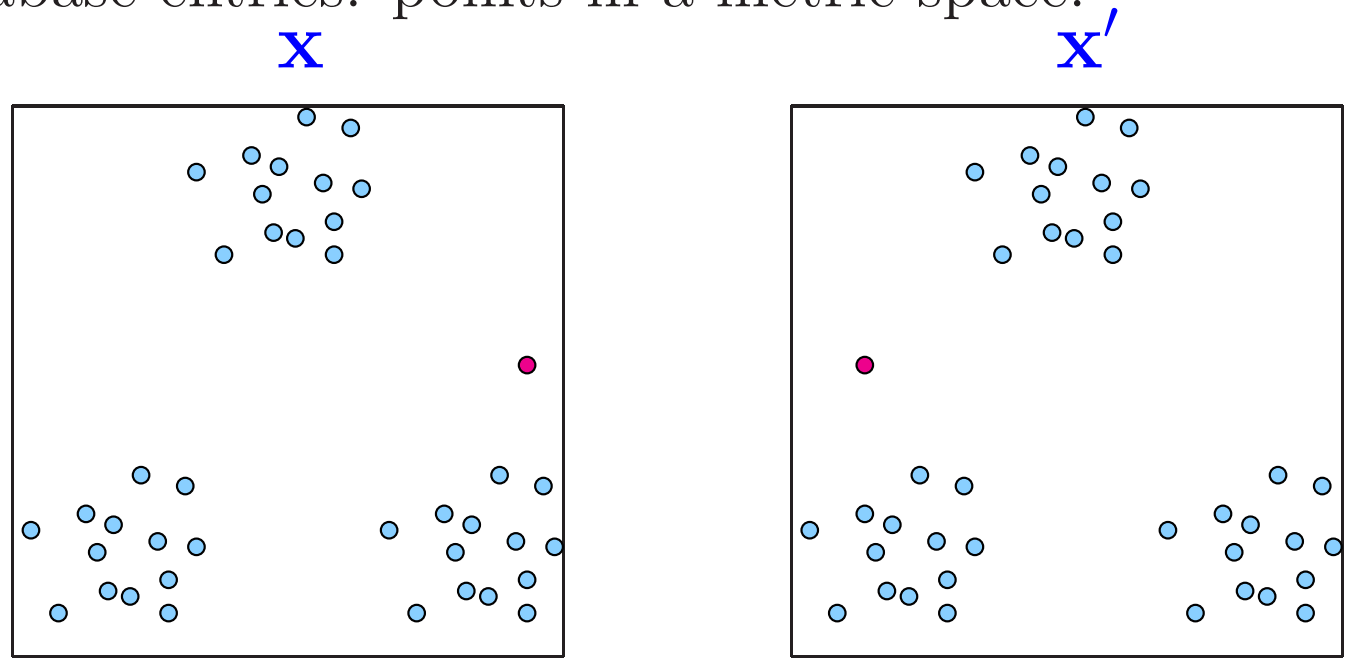



$$\mathsf{GS}_{\text{MST-weight}} = 1$$

# *High global sensitivity: example 3*

*Example 3: cluster centers*

Database entries: points in a metric space.



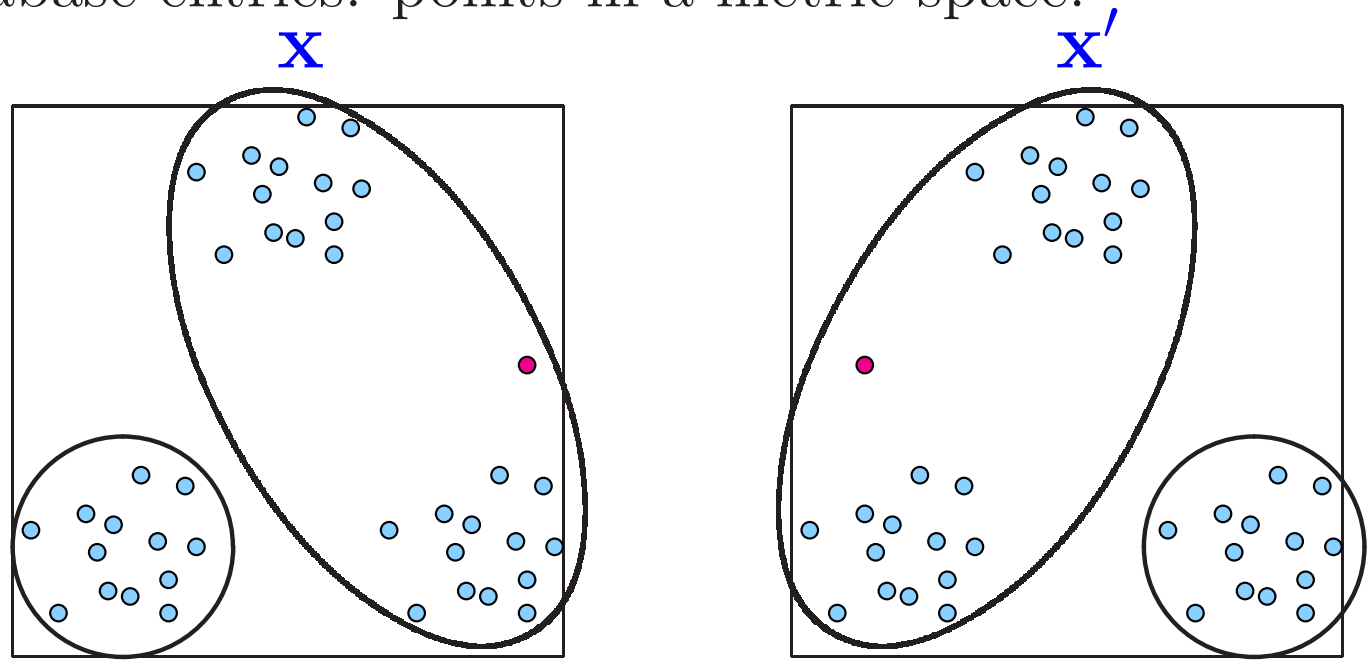

Global sensitivity of cluster centers is roughly the diameter of the space.

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

# *High global sensitivity: example 3*

*Example 3: cluster centers*

Database entries: points in a metric space.



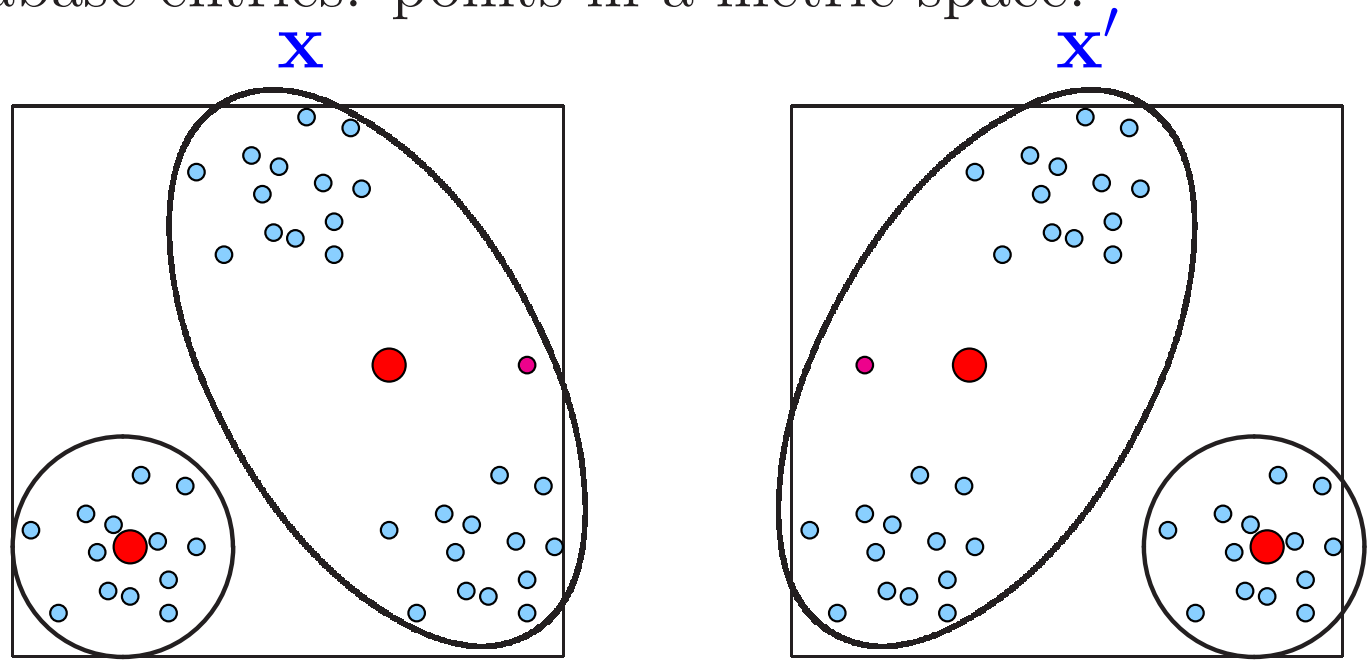

Global sensitivity of cluster centers is roughly the diameter of the space.

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

# *High global sensitivity: example 3*

*Example 3: cluster centers*

Database entries: points in a metric space.




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

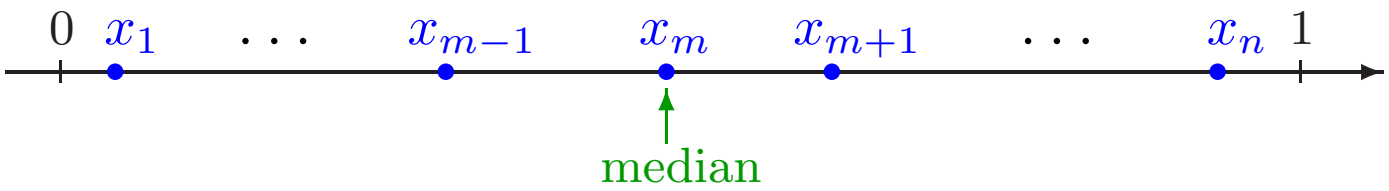
- But intuitively, if clustering is "good", cluster centers should be insensitive.

# *Local sensitivity*

Local sensitivity $\mathsf{LS}_f(x) = \max_{x' \text{: neighbor of } x} \|f(x) - f(x')\|_1$

*Reminder:* $\mathsf{GS}_f = \max_x \mathsf{LS}_f(x)$

*Example: median* for $0 \le x_1 \le \cdots \le x_n \le 1$, odd $n$




$$\mathsf{LS}_{\text{median}}(x) = \max(x_m - x_{m-1}, x_{m+1} - x_m)$$

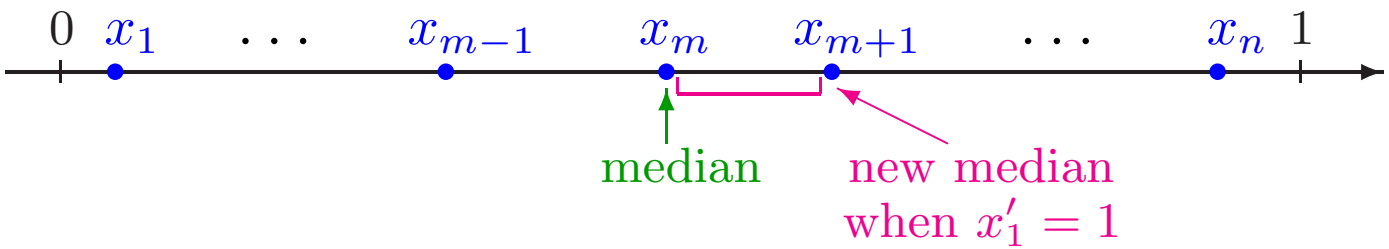
*Goal:* Release $f(x)$ with less noise when $\mathsf{LS}_f(x)$ is lower.

# *Local sensitivity*

Local sensitivity $\mathsf{LS}_f(x) = \max_{x' \text{: neighbor of } x} \|f(x) - f(x')\|_1$

*Reminder:* $\mathsf{GS}_f = \max_x \mathsf{LS}_f(x)$

*Example: median* for $0 \le x_1 \le \cdots \le x_n \le 1$, odd $n$




$$\mathsf{LS}_{\text{median}}(x) = \max(x_m - x_{m-1}, x_{m+1} - x_m)$$

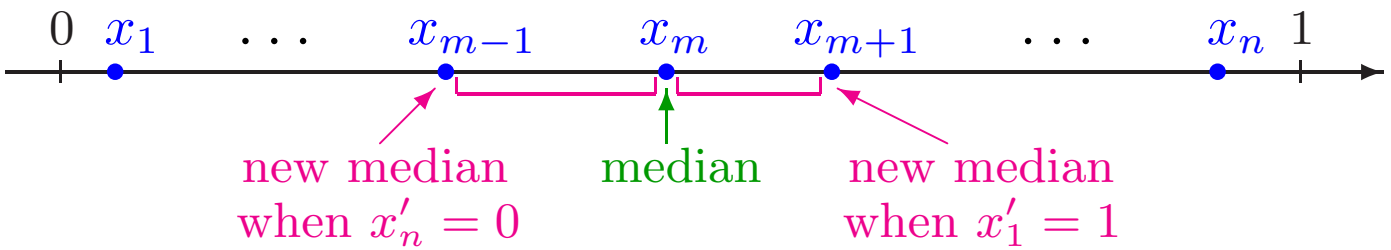
*Goal:* Release $f(x)$ with less noise when $\mathsf{LS}_f(x)$ is lower.

# *Local sensitivity*

Local sensitivity $\mathsf{LS}_f(x) = \max_{x' \text{: neighbor of } x} \|f(x) - f(x')\|_1$

*Reminder:* $\mathsf{GS}_f = \max_x \mathsf{LS}_f(x)$

*Example: median* for $0 \le x_1 \le \cdots \le x_n \le 1$, odd $n$




$$\mathsf{LS}_{\text{median}}(x) = \max(x_m - x_{m-1}, x_{m+1} - x_m)$$

*Goal:* Release $f(x)$ with less noise when $\mathsf{LS}_f(x)$ is lower.

## *Instance-based noise: first attempt*

Can we have noise magnitude $\propto \mathsf{LS}_f(x)$ instead of $\mathsf{GS}_f$?

*Problem:* Noise magnitude might reveal information.

*Example: median*

$$x = \underbrace{0 \cdots 0}_{\frac{n-3}{2}} 000 \underbrace{1 \cdots 1}_{\frac{n-3}{2}} \qquad x' = \underbrace{0 \cdots 0}_{\frac{n-3}{2}} 001 \underbrace{1 \cdots 1}_{\frac{n-3}{2}}$$

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

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

$$\Pr[A(x) = 0] = 1 \qquad \Pr[A(x') = 0] = 0$$

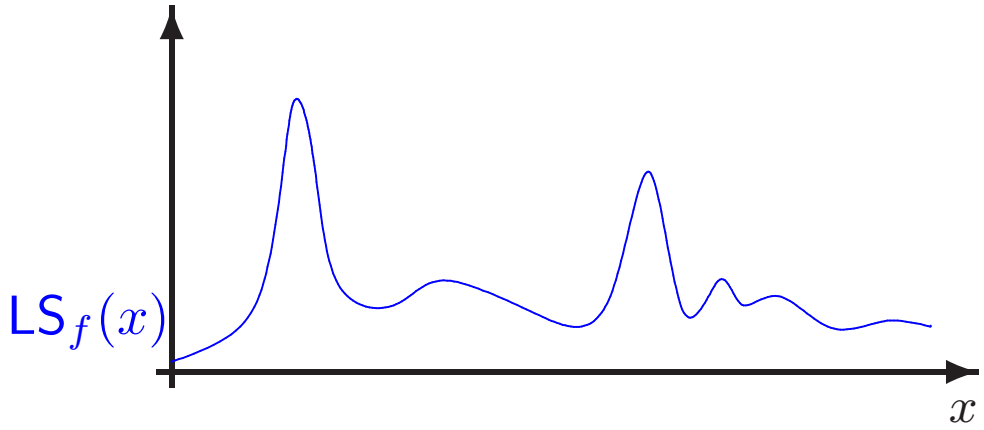
$A$ is not $\varepsilon$-differentially private

*Lesson:* Noise magnitude must be an insensitive function.

## *Smooth bounds on local sensitivity*

Design sensitivity function $S(x)$

- $S(x)$ is an $\varepsilon$-smooth upper bound on $\mathsf{LS}_f(x)$ if:
  - for all $x$: $S(x) \geq \mathsf{LS}_f(x)$
  - for all neighbors $x, x'$ : $S(x) \leq e^{\varepsilon} S(x')$




**Theorem**

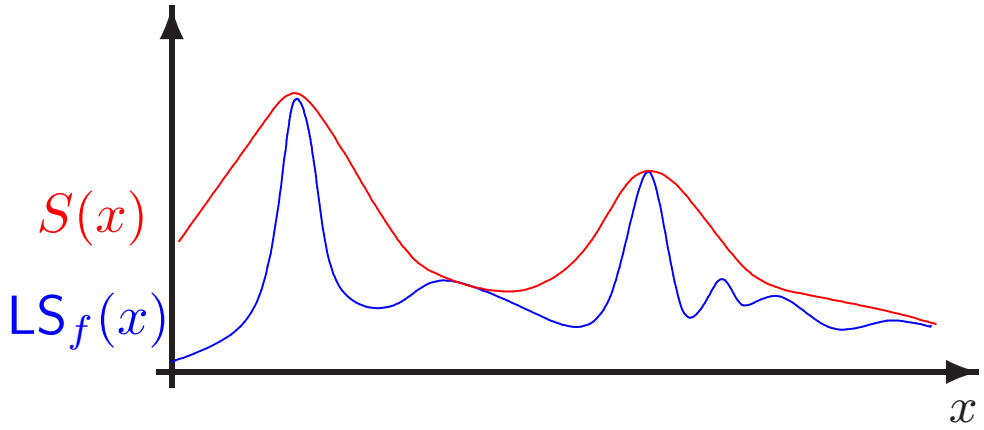
*If* $A(x) = f(x) + \mathsf{noise}\left(\dfrac{S(x)}{\varepsilon}\right)$ *then* $A$ *is* $\varepsilon'$*-differentially private.*

*Example:* $\mathsf{GS}_f$ is always a smooth bound on $\mathsf{LS}_f(x)$

## *Smooth bounds on local sensitivity*

Design sensitivity function $S(x)$

- $S(x)$ is an $\varepsilon$-smooth upper bound on $\mathsf{LS}_f(x)$ if:
  - for all $x$: $S(x) \geq \mathsf{LS}_f(x)$
  - for all neighbors $x, x'$ : $S(x) \leq e^{\varepsilon} S(x')$




**Theorem**

*If* $A(x) = f(x) + \mathsf{noise}\left(\dfrac{S(x)}{\varepsilon}\right)$ *then* $A$ *is* $\varepsilon'$*-differentially private.*

*Example:* $\mathsf{GS}_f$ is always a smooth bound on $\mathsf{LS}_f(x)$

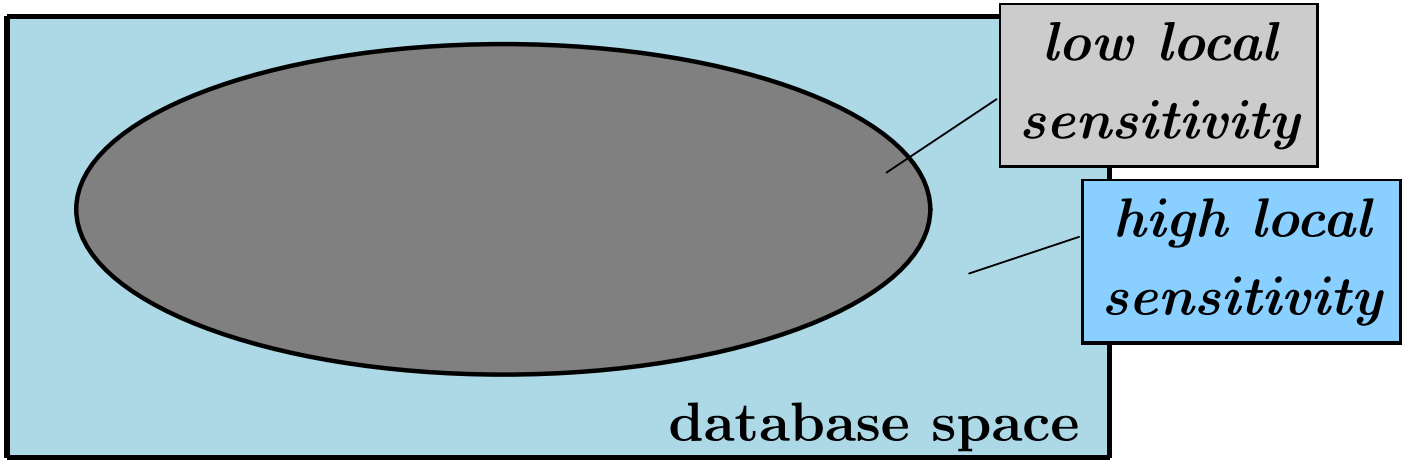
# *Smooth Sensitivity*

Smooth sensitivity $\mathsf{S}^*_f(x) = \max_y \left(\mathsf{LS}_f(y) e^{-\varepsilon \cdot \mathsf{dist}(x,y)}\right)$

**Lemma**

*For every $\varepsilon$-smooth bound $S$:* $\mathsf{S}^*_f(x) \leq S(x)$ *for all $x$.*

*Intuition:* little noise when **far** from sensitive instances



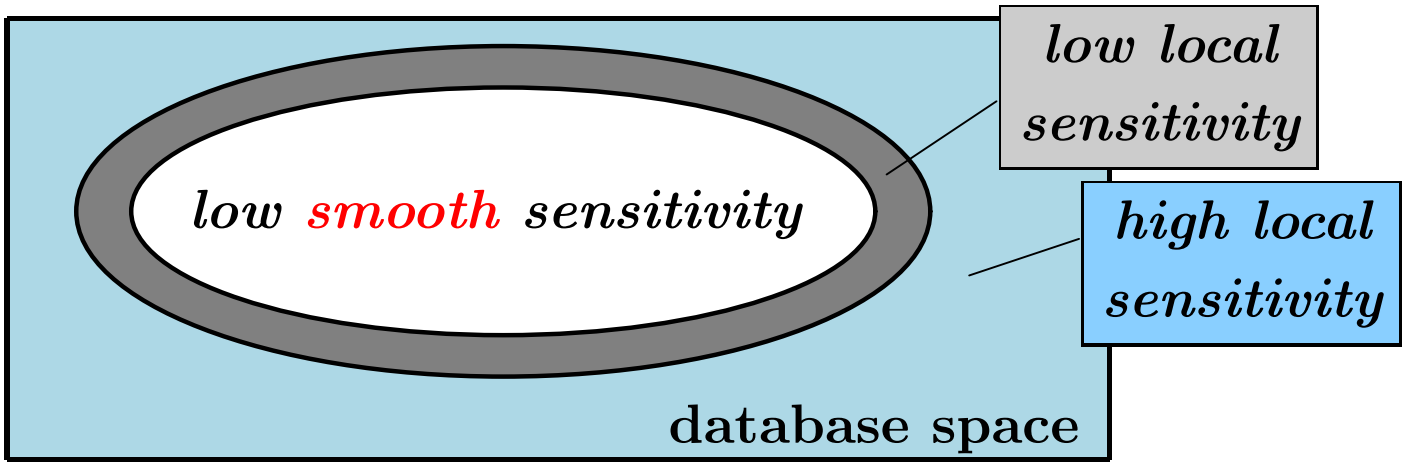
# *Smooth Sensitivity*

Smooth sensitivity $\mathsf{S}^*_f(x) = \max_y \left(\mathsf{LS}_f(y) e^{-\varepsilon \cdot \mathsf{dist}(x,y)}\right)$

**Lemma**

*For every $\varepsilon$-smooth bound $S$:* $\mathsf{S}^*_f(x) \leq S(x)$ *for all $x$.*

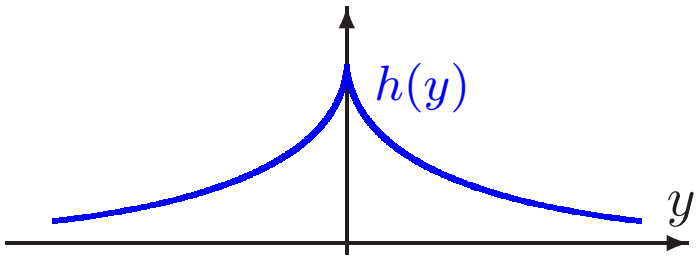
*Intuition:* little noise when **far** from sensitive instances



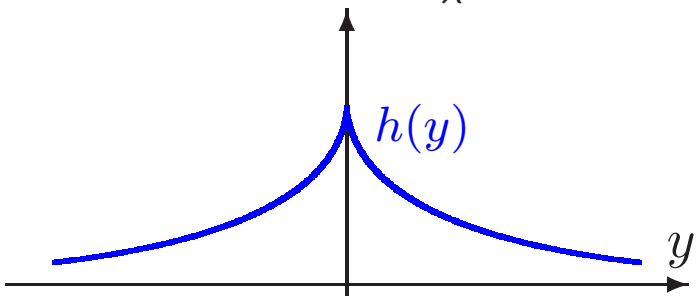
# *Admissible noise distributions*

Two requirements on density $h$ of the noise distribution

1. Sliding property: for all $y, \delta$ : $\frac{h(y)}{h(y+\delta)} \leq e^{\alpha \|\delta\|_1}$. [DMNS06]



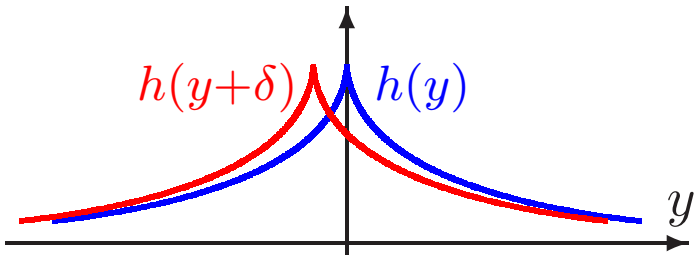

2. Scaling property: for all $y, \lambda$ : $\frac{h(y)}{\frac{1}{\lambda} h(\lambda y)} \leq \lambda^{\alpha}$.



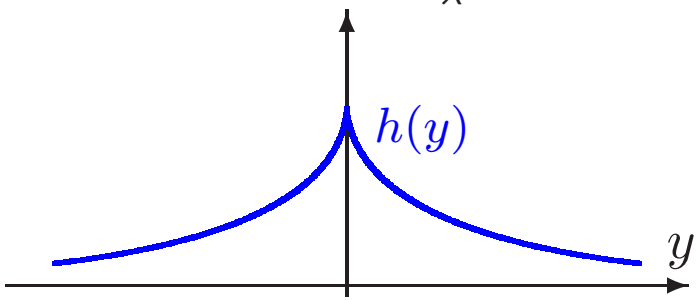
# *Admissible noise distributions*

Two requirements on density $h$ of the noise distribution

1. Sliding property: for all $y, \delta$ : $\frac{h(y)}{h(y+\delta)} \leq e^{\alpha \|\delta\|_1}$. [DMNS06]



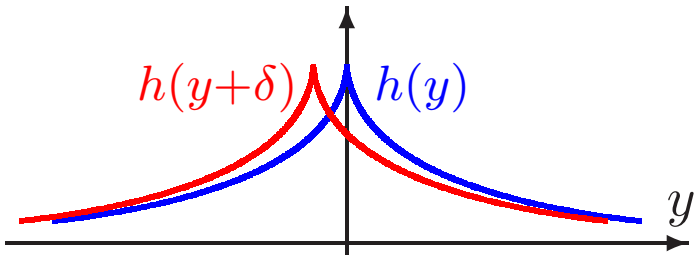

2. Scaling property: for all $y, \lambda$ : $\frac{h(y)}{\frac{1}{\lambda} h(\lambda y)} \leq \lambda^{\alpha}$.



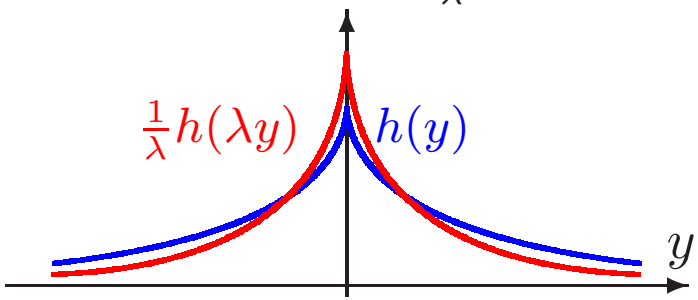
# *Admissible noise distributions*

Two requirements on density $h$ of the noise distribution

1. Sliding property: for all $y, \delta$ : $\frac{h(y)}{h(y+\delta)} \leq e^{\alpha \|\delta\|_1}$. [DMNS06]




2. Scaling property: for all $y, \lambda$ : $\frac{h(y)}{\frac{1}{\lambda} h(\lambda y)} \leq \lambda^{\alpha}$.



# *Admissible noise distributions: examples*

There are "nice" admissible noise distributions:

(a) *heavy-tail* (density $h(y) \propto \frac{1}{1+y^c}$) and

(b) *Laplacian* or *Gaussian*

for all "interesting" metrics on the output space

– e.g., $L_1^d$,$L_2^d$, earthmover

# *Computing smooth sensitivity*

*Recall:* Smooth sensitivity $\mathsf{S}^*_f(x) = \max_y \left(\mathsf{LS}_f(y) e^{-\varepsilon \cdot \mathsf{dist}(x,y)}\right)$

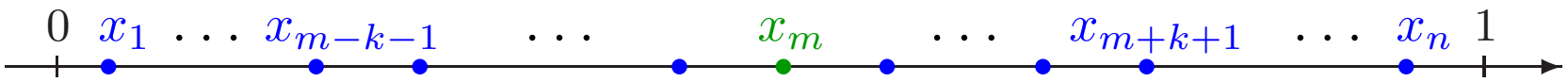
**Observation**

$$\mathsf{S}^*_f(x) = \max_{k=0,1,\ldots,n} e^{-k\varepsilon} \cdot \mathsf{LS}^k_f(x)$$

$$\text{where } \mathsf{LS}^k_f(x) = \max_{y:\mathsf{dist}(x,y)\leq k} \mathsf{LS}_f(y).$$

*Example: median*

$$\mathsf{LS}^k_{\text{median}}(x) = \max_{t=0,1,\ldots,k+1} (x_{m+t+k+1} - x_{m+t})$$



# *Computing smooth sensitivity*

*Recall:* Smooth sensitivity $\mathsf{S}^*_f(x) = \max_y \left(\mathsf{LS}_f(y) e^{-\varepsilon \cdot \mathsf{dist}(x,y)}\right)$

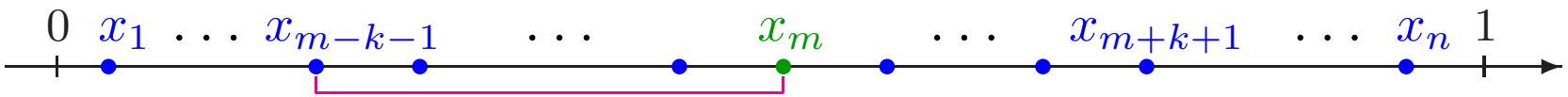
**Observation**

$$\mathsf{S}^*_f(x) = \max_{k=0,1,\ldots,n} e^{-k\varepsilon} \cdot \mathsf{LS}^k_f(x)$$

$$\textit{where } \mathsf{LS}^k_f(x) = \max_{y:\mathsf{dist}(x,y)\le k} \mathsf{LS}_f(y).$$

*Example: median*

$$\mathsf{LS}^k_{\text{median}}(x) = \max_{t=0,1,\ldots,k+1} (x_{m+t+k+1} - x_{m+t})$$



# *Computing smooth sensitivity*

*Recall:* Smooth sensitivity $\mathsf{S}^*_f(x) = \max_y \left(\mathsf{LS}_f(y) e^{-\varepsilon \cdot \mathsf{dist}(x,y)}\right)$

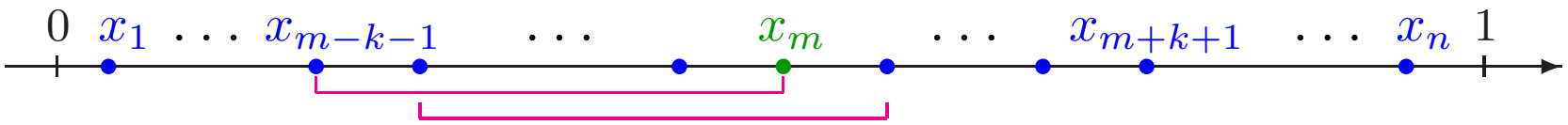
**Observation**

$$\mathsf{S}^*_f(x) = \max_{k=0,1,\ldots,n} e^{-k\varepsilon} \cdot \mathsf{LS}^k_f(x)$$

$$\textit{where } \mathsf{LS}^k_f(x) = \max_{y:\mathsf{dist}(x,y)\le k} \mathsf{LS}_f(y).$$

*Example: median*

$$\mathsf{LS}^k_{\text{median}}(x) = \max_{t=0,1,\ldots,k+1} (x_{m+t+k+1} - x_{m+t})$$



# *Computing smooth sensitivity*

*Recall:* Smooth sensitivity $\mathsf{S}^*_f(x) = \max_y \left(\mathsf{LS}_f(y) e^{-\varepsilon \cdot \mathsf{dist}(x,y)}\right)$

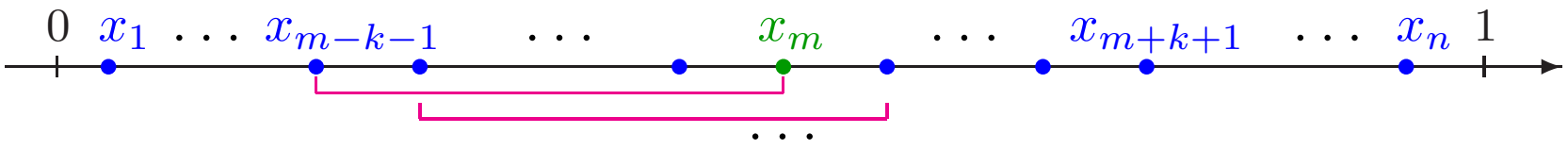
**Observation**

$$\mathsf{S}^*_f(x) = \max_{k=0,1,\ldots,n} e^{-k\varepsilon} \cdot \mathsf{LS}^k_f(x)$$

$$\textit{where } \mathsf{LS}^k_f(x) = \max_{y:\mathsf{dist}(x,y)\le k} \mathsf{LS}_f(y).$$

*Example: median*

$$\mathsf{LS}^k_{\text{median}}(x) = \max_{t=0,1,\ldots,k+1} (x_{m+t+k+1} - x_{m+t})$$



# *Computing smooth sensitivity*

*Recall:* Smooth sensitivity $\mathsf{S}^*_f(x) = \max_y \left(\mathsf{LS}_f(y) e^{-\varepsilon \cdot \mathsf{dist}(x,y)}\right)$

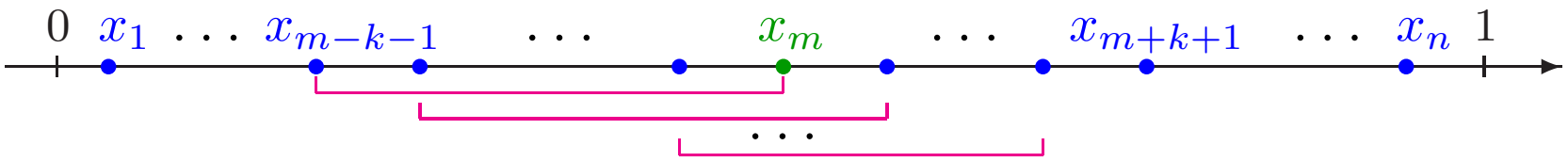
**Observation**

$$\mathsf{S}^*_f(x) = \max_{k=0,1,\ldots,n} e^{-k\varepsilon} \cdot \mathsf{LS}^k_f(x)$$

$$\textit{where } \mathsf{LS}^k_f(x) = \max_{y:\mathsf{dist}(x,y)\le k} \mathsf{LS}_f(y).$$

*Example: median*

$$\mathsf{LS}^k_{\text{median}}(x) = \max_{t=0,1,\ldots,k+1} (x_{m+t+k+1} - x_{m+t})$$



# *Computing smooth sensitivity*

*Recall:* Smooth sensitivity $\mathsf{S}^*_f(x) = \max_y \left(\mathsf{LS}_f(y) e^{-\varepsilon \cdot \mathsf{dist}(x,y)}\right)$

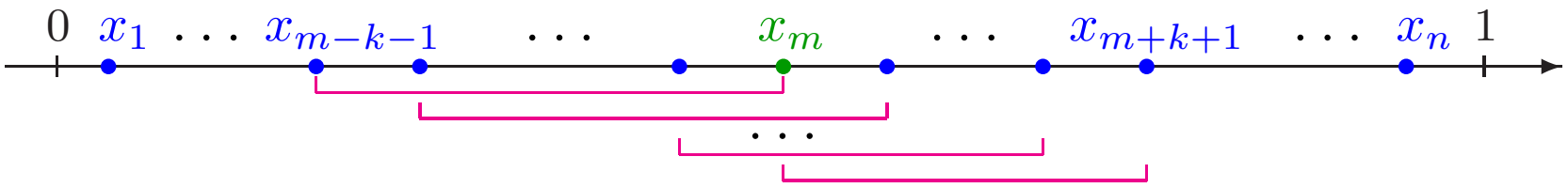
**Observation**

$$\mathsf{S}^*_f(x) = \max_{k=0,1,\ldots,n} e^{-k\varepsilon} \cdot \mathsf{LS}^k_f(x)$$

$$\textit{where } \mathsf{LS}^k_f(x) = \max_{y:\mathsf{dist}(x,y)\le k} \mathsf{LS}_f(y).$$

*Example: median*

$$\mathsf{LS}^k_{\text{median}}(x) = \max_{t=0,1,\ldots,k+1} (x_{m+t+k+1} - x_{m+t})$$




- $\mathsf{S}^*_{\text{median}}(x)$ is computable in $O(n^2)$ time.

## *Computing smooth sensitivity*

Example functions with computable smooth sensitivity

- *Median* & *minimum* of numbers in a bounded interval
- *MST cost* when weights are bounded
- *Number of triangles* in a graph

Approximating smooth sensitivity

- only smooth upper bounds on $\mathsf{LS}$ are meaningful
- simple generic methods for smooth approximations
  - work for *median* and *1-median in* $L_1^d$
  - for *median* on presorted inputs they yeild
    - a factor of 2 approximation in $O(n)$ time
    - $\pm\frac{1}{\text{poly}\,(n)}$ approximation in $O\left(\frac{\log^2 n}{\varepsilon^2}\right)$ time
  - for *1-median in* $L_1^d$*:* $d$-approximation in time $\tilde{O}(nd)$

## *New goal*

- Smooth sensitivity framework requires understanding combinatorial structure of $f$

  – hard in general

- Goal: an automatable transformation from an arbitrary $f$ into an $\varepsilon$-diff. private $A$

  – $A(x) \approx f(x)$ for "good" instances $x$

## *Road map*

I. Introduction
- Definitions
- Global sensitivity framework [DMNS06]

II. Smooth sensitivity framework
- Motivating examples
- The framework
- Computing and approximating smooth sensitivity

III. Sample-and-Aggregate framework

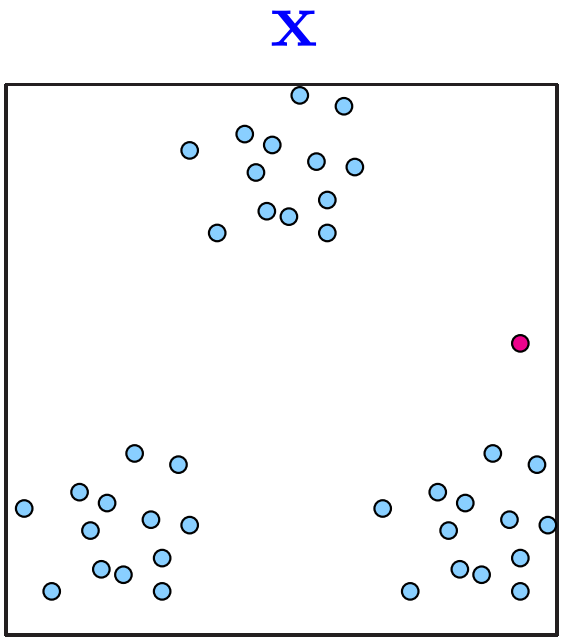
# *What about cluster centers?*

*Example: k-means cluster centers*

Database entries $x_i$: points in $\mathbb{R}^d$

Output $f(x)$: $k$ centers $(c_1, \ldots, c_k)$ that minimize the cost

$$\text{cost} = \sum_{i \in \{1,\ldots,n\}} \|x_i - c_j\|_2^2, \text{ where } c_j = \text{nearest center to } x_i$$



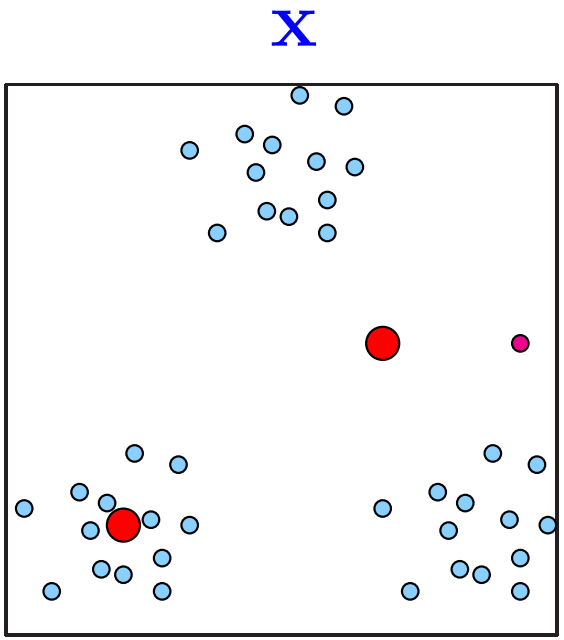
# *What about cluster centers?*

*Example: k-means cluster centers*

Database entries $x_i$: points in $\mathbb{R}^d$

Output $f(x)$: $k$ centers $(c_1, \ldots, c_k)$ that minimize the cost

$$\text{cost} = \sum_{i \in \{1,\ldots,n\}} \|x_i - c_j\|_2^2, \text{ where } c_j = \text{nearest center to } x_i$$



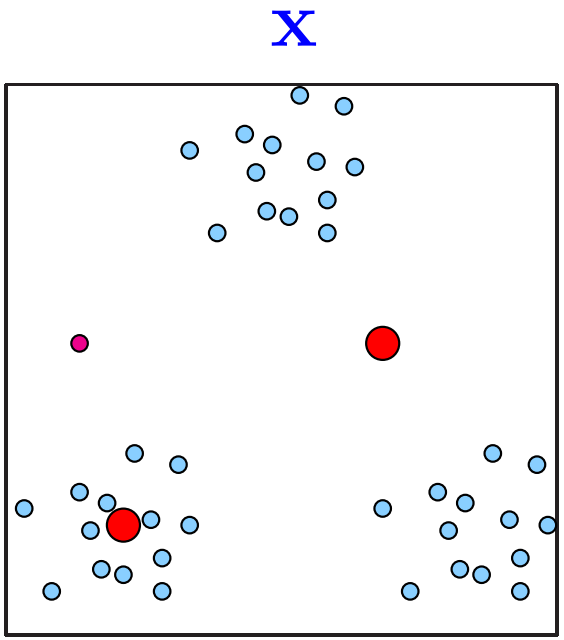
# *What about cluster centers?*

*Example: k-means cluster centers*

Database entries $x_i$: points in $\mathbb{R}^d$

Output $f(x)$: $k$ centers $(c_1, \ldots, c_k)$ that minimize the cost

$$\text{cost} = \sum_{i \in \{1,\ldots,n\}} \|x_i - c_j\|_2^2, \text{ where } c_j = \text{nearest center to } x_i$$



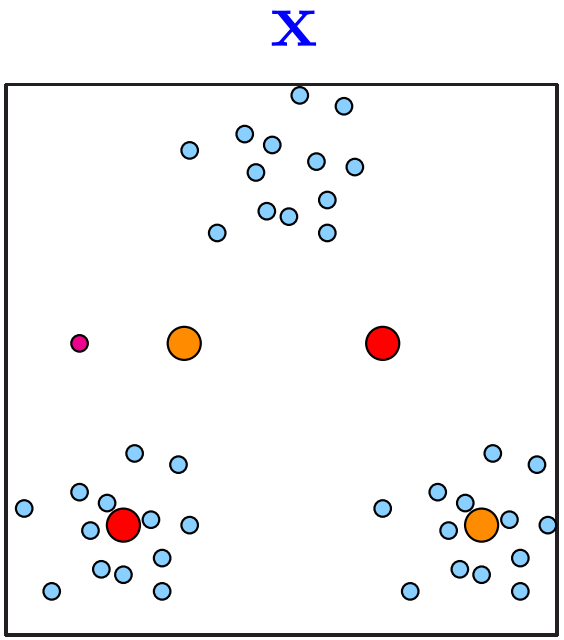
# *What about cluster centers?*

*Example: k-means cluster centers*

Database entries $x_i$: points in $\mathbb{R}^d$

Output $f(x)$: $k$ centers $(c_1, \ldots, c_k)$ that minimize the cost

$$\text{cost} = \sum_{i \in \{1,\ldots,n\}} \|x_i - c_j\|_2^2, \text{ where } c_j = \text{nearest center to } x_i$$



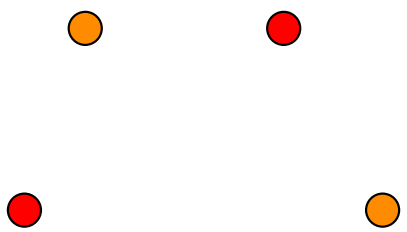
## *What about cluster centers?*

*Example: k-means cluster centers*

Database entries $x_i$: points in $\mathbb{R}^d$

Output $f(x)$: $k$ centers $(c_1, \ldots, c_k)$ that minimize the cost

$$\text{cost} = \sum_{i \in \{1,\ldots,n\}} \|x_i - c_j\|_2^2, \text{ where } c_j = \text{nearest center to } x_i$$



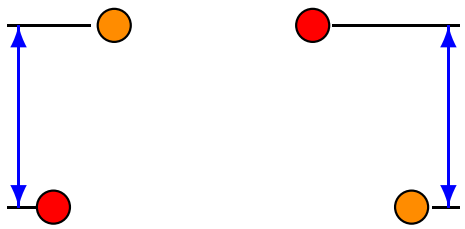
## *What about cluster centers?*

*Example: k-means cluster centers*

Database entries $x_i$: points in $\mathbb{R}^d$

Output $f(x)$: $k$ centers $(c_1, \ldots, c_k)$ that minimize the cost

$$\text{cost} = \sum_{i \in \{1,\ldots,n\}} \|x_i - c_j\|_2^2, \text{ where } c_j = \text{nearest center to } x_i$$



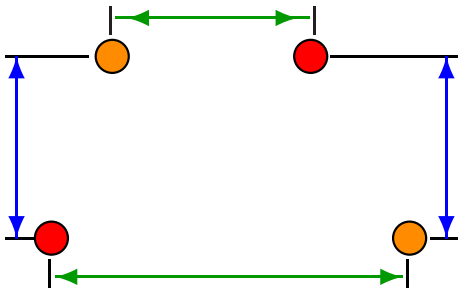
## *What about cluster centers?*

*Example: k-means cluster centers*

Database entries $x_i$: points in $\mathbb{R}^d$

Output $f(x)$: $k$ centers $(c_1, \ldots, c_k)$ that minimize the cost

$$\text{cost} = \sum_{i \in \{1,\ldots,n\}} \|x_i - c_j\|_2^2, \text{ where } c_j = \text{nearest center to } x_i$$



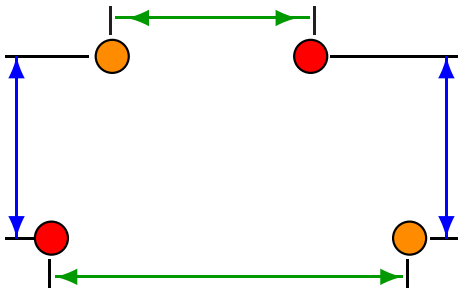
## *What about cluster centers?*

*Example: k-means cluster centers*

Database entries $x_i$: points in $\mathbb{R}^d$

Output $f(x)$: $k$ centers $(c_1, \ldots, c_k)$ that minimize the cost

$$\text{cost} = \sum_{i \in \{1,\ldots,n\}} \|x_i - c_j\|_2^2, \text{ where } c_j = \text{nearest center to } x_i$$



Earthmover distance

$$\mathsf{dist}(c, c') = \min_{\pi} \|(c_1, \ldots, c_k), (c'_{\pi(1)}, \ldots, c'_{\pi(k)})\|_2$$

## *Sample-and-aggregate method*

**Theorem**

*If $f$ can be approximated on $x$*
*from small samples*
*then $f$ can be released with little noise*

## *Sample-and-aggregate method*

**Theorem**

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

## *Sample-and-aggregate method*

**Theorem**

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

- Works in all "interesting" metric spaces
- Example applications
  - $k$-means cluster centers (if data is separated a.k.a. [Ostrovsky Rabani Schulman Swamy 06])
  - fitting mixtures of Gaussians (if data is i.i.d., using [Achlioptas McSherry 05])
  - PAC concepts (if uniquely learnable,
    i.e., if learning algorithm always outputs
    the same hypothesis or something close to it)
    general [Kasiviswanathan Lee Nissim Raskhodnikova Smith]

## *Sampling in the* GS *framework*

**Lemma [DMNS06]**

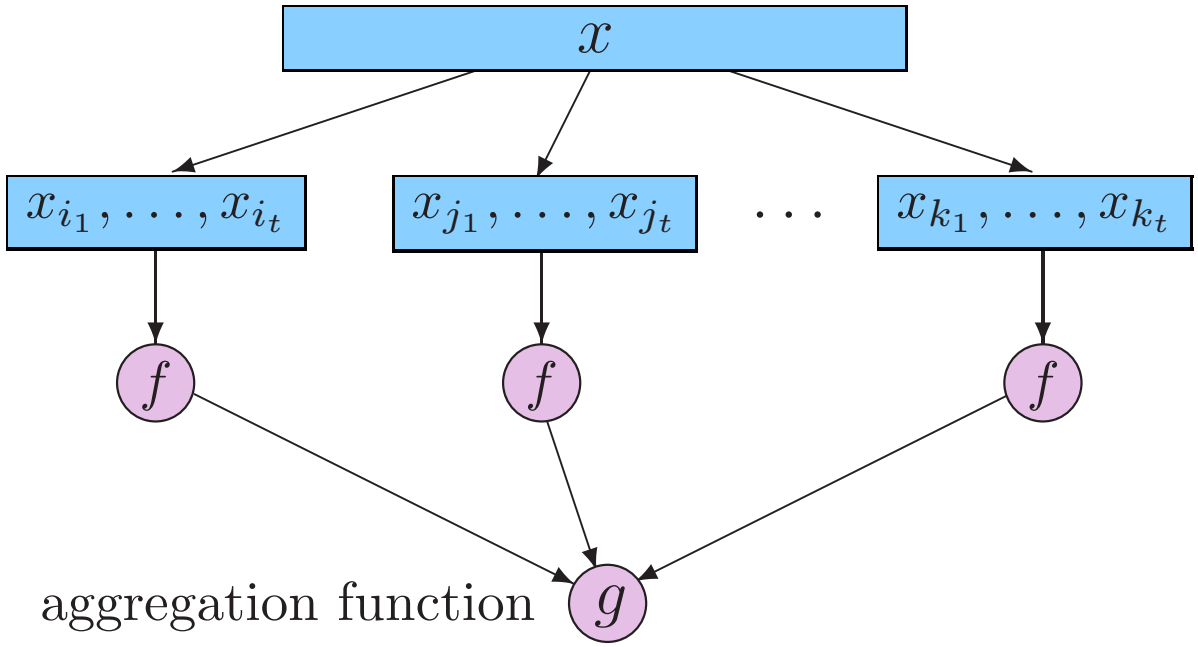
*If $f$ can be approximated on all $x$ within distance $r$ from samples of size $\alpha n$ then* $\mathsf{GS}_f \leq r$.

- need guarantee for all instances
- only a proof technique to bound $\mathsf{GS}_f$
- our result is algorithmic

# *Sample-and-Aggregate*

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

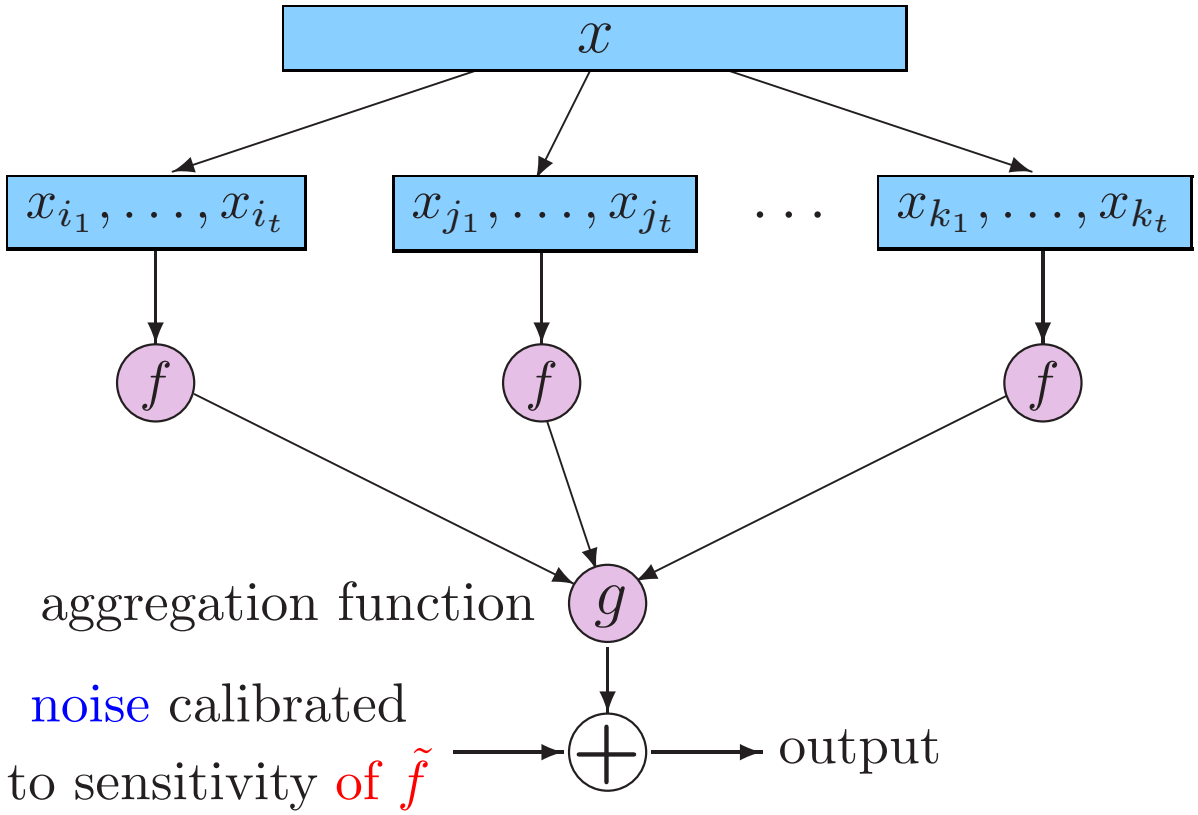
$$\tilde{f}(x) = g(f(sample_1), f(sample_2), \ldots, f(sample_s))$$



# *Sample-and-Aggregate*

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

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



## *Good aggregation functions* $g$

- $g(z) = \text{average}(z)$
  - works for $L_1$ and $L_2$
- $g(z) = \text{center of attention of } z$
  - works for arbitrary metrics
    (in particular, for Earthmover)
  - gives lower noise for $L_1$ and $L_2$

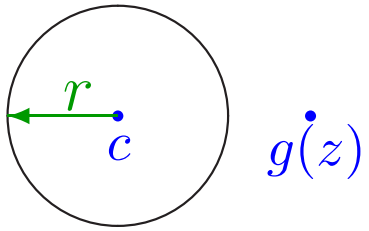
## *The center of attention*

The center of attention, $g(z_1, \ldots, z_s)$, is the center of the smallest ball containing $\geq t_0$ of $z_i$'s, where $t_0 = \frac{s+1}{2} + 1$.

- Related concept, different use: $t$-radius of a point = the distance to its $t$-th nearest neighbor. [CDMSW05]

**Claim: $g$ is a good aggregation**

*If $\frac{2}{3}$ of $z_i$'s $\in B(c, r)$ then $g(z) \in B(c, 2r)$.*



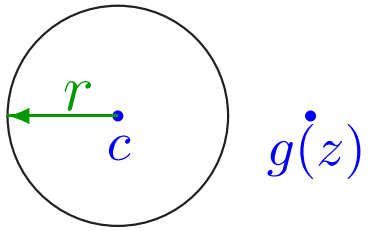
## *The center of attention*

The center of attention, $g(z_1, \ldots, z_s)$, is the center of the smallest ball containing $\geq t_0$ of $z_i$'s, where $t_0 = \frac{s+1}{2} + 1$.

- Related concept, different use: $t$-radius of a point = the distance to its $t$-th nearest neighbor. [CDMSW05]

**Claim: $g$ is a good aggregation**

*If $\frac{2}{3}$ of $z_i$'s $\in B(c, r)$ then $g(z) \in B(c, 2r)$. Think $c = f(x)$.*



## *The center of attention*

The center of attention, $g(z_1, \ldots, z_s)$, is the center of the smallest ball containing $\geq t_0$ of $z_i$'s, where $t_0 = \frac{s+1}{2} + 1$.

- Related concept, different use: $t$-radius of a point = the distance to its $t$-th nearest neighbor. [CDMSW05]

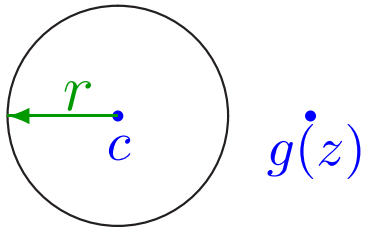
**Claim: $g$ is a good aggregation**

*If $\frac{2}{3}$ of $z_i$'s $\in B(c, r)$ then $g(z) \in B(c, 2r)$. Think $c = f(x)$.*

*Proof:* Let $r_{t_0}$ = smallest $t_0$-radius.

$c$ has $t_0$-radius at most $r$.

$r_{t_0} \leq r$. $\square$



## *The center of attention*

The center of attention, $g(z_1, \ldots, z_s)$, is the center of the smallest ball containing $\geq t_0$ of $z_i$'s, where $t_0 = \frac{s+1}{2} + 1$.

- Related concept, different use: $t$-radius of a point = the distance to its $t$-th nearest neighbor. [CDMSW05]

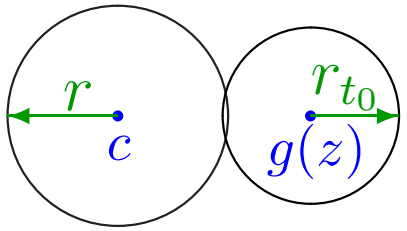
**Claim: $g$ is a good aggregation**

*If $\frac{2}{3}$ of $z_i$'s $\in B(c, r)$ then $g(z) \in B(c, 2r)$. Think $c = f(x)$.*

*Proof:* Let $r_{t_0}$ = smallest $t_0$-radius.

$c$ has $t_0$-radius at most $r$.

$r_{t_0} \leq r$. $\square$



## *The center of attention*

The center of attention, $g(z_1, \ldots, z_s)$, is the center of the smallest ball containing $\geq t_0$ of $z_i$'s, where $t_0 = \frac{s+1}{2} + 1$.

- Related concept, different use: $t$-radius of a point = the distance to its $t$-th nearest neighbor. [CDMSW05]

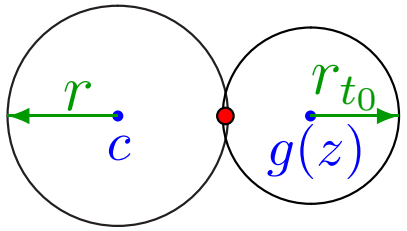
**Claim: $g$ is a good aggregation**

*If $\frac{2}{3}$ of $z_i$'s $\in B(c, r)$ then $g(z) \in B(c, 2r)$. Think $c = f(x)$.*

*Proof:* Let $r_{t_0}$ = smallest $t_0$-radius.

$c$ has $t_0$-radius at most $r$.

$r_{t_0} \leq r$. $\square$



## *The center of attention*

The center of attention, $g(z_1, \ldots, z_s)$, is the center of the smallest ball containing $\geq t_0$ of $z_i$'s, where $t_0 = \frac{s+1}{2} + 1$.

- Related concept, different use: $t$-radius of a point = the distance to its $t$-th nearest neighbor. [CDMSW05]

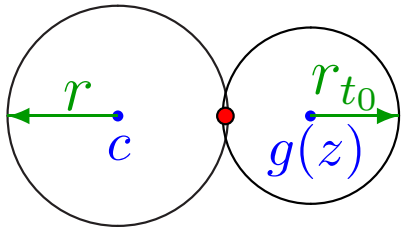
**Claim: $g$ is a good aggregation**

*If $\frac{2}{3}$ of $z_i$'s $\in B(c, r)$ then $g(z) \in B(c, 2r)$. Think $c = f(x)$.*

*Proof:* Let $r_{t_0}$ = smallest $t_0$-radius.

$c$ has $t_0$-radius at most $r$.

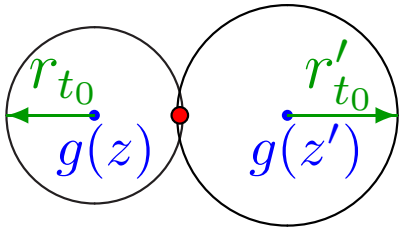
$r_{t_0} \leq r$. $\square$




Efficient aggregation: $z_i$ with the smallest $t_0$-radius

# *Sensitivity of the center of attention*

$\mathsf{LS}_g(z) = \max_{z'} \mathsf{dist}(g(z), g(z')) \leq 2r_{t_0+1}$



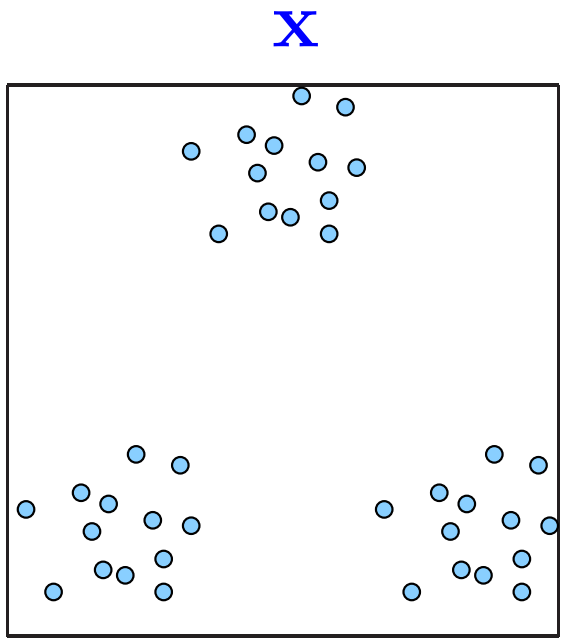

$S(z) = 2 \max\limits_{k=0,\ldots,s} r_{t_0+k+1} \cdot e^{-\varepsilon k}$

# *Application to $k$-means cluster centers*

[OstrovskyRabaniSchulmanSwamy06]:

$x$ is $\psi$-separated if

$$\frac{\text{cost of } k\text{-clustering } x}{\text{cost of } (k-1)\text{-clustering } x} \leq \psi$$




**Theorem**

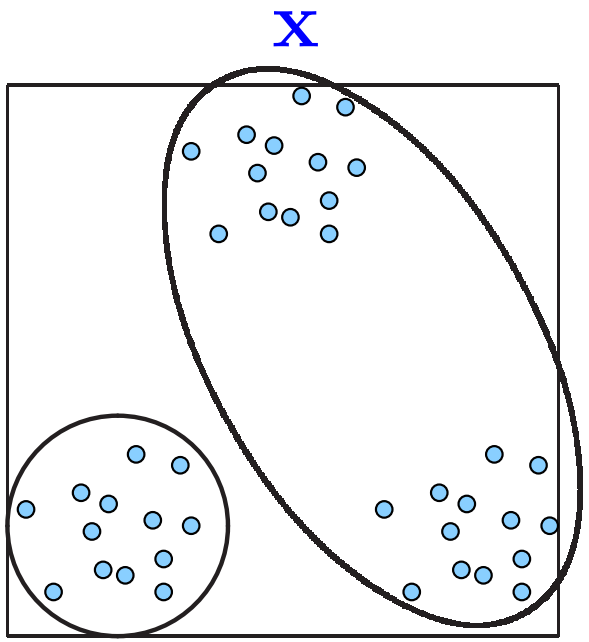
*If $x$ is $\psi$-separated then centers $(c_1, \ldots, c_k)$ can be released with noise magnitude $O(\psi \frac{\sqrt{k}}{\varepsilon})$ per dimension.*

# *Application to $k$-means cluster centers*

[OstrovskyRabaniSchulmanSwamy06]:

$x$ is $\psi$-separated if

$$\frac{\text{cost of } k\text{-clustering } x}{\text{cost of } (k-1)\text{-clustering } x} \leq \psi$$




**Theorem**

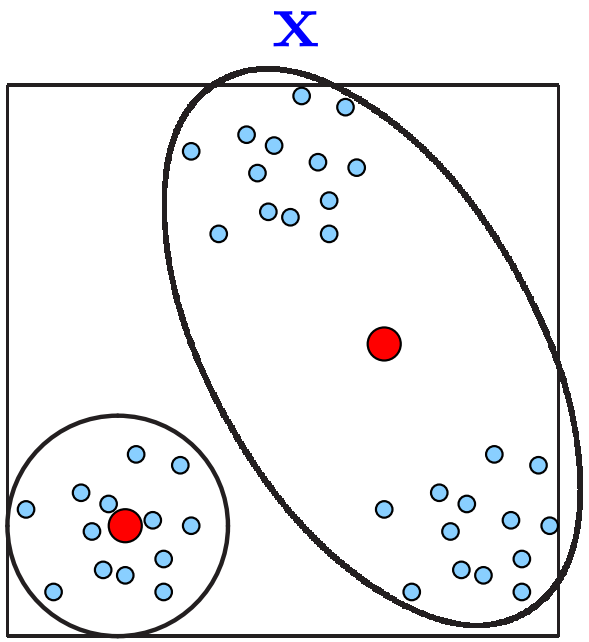
*If $x$ is $\psi$-separated then centers $(c_1, \ldots, c_k)$ can be released with noise magnitude $O(\psi \frac{\sqrt{k}}{\varepsilon})$ per dimension.*

# *Application to $k$-means cluster centers*

[OstrovskyRabaniSchulmanSwamy06]:

$x$ is $\psi$-separated if

$$\frac{\text{cost of } k\text{-clustering } x}{\text{cost of } (k-1)\text{-clustering } x} \leq \psi$$




**Theorem**

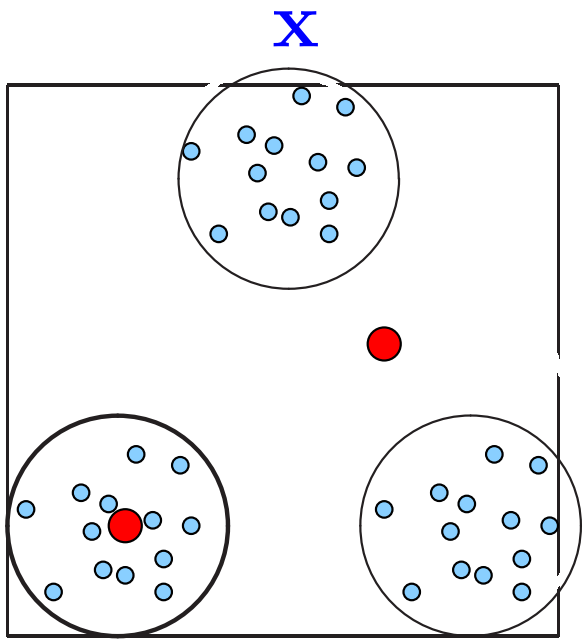
*If $x$ is $\psi$-separated then centers $(c_1, \ldots, c_k)$ can be released with noise magnitude $O(\psi \frac{\sqrt{k}}{\varepsilon})$ per dimension.*

# *Application to $k$-means cluster centers*

[OstrovskyRabaniSchulmanSwamy06]:

$x$ is $\psi$-separated if

$$\frac{\text{cost of } k\text{-clustering } x}{\text{cost of } (k-1)\text{-clustering } x} \leq \psi$$




**Theorem**

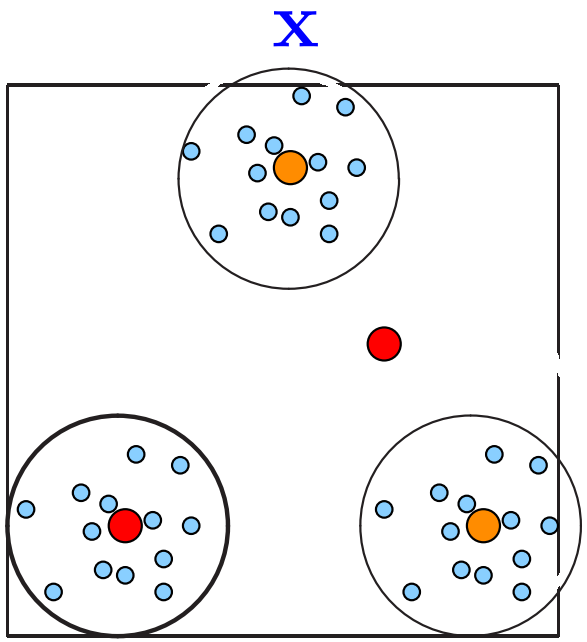
*If $x$ is $\psi$-separated then centers $(c_1, \ldots, c_k)$ can be released with noise magnitude $O(\psi \frac{\sqrt{k}}{\varepsilon})$ per dimension.*

# *Application to $k$-means cluster centers*

[OstrovskyRabaniSchulmanSwamy06]:

$x$ is $\psi$-separated if

$$\frac{\text{cost of } k\text{-clustering } x}{\text{cost of } (k-1)\text{-clustering } x} \leq \psi$$




**Theorem**

*If $x$ is $\psi$-separated then centers $(c_1, \ldots, c_k)$ can be released with noise magnitude $O(\psi \frac{\sqrt{k}}{\varepsilon})$ per dimension.*

## *Road map*

I. Introduction
- Definitions
- Global sensitivity framework [DMNS06]

II. Smooth sensitivity framework
- Motivating examples
- The framework
- Computing and approximating smooth sensitivity

III. Sample-and-Aggregate framework

## *Conclusion: fundamental question*

Which computations are not too sensitive to individual inputs?

Which functions $f$ admit $\varepsilon$-differentially private approximation $A$?

## *Conclusion: private data analysis*

We can release function $f$ of the database $x$
with noise tuned to the sensitivity of $f$ at $x$

- when smooth sensitivity can be computed/approximated
- by using sample and aggregate (automatable transformation)

## *Conclusion: private data analysis*

We can release function $f$ of the database $x$
with noise tuned to the sensitivity of $f$ at $x$

- when smooth sensitivity can be computed/approximated
- by using sample and aggregate (automatable transformation)

Recent developments:

- Releasing classes of functions [Blum Ligett Roth 08]
- Tasks not captured by function approximation
  - auction design [McSherry Talwar 07]
  - supervised learning [Kasiviswanathan Lee Nissim R Smith]
    Tomorrow at 10:30am: CIS seminar by Adam Smith
  - synthetic data [Machanavajjhala Kifer Abowd Gehrke Vilhuber]

Open: releasing non-numeric data (e.g., graphs, text)