Differential Privacy II:
Basic Tools

Adam Smith
Computer Science & Engineering Department
Penn State

IPAM Data 2010
June 8, 2009

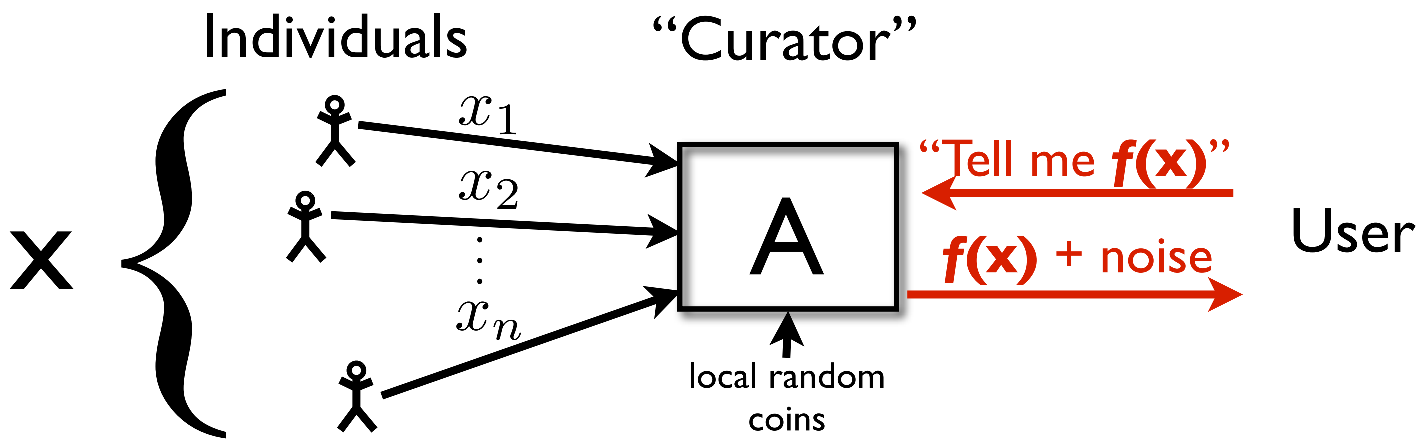
Techniques for differential privacy

- Global sensitivity and noise addition
  - basic framework
  - statistical examples
  - extensions
- Local sensitivity
  - When global sensitivity just won't cut it
- Exponential sampling
  - When noise addition makes no sense

Techniques for differential privacy

- Global sensitivity and noise addition
  - ➢ basic framework
  - ➢ statistical examples
  - ➢ extensions
- Local sensitivity
  - ➢ When global sensitivity just won't cut it
- Exponential sampling
  - ➢ When noise addition makes no sense

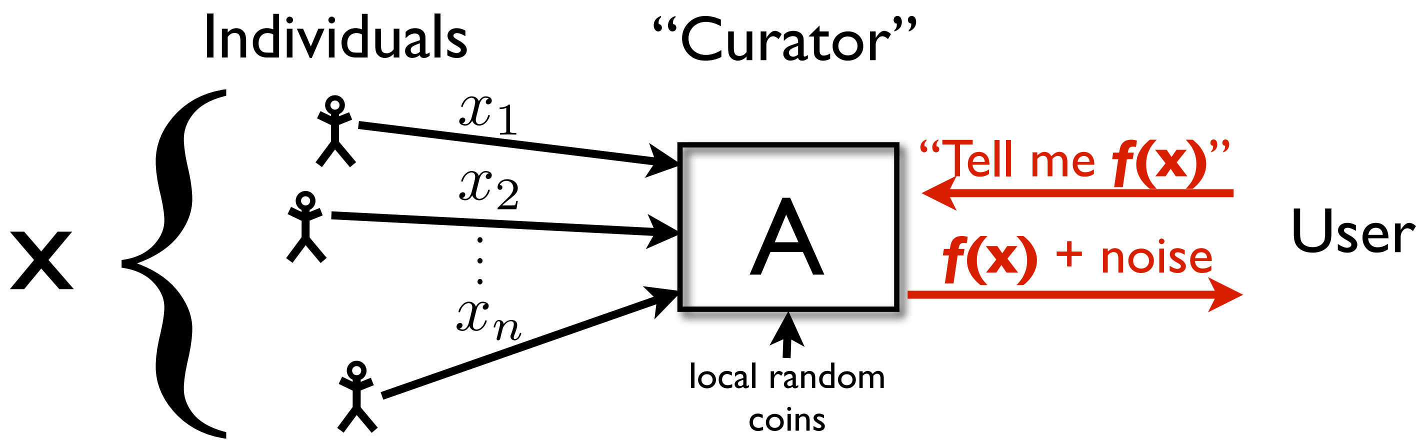
Output Perturbation



- May be repeated many times
  - **Composition Lemma:** q releases are jointly qε-differentially private
- May be noninteractive
  - Non-interactive: release pre-defined summary stats + noise

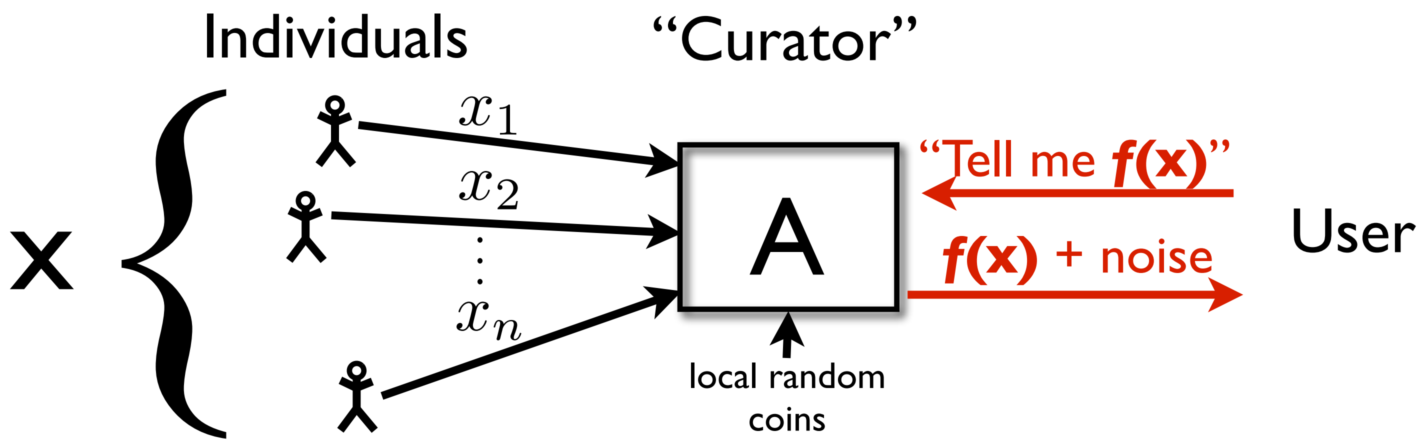
- How much noise is sufficient?

Global Sensitivity [DMNS06]



- Consider $f : \mathcal{D}^n \to \mathbb{R}^d$ (for convenience: fix **n**)
- Intuition: ***f*(x)** can be released accurately when ***f*** is insensitive to individual entries $x_1, x_2, \ldots, x_n$
- Global Sensitivity: $\text{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
- Example: If f(x) = #{diabetics in data set}, then $\text{GS}_f = 1$

Global Sensitivity [DMNS06]



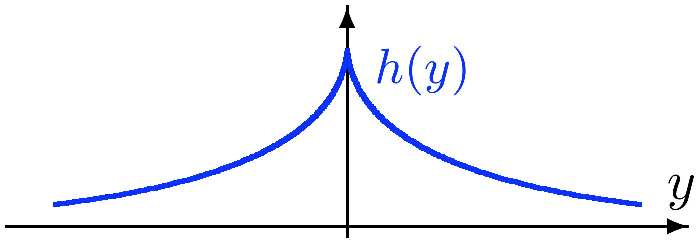
- Consider $f : \mathcal{D}^n \to \mathbb{R}^d$ (for convenience: fix **n**)
- Intuition: ***f*(x)** can be released accurately when ***f*** is insensitive to individual entries $x_1, x_2, \ldots, x_n$
- Global Sensitivity: $\mathrm{GS}_f = \max_{\text{neighbors } x,x'} \|f(x) - f(x')\|_1$
- Example: If f(x) = #{diabetics in data set}, then $\mathrm{GS}_f = 1$

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

Global Sensitivity: Noise Distribution

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

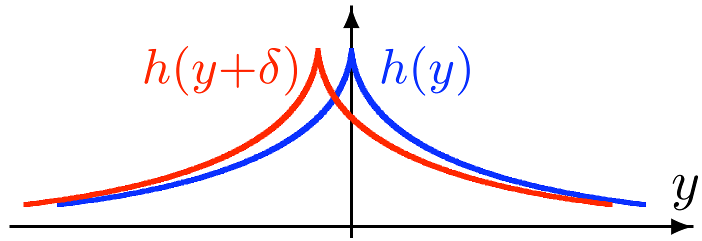
Laplace distribution $\mathsf{Lap}(\lambda)$ has density $h(y) \propto e^{-\frac{\|y\|_1}{\lambda}}$



Global Sensitivity: Noise Distribution

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

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

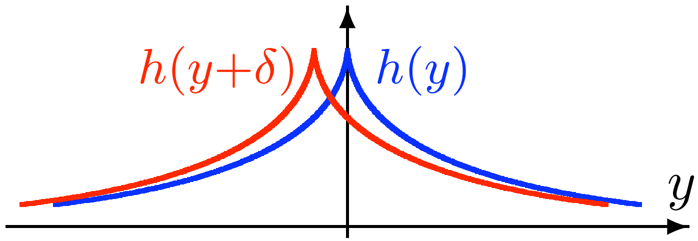


Sliding property of $\mathsf{Lap}\left(\frac{\mathsf{GS}_f}{\varepsilon}\right)$: $\frac{h(y)}{h(y+\delta)} \leq e^{\varepsilon \cdot \frac{\|\delta\|}{\mathsf{GS}_f}}$ for all y, δ

Global Sensitivity: Noise Distribution

Theorem: If $\mathsf{A(x) = f(x) + Lap}\left(\frac{\mathsf{GS_f}}{\epsilon}\right)^{\mathsf{d}}$, then A is ϵ-differentially 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, δ

Proof idea:

$$A(x): \text{ blue curve}$$

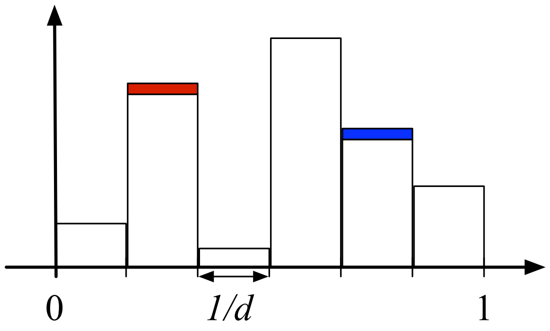
$$A(x'): \text{ red curve}$$

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

Example: Histograms

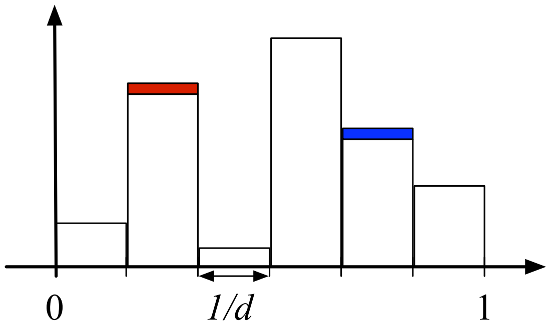
$$\mathsf{f}(\mathsf{x}) = (\mathsf{n}_1, \mathsf{n}_2, \ldots, \mathsf{n}_\mathsf{d}) \text{ where } \mathsf{n}_\mathsf{j} = \#\{\mathsf{i} : \mathsf{x}_\mathsf{i} \text{ in } j\text{-th bin}\}$$

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



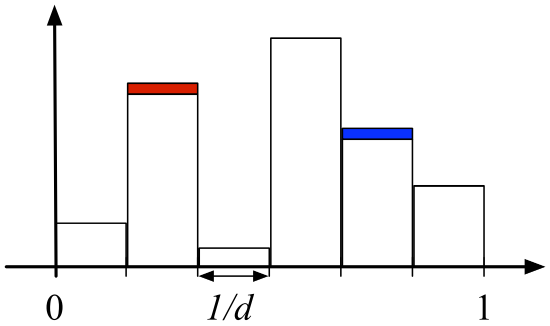
Example: Histograms

- 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
- Example
  - D = [0,1]
  - bins = intervals



Contingency 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 $O(\frac{1}{n})$
- Problem for practice:
  - Some entries may be negative. Multiple tables inconsistent.
  - [BCDKMT] 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)

Covariance Matrix

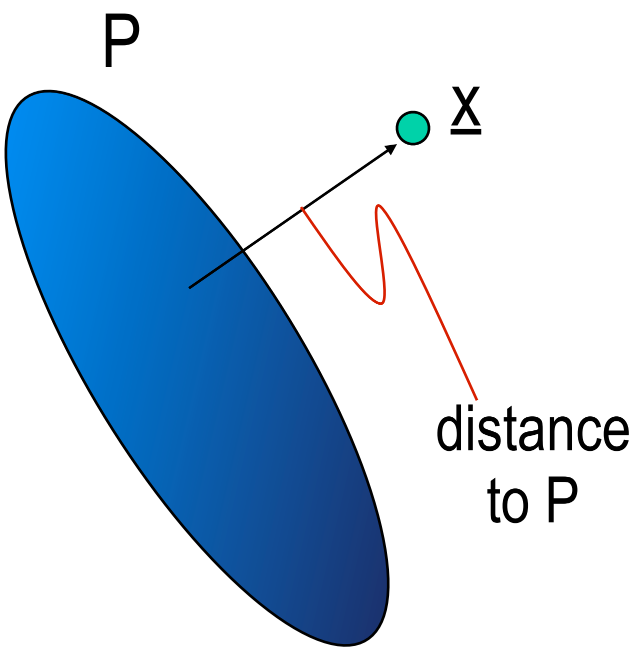
- Suppose each person's data is a real vector
- Database is a matrix X
- The covariance matrix of X is (roughly) the matrix $f(X) = X^\top X$
  - Entries measure correlation between attributes
  - First step of many analyses, e.g. PCA

$$X = \begin{pmatrix} - & x_1 & - \\ - & x_2 & - \\ & \vdots & \\ - & x_n & - \end{pmatrix}$$

- **Lemma**: If $\mathcal{D} = \{x \in \mathbb{R}^d : \|x\|_1 \leq 1\}$ then $GS_f \leq 1$
  - **Proof**: Write $f(X) = X^\top X = \sum_{i=1}^n x_i^\top x_i$
    Observe that $\|x_i^\top x_i\|_1 \leq \|x_i\|_1^2$
- Constant noise per entry suffices for differential privacy

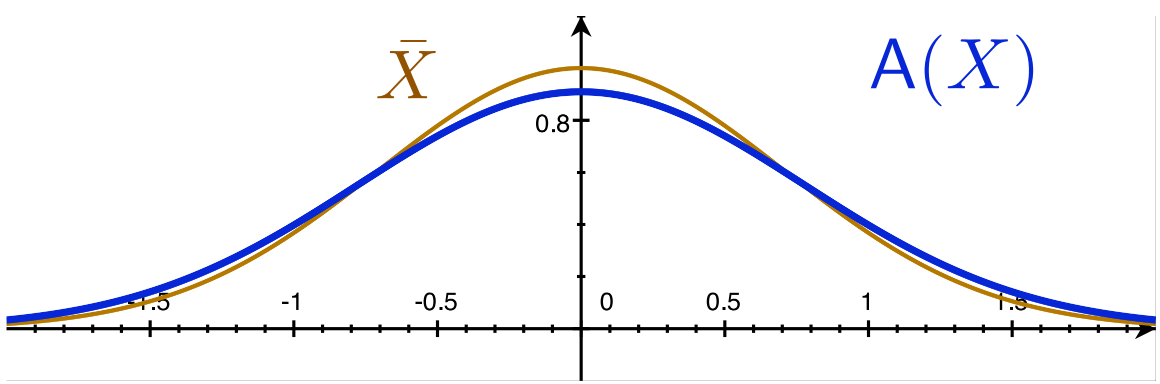
Example: Distance to a Property

- Say P = set of "good" databases
  - e.g. well-clustered databases
- Distance to P =
  # points in x that must be changed to make x in P
  - Always has GS = 1
- Example:
  - Distance to data set with "good clustering"



When Does Noise **Not** Matter?

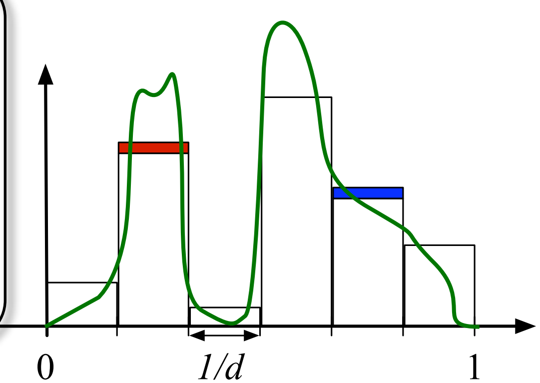
- Average: A(x) $= \bar{x} +$ Lap$(\frac{1}{\epsilon n})$
  - Suppose $X_1, X_2, X_3, \ldots, X_n$ are i.i.d. random variables
  - $\bar{X}$ is a random variable, and $\sqrt{n} \cdot (\bar{X} - \mu) \xrightarrow{\mathcal{D}} \mathrm{Normal}$
  - $\boxed{\dfrac{\mathrm{A}(X) - \bar{X}}{\mathrm{StdDev}(\bar{X})}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



Example: Histograms

- Say $x_1, x_2, \ldots, x_n$ in domain ~~D~~ [0,1]
  - ➢ Partition [0,1] into d disjoint ~~bins~~ intervals
  - ➢ $f(x) = (n_1, n_2, \ldots, n_d)$ 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

- For any smooth density h, if X_i i.i.d. ~ h, noisy histogram converges to h
  - ➢ Expected L_2 error $O(\frac{1}{\sqrt[3]{n}})$ if $n \gg \frac{1}{\epsilon^3}$
  - ➢ Same as "best" non-private histogram
  - ➢ Noted independently by [Wasserman-Zhou '09, S. '09]



Variants in other metrics

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

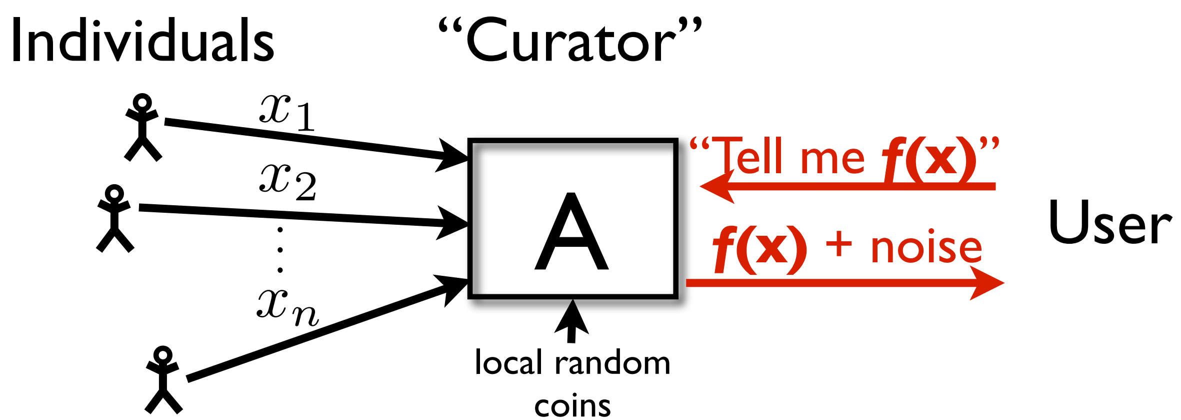
Theorem: If $A(x) = f(x) + \cancel{\text{Lap}\left(\frac{GS_f}{\epsilon}\right)^d}$, then A is $\cancel{\epsilon}$-differentially private.

$$N\left(0, \left(\frac{GS_f \cdot 3 \cdot \sqrt{\ln(1/\delta)}}{\epsilon}\right)^2\right) \qquad (\epsilon, \delta)$$

- 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 $\frac{d}{\epsilon}$

Using global sensitivity

- Many natural functions have low GS, e.g.:
  - Sample mean, histogram, covariance matrix, distance to a function, estimators with bounded "sensitivity curve", ...
- More generally, view as "programming interface"



  - May be repeated many times
    - **Composition Lemma:** q releases are jointly qε-differentially private
  - May be noninteractive
    - Non-interactive: release pre-defined summary stats + noise

Using global sensitivity

- Many natural functions have low GS, e.g.:
  - Sample mean, histogram, covariance matrix, distance to a function, estimators with bounded "sensitivity curve", ...
- More generally, view as "programming interface"
  - Many algorithms can be expressed as a sequence of low-sensitivity queries
    - [BDMN] perceptron, k-means, "SQ" learning algorithms
    - [FFKN] coreset computation for clustering
    - [MW] gradient ascent algorithm for logistic regression
  - Post-processing can improve accuracy
    - [BCDKMT] Multiple contingency tables
    - [HRMS] Sorted histograms
  - Applications made easier by SQL-like language [McSherry]

"Programming": k-means algorithm

- Given n points in $\mathbb{R}^d$, want natural "grouping"
- Start with k candidate "cluster centers" $m_1,\ldots,m_k$
- For T rounds:
  - S_j = {x_i: closest center is m_j} (Voronoi partition)
  - m_j = average of points in S_j (new candidate centers)

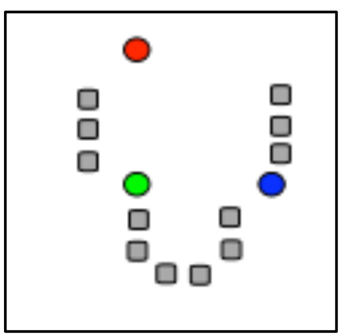


Image credit: Wikipedia

"Programming": k-means algorithm

- Given n points in $\mathbb{R}^d$, want natural "grouping"
- Start with k candidate "cluster centers" $m_1,\ldots,m_k$
- For T rounds:
  - S_j = {x_i: closest center is m_j} (Voronoi partition)
  - m_j = average of points in S_j (new candidate centers)

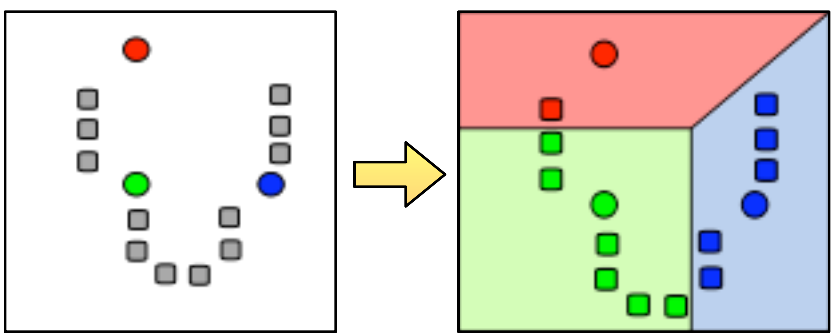


Image credit: Wikipedia

“Programming”: k-means algorithm

- Given n points in $\mathbb{R}^d$, want natural “grouping”
- Start with k candidate “cluster centers” $m_1,\ldots,m_k$
- For T rounds:
  - $S_j = \{x_i$: closest center is $m_j\}$ (Voronoi partition)
  - m_j = average of points in S_j (new candidate centers)

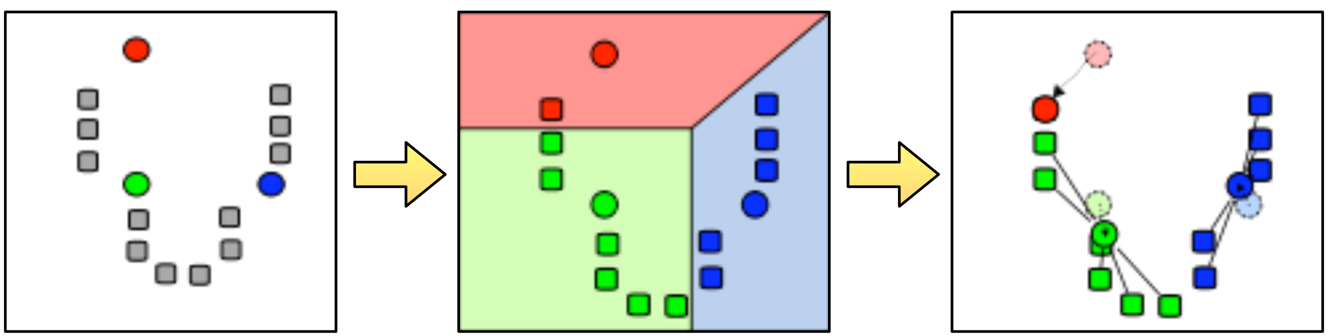


Image credit: Wikipedia

"Programming": k-means algorithm

- Given n points in $\mathbb{R}^d$, want natural "grouping"
- Start with k candidate "cluster centers" $m_1,\ldots,m_k$
- For T rounds:
  - ➢ S_j = {x_i: closest center is m_j} (Voronoi partition)
  - ➢ m_j = average of points in S_j (new candidate centers)

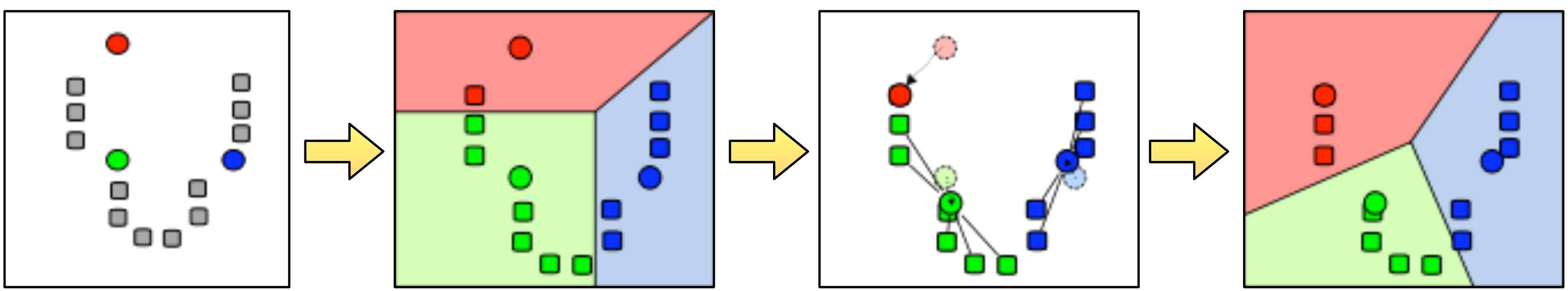
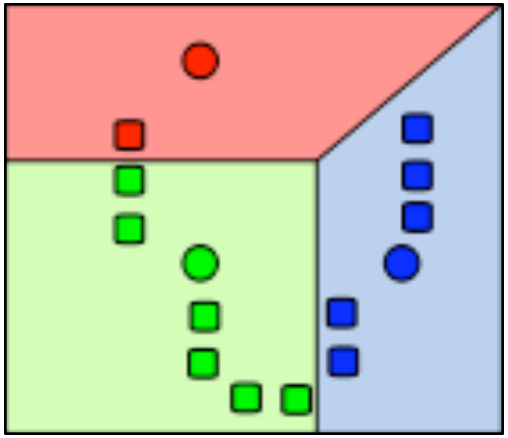


Image credit: Wikipedia

k-means via low-sensitivity queries [BDMN]

- Suppose $\mathcal{D} = \{x \in \mathbb{R}^d : \|x\|_1 \le 1\}$
- For T rounds:
  - ➢ $S_j = \{x_i$: closest center is $m_j\}$
  - ➢ m_j = average of points in S_j



- Differentially private version: In each round,
  - ➢ Ask two queries:
    - $(c_1,\ldots,c_k)$ = (noisy) counts for Voronoi partition (GS = 1)
    - $(M_1,\ldots,M_k)$ = (noisy) sums of points in each Voronoi cell (GS = 1)
  - ➢ Set $m_j = M_j / c_j$
- Set $\epsilon' = \frac{2T}{\epsilon}$, answer queries with noise Lap$(\frac{1}{\epsilon'})$ per entry

Better accuracy via consistency

- Can sometimes "post-process" perturbed answers to reduce noise
  - Use extra structure in desired output
- Example: [HRMS]
  - Data: x_i = website visited today by Penn State student i
  - Goal: release popularity distribution of websites
  - No site names
  - Answer = Sorted histogram
  - Idea: after adding noise, output "closest" sorted sequence

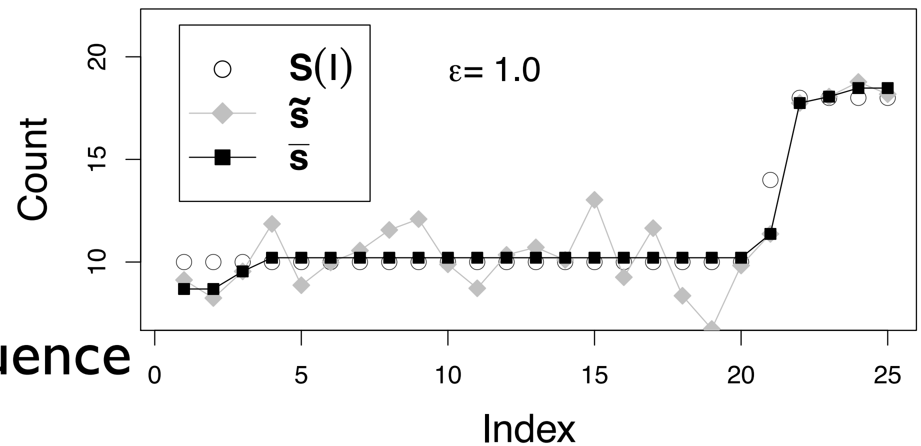
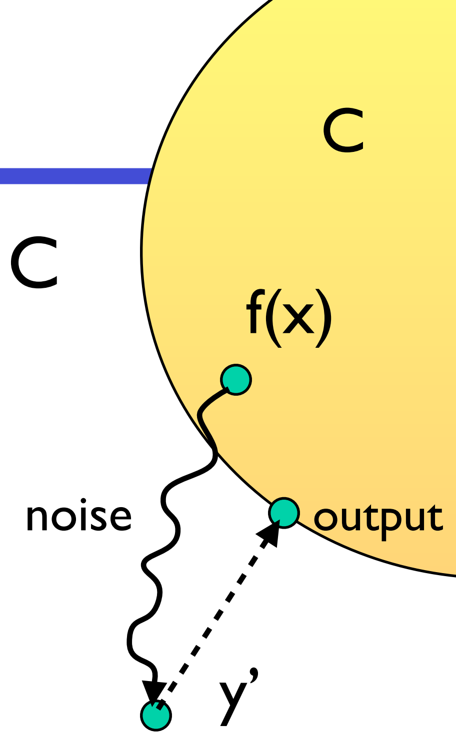


Image credit: Hays et al, arXiv:0904.0942

Better accuracy via consistency



- Suppose that original answer must lie in set C
  - e.g. $C = \{y \text{ in } R^d : y_1 \leq y_2 \leq \ldots \leq y_d\}$
- Idea:
  - Compute y' = f(x) + noise
  - Release closest point in C to y'
- **Proposition:** If C is convex, L_2 error never increases
- Sometimes improves significantly, e.g.
  - [HMRS]: If sorted histogram changes slowly, error drops to from $\frac{d}{\epsilon}$ to $\frac{\mathrm{polylog}(d)}{\epsilon}$
  - [BCDKMT]: If releasing all k-way contingency tables, can project onto consistent tables and save factor of 2^k in noise

Global Sensitivity Summary

- Simple framework for output perturbation with strong privacy guarantees
  - Noise levels small enough to allow meaningful analysis
  - General interface
- Improved in several respects
  - Local vs global sensitivity [NRS]: Add less noise on "good" instances
  - Releasing many functions simultaneously [BLR,DNNRV,RR]
  - Beyond function approximation: many tasks not so simple
    - Auction design [MT], learning [KLNRS,CM,...], inference [MKAGV,WZ],...

Local and Smooth Sensitivity

High Global Sensitivity: Median

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

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

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

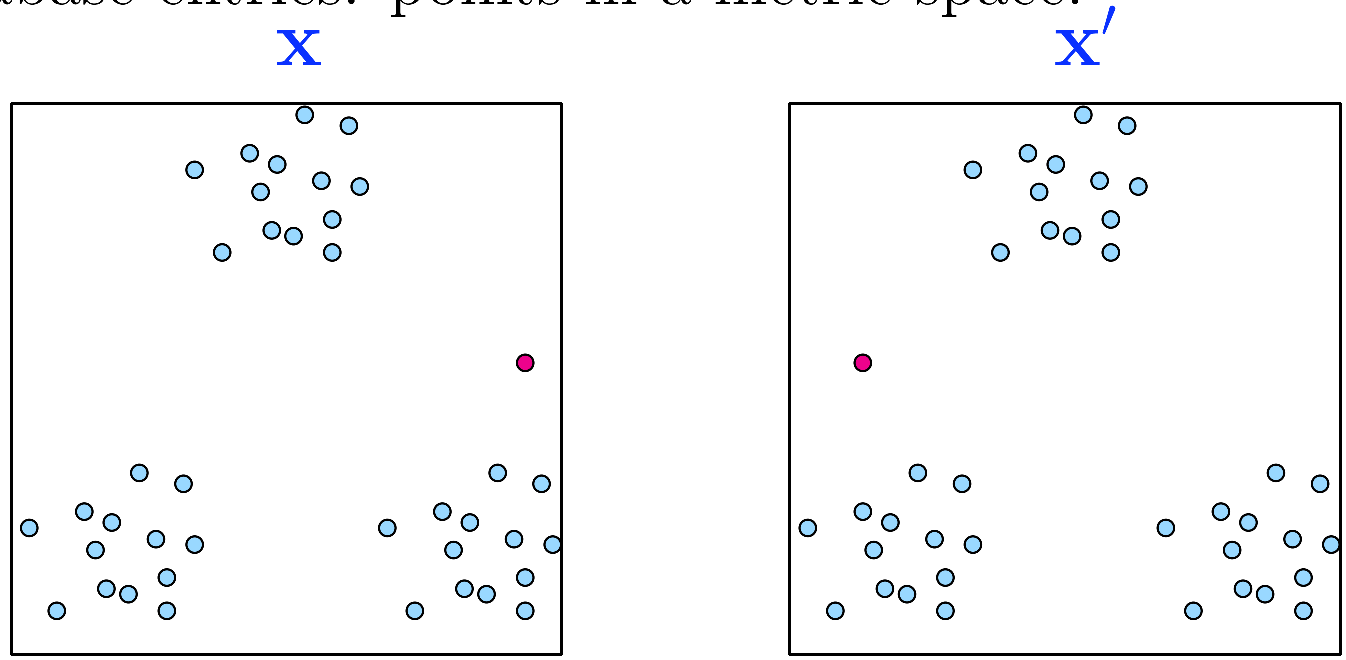
$$\mathsf{GS}_{\text{median}} = 1$$

- 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: Cluster centers

Database entries: points in a metric space.

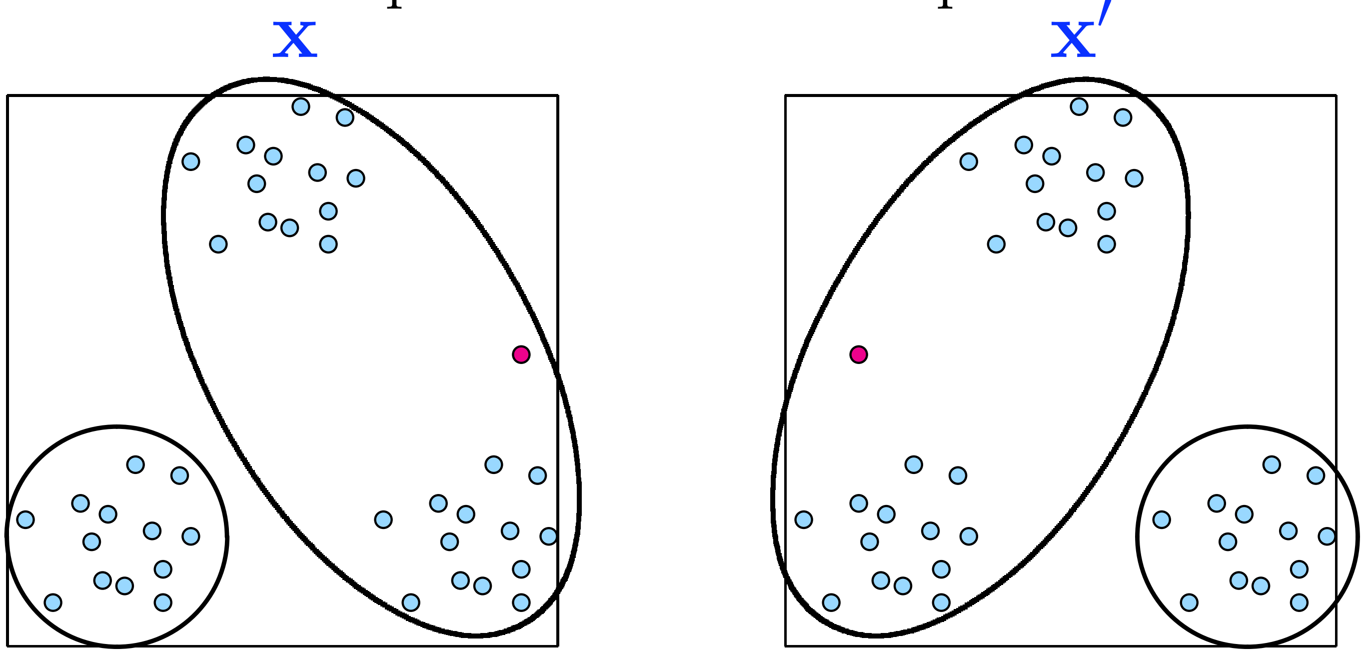


Global sensitivity of cluster centers is roughly the diameter of the space.

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

High Global Sensitivity: Cluster centers

Database entries: points in a metric space.

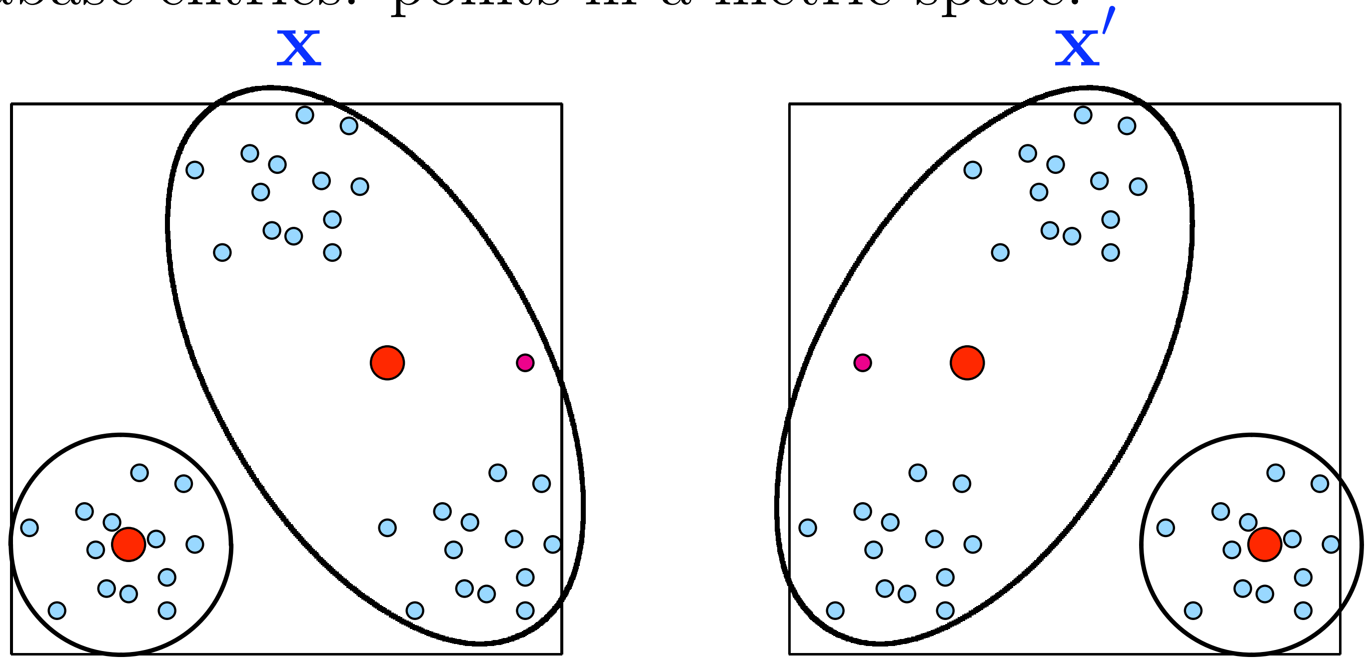


Global sensitivity of cluster centers is roughly the diameter of the space.

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

High Global Sensitivity: Cluster centers

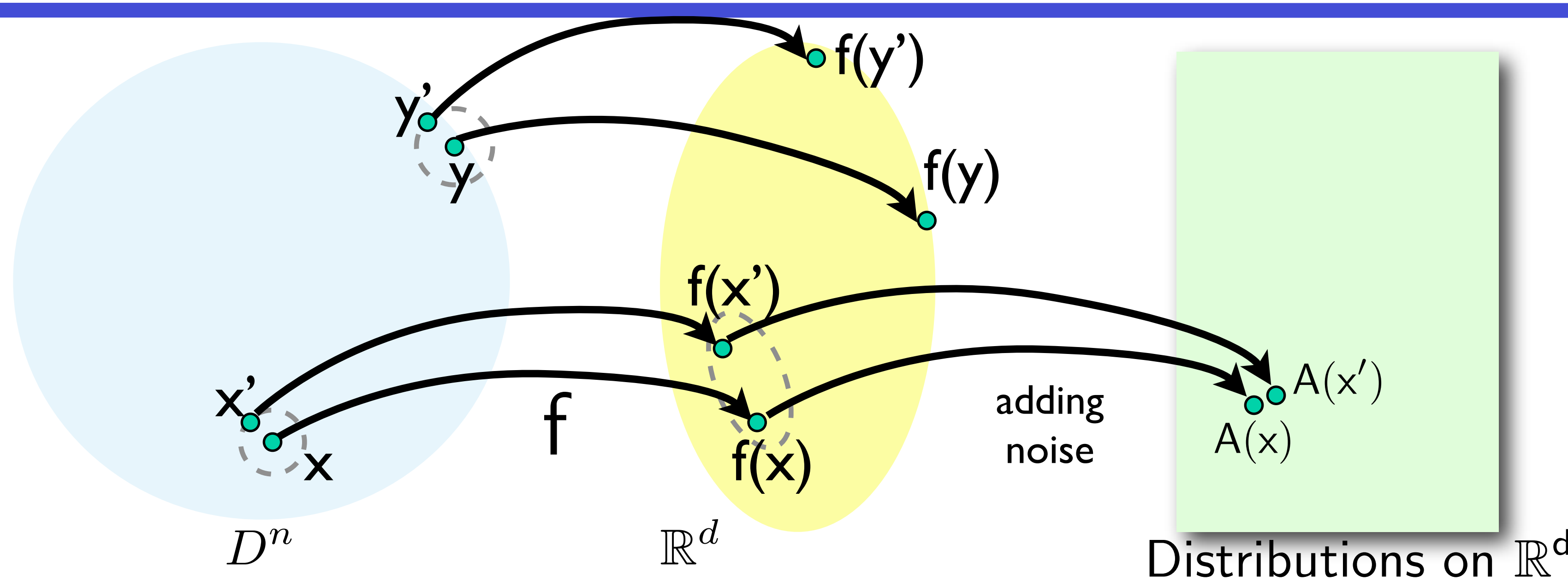
Database entries: points in a metric space.



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

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

Global versus local [NRS07]

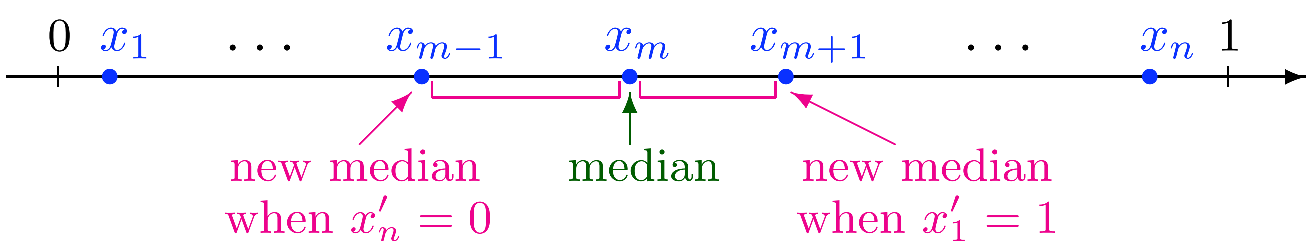


- Global sensitivity is worst case over inputs
- Local sensitivity: $$\mathsf{LS}_f(x) = \max_{x' \text{ neighbor of } x} \|f(x) - f(x')\|_1$$
- Reminder: $\mathsf{GS}_f(x) = \max_x \mathsf{LS}_f(x)$
- **Goal:** add less noise when local sensitivity is lower

Local Sensitivity

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

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



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

Instance-based noise: first attempt

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

Problem: Noise magnitude might reveal information.

Example: median

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

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

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

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

A is not ε-indistinguishable

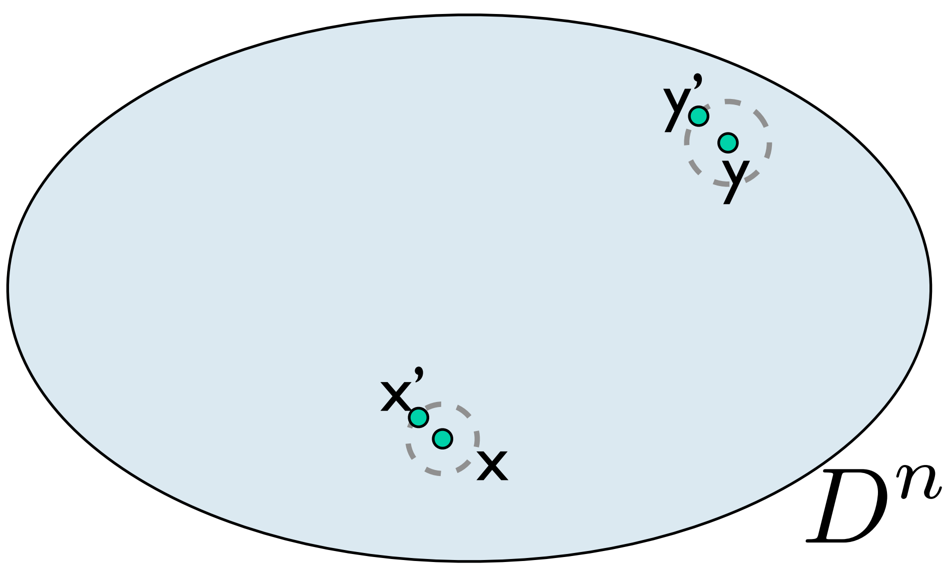
Lesson: Noise magnitude must be an insensitive function.

Instance-based noise

- Problem: can't be close to high-sensitivity instance
- Two approaches:
  - ➢ [NRS'07] Compute a "smoothed" version of local sensitivity
  - ➢ [DL'09+] Use global sensitivity to get a diffe.p. upper bound on local sensitivity.

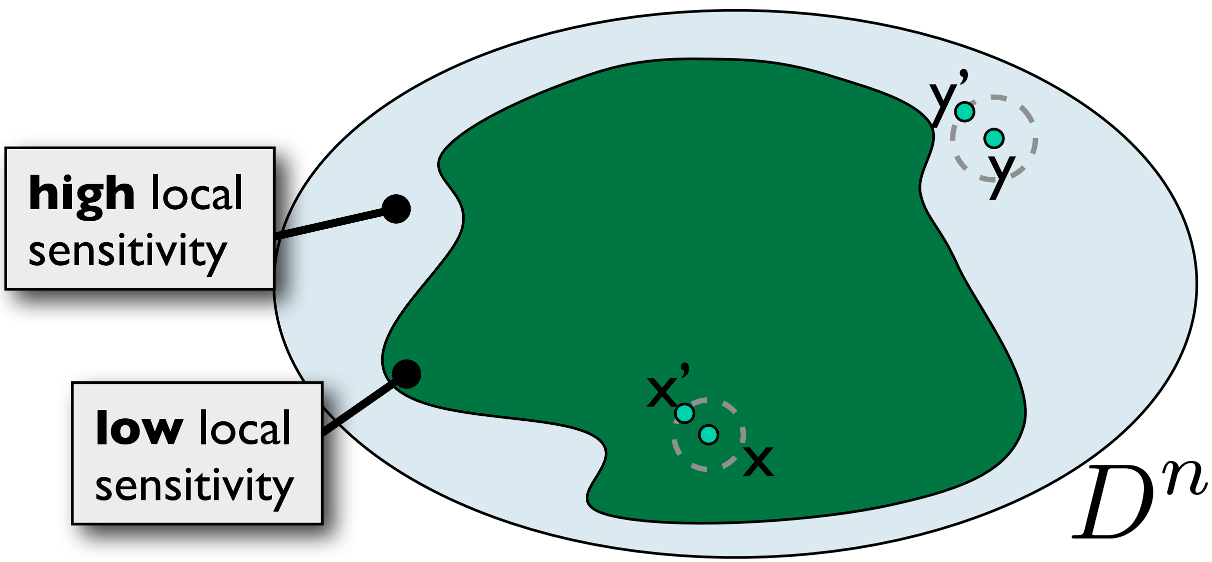
Instance-based noise

- Problem: can’t be close to high-sensitivity instance
- Two approaches:
  - [NRS’07] Compute a “smoothed” version of local sensitivity
  - [DL’09+] Use global sensitivity to get a diffe.p. upper bound on local sensitivity.



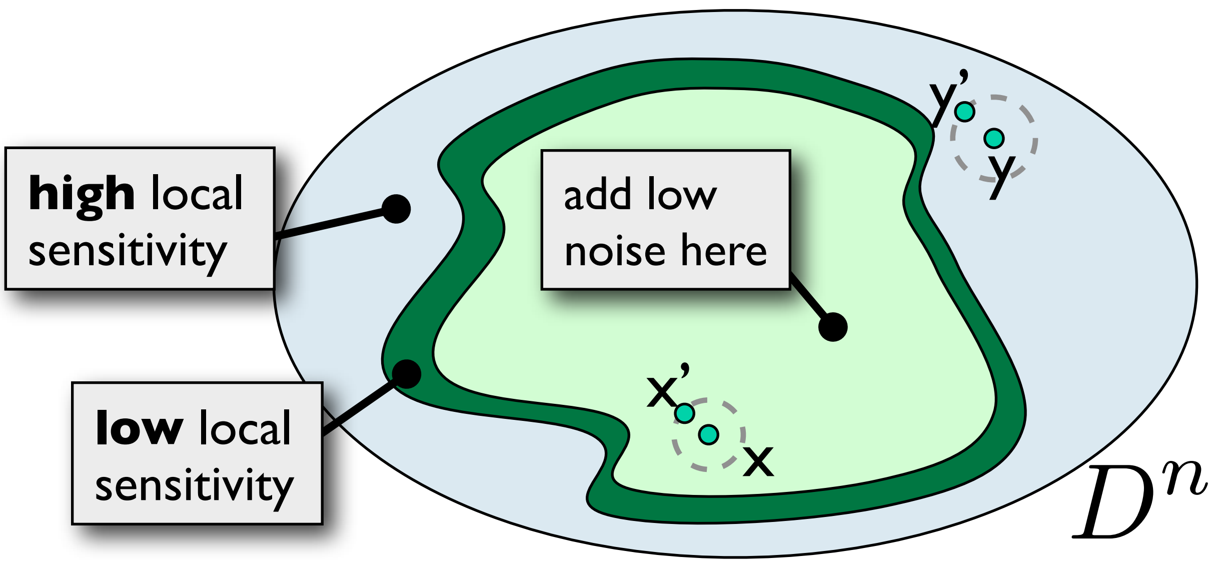
Instance-based noise

- Problem: can't be close to high-sensitivity instance
- Two approaches:
  - [NRS'07] Compute a "smoothed" version of local sensitivity
  - [DL'09+] Use global sensitivity to get a diffe.p. upper bound on local sensitivity.



Instance-based noise

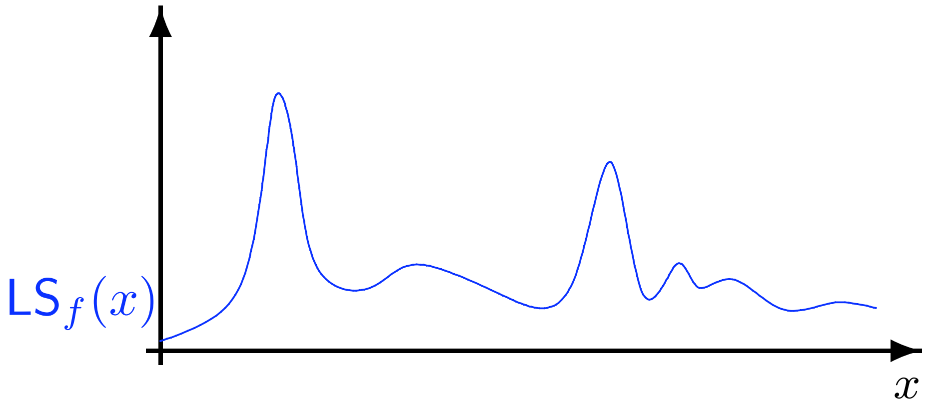
- Problem: can't be close to high-sensitivity instance
- Two approaches:
  - [NRS'07] Compute a "smoothed" version of local sensitivity
  - [DL'09+] Use global sensitivity to get a diffe.p. upper bound on local sensitivity.



Smooth Bounds on Sensitivity

Design sensitivity function $S(x)$

- $S(x)$ is an ε-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

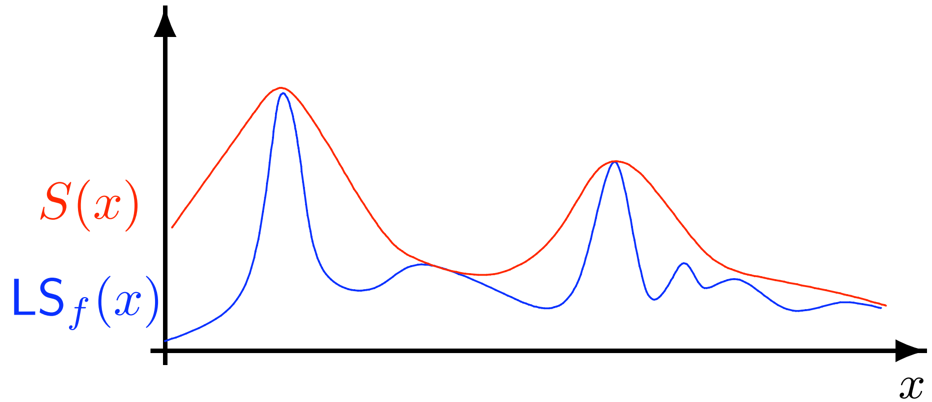
If $A(x) = f(x) + \mathsf{noise}\left(\frac{S(x)}{\varepsilon}\right)$ *then* A *is* ε'*-indistinguishable.*

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

Smooth Bounds on Sensitivity

Design sensitivity function $S(x)$

- $S(x)$ is an ε-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 $A(x) = f(x) + \mathsf{noise}\left(\frac{S(x)}{\varepsilon}\right)$ *then* A *is* ε'*-indistinguishable.*

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

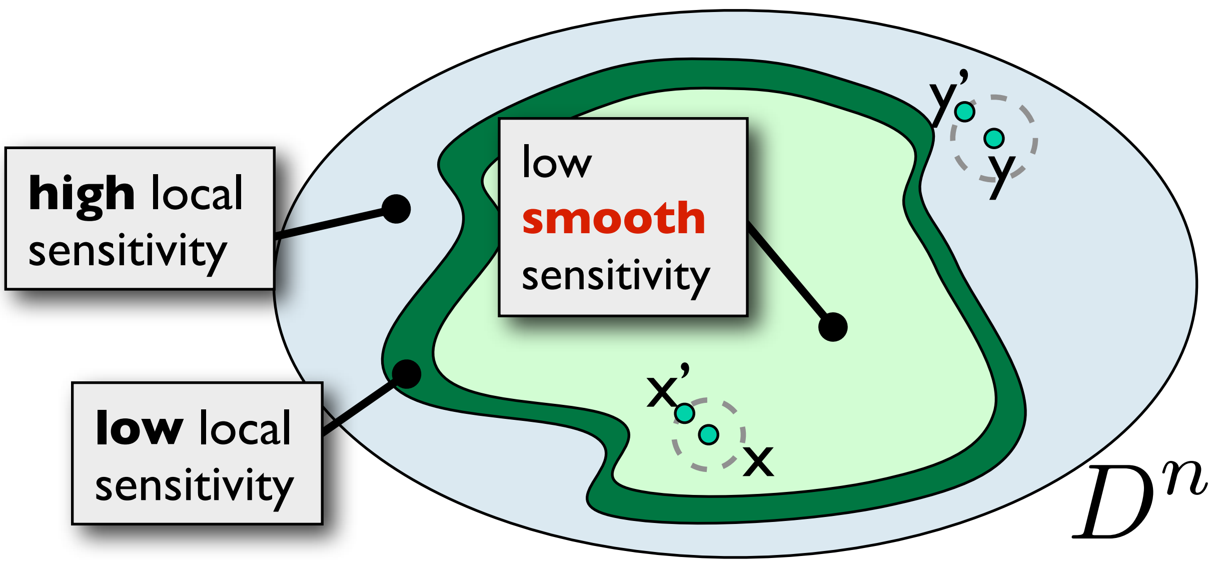
Smooth Bounds on Sensitivity

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

Lemma

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

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



Computing Smooth Sensitivity

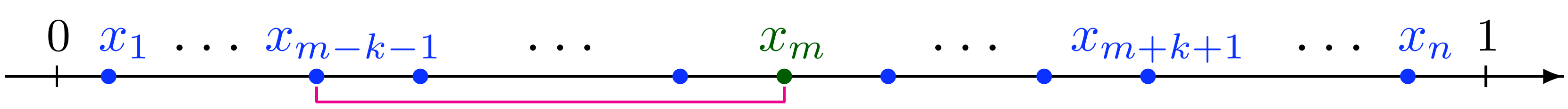
Observation

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

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

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

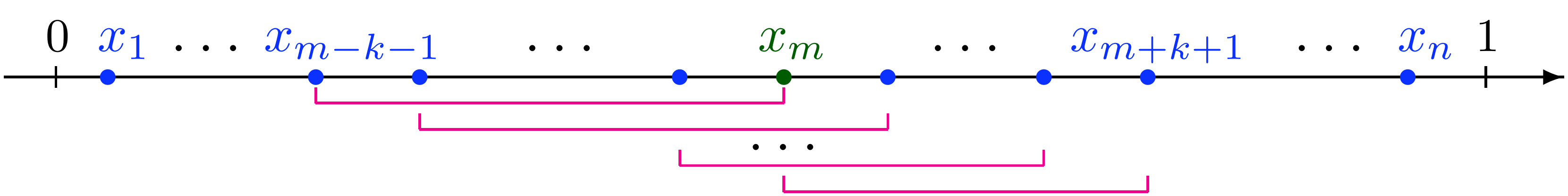
Observation

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

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

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

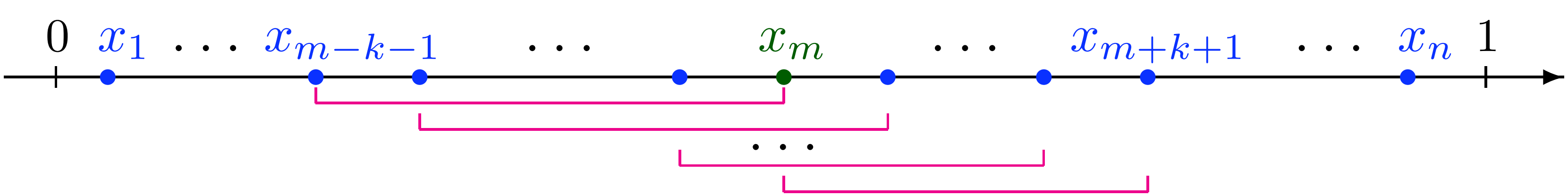
Observation

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

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

Example: median

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



[Orshanskiy] S^* is computable in $O(n \log n)$ time.

Computing Smooth Sensitivity

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

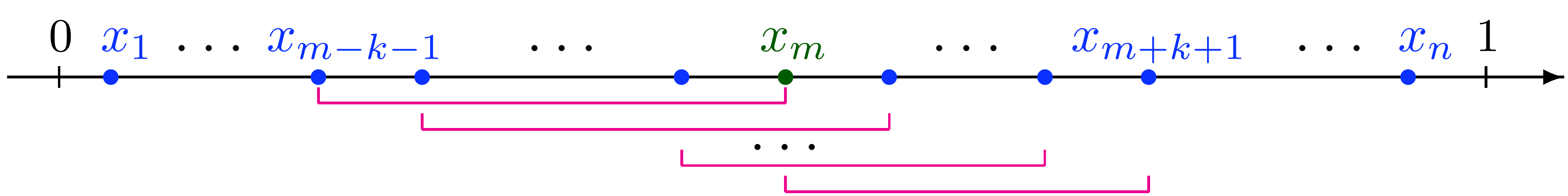
Observation

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

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

Example: median

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



[Orshanskiy] S^* is computable in $O(n \log n)$ time.

Algorithmic Questions

- Applying this framework requires computing smooth bounds on sensitivity
  - When can compute smooth bounds efficiently?
  - How can we avoid this for "complicated" functions?

Results [NRS '07,DL'09]

- [NRS] Computation of smoothed sensitivity for several useful functions
  - Order statistics (e.g. median, quartiles, max, min)
  - Trimmed mean
  - # of triangles in a graph
  - Min. spanning tree cost
- [DL'09] Connection to "robust" statistics
  - Algorithms for bounding local sensitivity of order statistics, linear regression
- Generic framework for smoothing functions so they have low sensitivity
  - Based on sampling; see my Thursday talk

Exponential Sampling

Exponential Sampling [McSherry-Talwar]

- Sometimes noise addition makes no sense
  - mode of a distribution
  - minimum cut in a graph
  - decision tree classifier

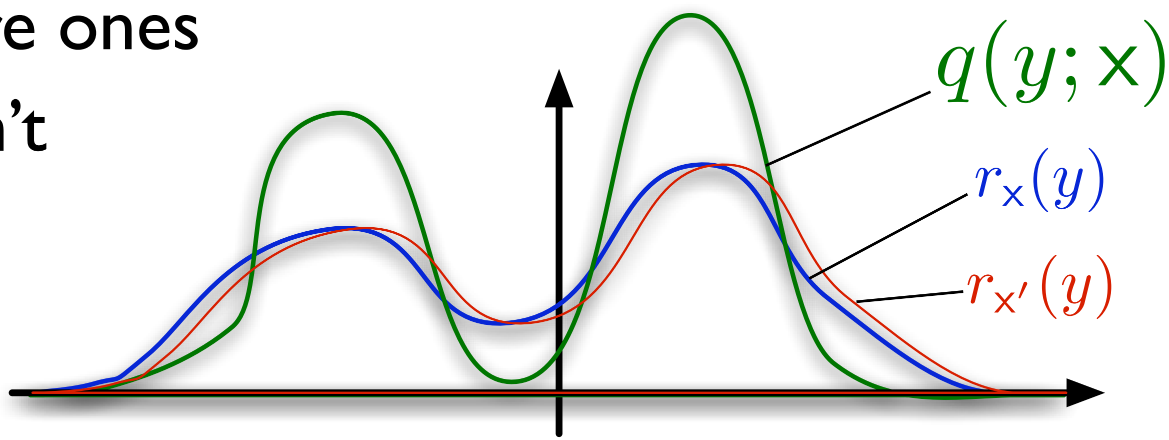
- [MT] Motivation: auction design
  - Differential privacy implies approximate truthfulness
- Subsequently applied broadly

Example: Mode

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

Procedure: Given x,

- Output website y_0 with probability $r_{\mathsf{x}}(y) \propto \exp(\epsilon q(y;\mathsf{x}))$
- Popular sites exponentially more likely than rare ones
- Website scores don't change too quickly



Example: Mode

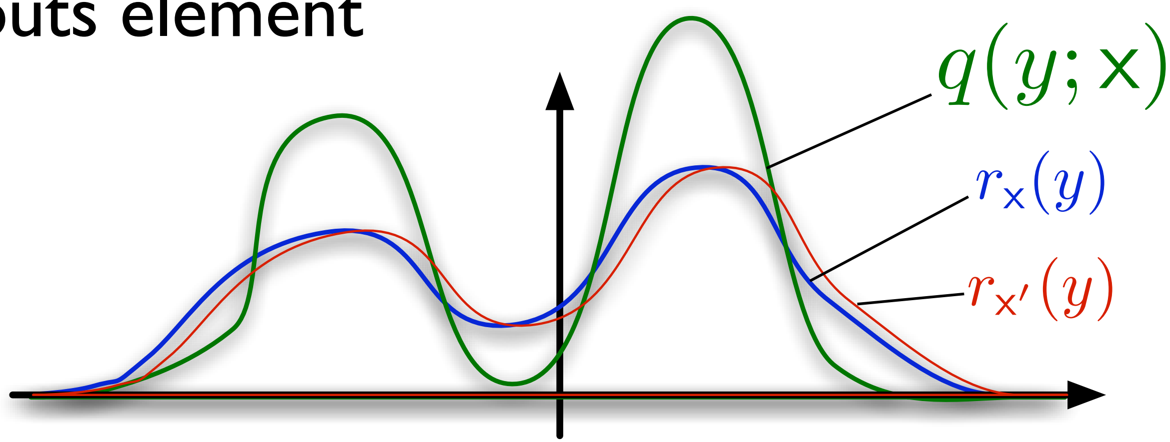
Procedure: Given x,

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

- **Claim:** The mechanism is 2ε-differentially private

$$\frac{r_{\mathsf{x}}(y)}{r_{\mathsf{x}'}(y)} = \frac{e^{\epsilon q(y;\mathsf{x})}}{e^{\epsilon q(y;\mathsf{x}')}} \cdot \frac{\sum_{z \in Y} e^{\epsilon q(z;\mathsf{x}')}}{\sum_{z \in Y} e^{\epsilon q(z;\mathsf{x})}} \leq e^{2\epsilon}$$

- In expectation, outputs element with # occurrences ≥ max - (ln |Y|) / ε



Exponential Sampling

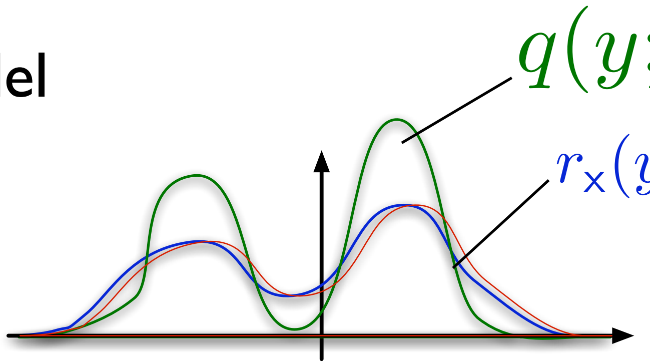
Ingredients:

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

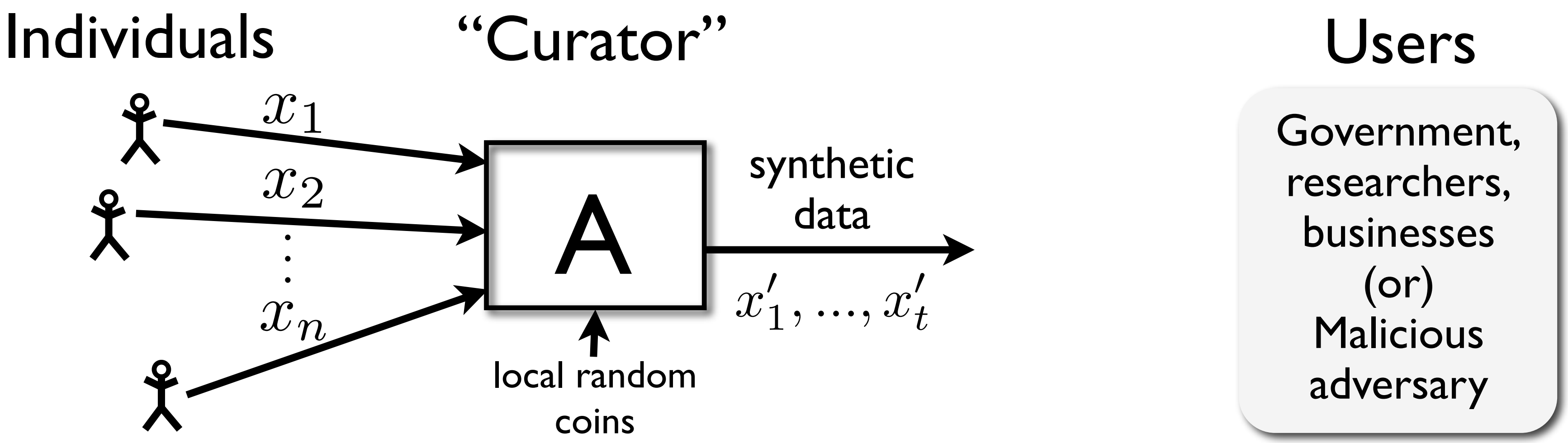
Procedure: Given x,

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

- **Example** [MKAGV]:
  - Y= parameter space for parametric model
  - q = log-likelihood based on x
  - Output draw from "squashed" posterior $r_{\mathsf{x}}(y) \propto p(y;\mathsf{x})^{\epsilon}$
  - Differentially private if log-likelihood is bounded

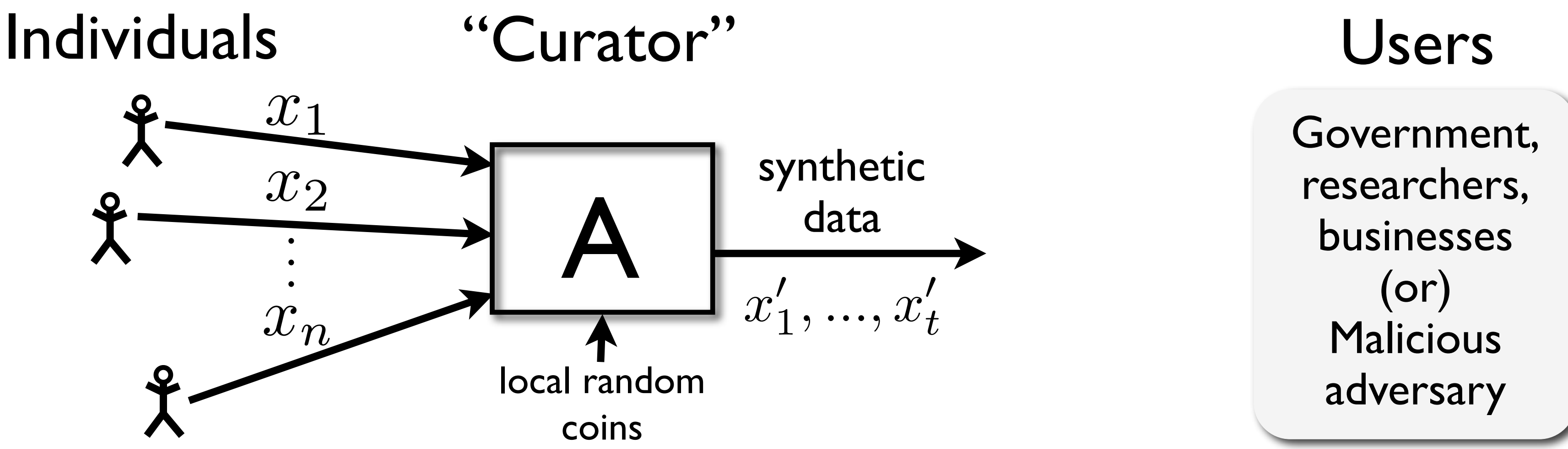


Application: Synthetic Data



- Goal: new data set with "similar" statistical properties
  - ➢ Specify precisely the set of preserved properties
  - ➢ [Blum, Ligett, Roth 2008] broad theoretical possibility results
  - ➢ Improved parameters, hardness [DNRRV], cont. data [WZ]
  - ➢ [Machanavajjhala, Kifer, Abowd, Gehrke, Vilhuber 2008, McSherry-Talwar 2008]
    - Differentially private geographic data, in use at US Census bureau

Application: Synthetic Data



- Goal: new data set with "similar" statistical properties
  - Specify precisely the set of preserved properties
  - [Blum, Ligett, Roth 2008] broad theoretical possibility results
  - Improved parameters, hardness [DNRRV], cont. data [WZ]
  - [Machanavajjhala, Kifer, Abowd, Gehrke, Vilhuber 2008, McSherry-Talwar 2008]
    - Differentially private geographic data, in use at US Census bureau

Synthetic Data [BLR]

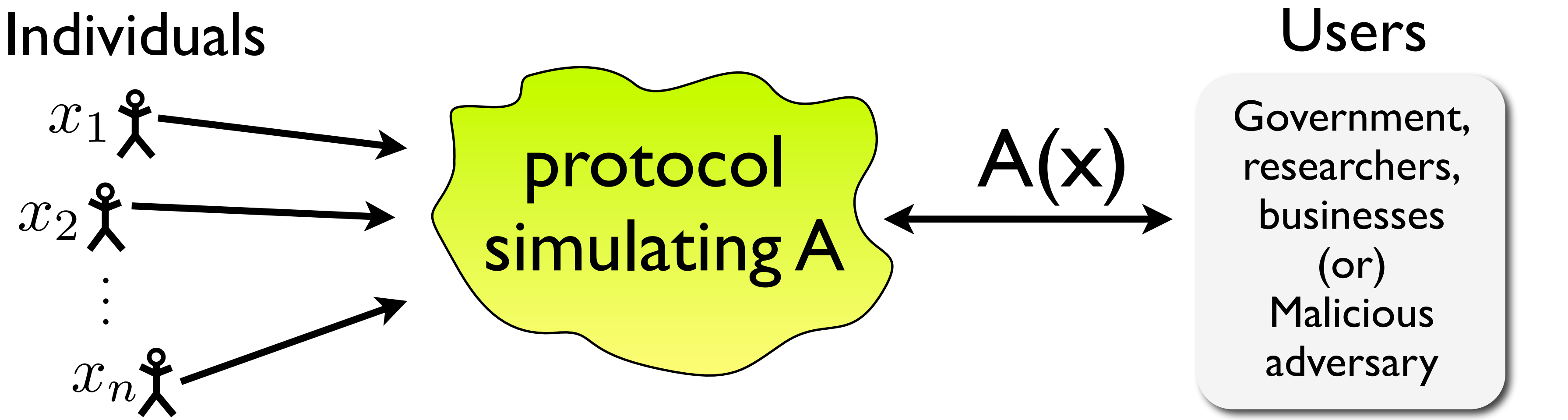
- Given:
  - ➢ collection of predicates $C=\{P_1,\ldots,P_K\}$
  - ➢ x = large data set
- Quality of a data set y:
  - ➢ $q(y;x) = -\max_{\{P\in C\}} |$ frequency of P in y - frequency of P in x $|$
- Y = {small data sets}
- Idea:
  - ➢ y is good for x if $q(y; x) \geq -10\%$, and bad if $q(y;x) \leq -20\%$.
  - ➢ A good small data set exists since a sample from x is good
  - ➢ Exponential mechanism assigns very low weight to bad y

Changing the Model: Reducing Trust

Changing the Model

- So far: trusted curator
  - ➢ single point of failure
- Approaches to reducing dependency
  - ➢ Randomized response [Warner, EGS, KLNRS]
    - Each individual keeps his data & randomizes answers to curator
  - ➢ Cryptographic "secure function evaluation" [DKMMN]
    - Individuals jointly, securely simulate a virtual curator
  - ➢ "Short memory" curators [DNPRY]
    - Curators keeps data only for limited time
    - Privacy is maintained even if curator's memory is leaked

Distributed Private Data Mining



- Eliminate the trusted "Curator" [DKMMN]
- Use cryptographic protocols to jointly mine shared data
  - Individuals retain data
  - Mining algorithm still needs to respect (differential) privacy; the crypto protocols address orthogonal concerns [BNO]

This talk: Techniques & Terminology

- Basic tools:
  - Noise addition via global sensitivity
  - local/smooth sensitivity, sample-aggregate
  - exponential sampling
- Things I didn't cover:
  - lower bounds [DMNS,GR,HT,KRS,...]
  - combinatorial optimization [GLMRT]
  - convex optimization [CM,...]
  - auction design [MT]
  - "directional" global sensitivity [HT]
  - relaxations of differential privacy [MGAKV,MPRV]
  - and more!

A quote

The work described herein has, for the first time, placed private data analysis on a strong mathematical foundation. The literature connects differential privacy to decision theory, economics, robust statistics, geometry, additive combinatorics, cryptography, complexity theory, learning theory, and machine learning. Differential privacy thrives because it is natural, it is not domain-specific, and it enjoys fruitful interplay with other fields. This flexibility gives hope for a principled approach to privacy in cases, like private data analysis, where traditional notions of cryptographic security are inappropriate or impracticable.

C. Dwork, *Comm. ACM*, to appear.