Privacy in Big Data Computing: Differential Privacy Techniques

Lecture

Instructor: Xiang-Yang Li
Professor, CS Department

Data Analysis

Huge social benefits from analyzing large collections of data:

Finding correlations

E.g. medical: genotype/phenotype correlations

Providing

Improve web search results, fit ads to queries

Publishing Official Statistics

Census, contingency tables

Datamining

Clustering, learning association rules, decision trees, separators, principal component analysis

WHAT ABOUT PRIVACY?

However: data contains **confidential** information

Modern Privacy of Data Analysis

Is public analysis of **private** data a meaningful/achievable Goal?

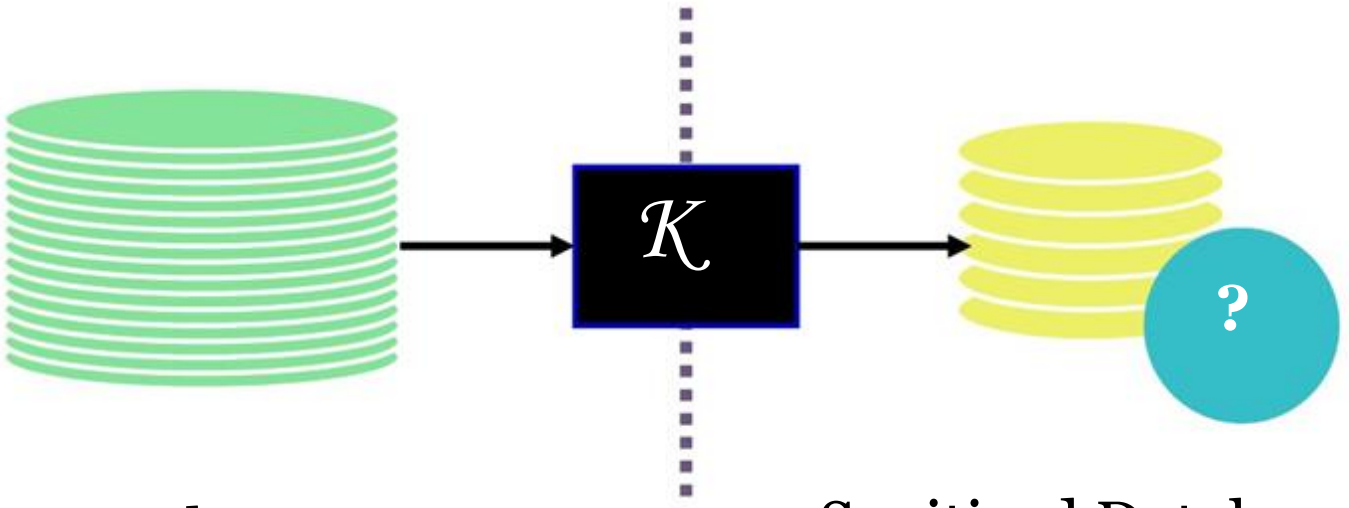
The holy grail:

Get **utility** of statistical analysis while **protecting privacy** of every **individual** participant

Ideally:

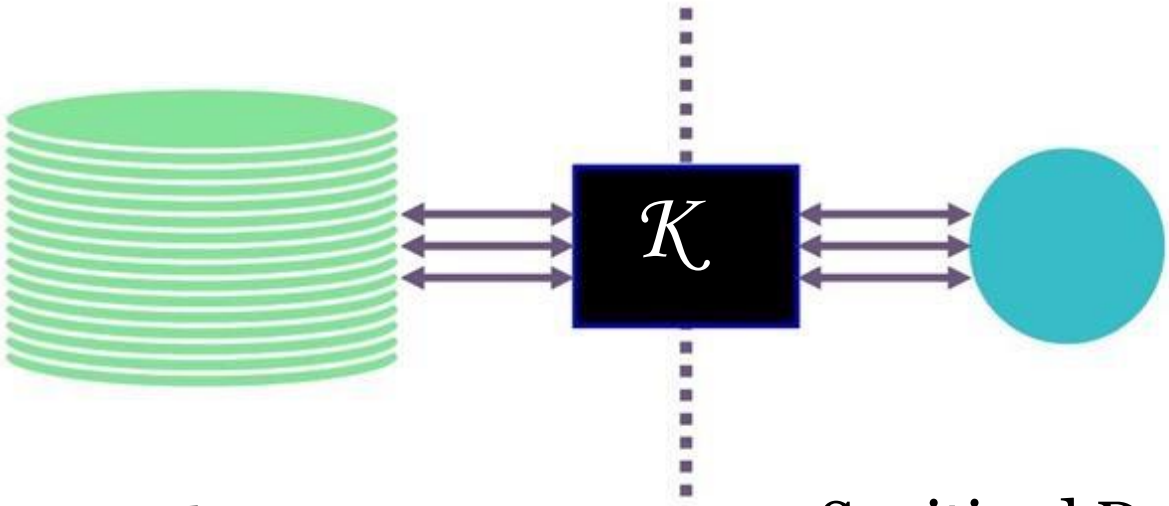
"privacy-preserving" sanitization allows reasonably **accurate answers** to **meaningful information**

Two Models: privacy preserving data publishing —-studied



Non-Interactive: Data are sanitized and released

Two Models: privacy preserving data services/query



Database

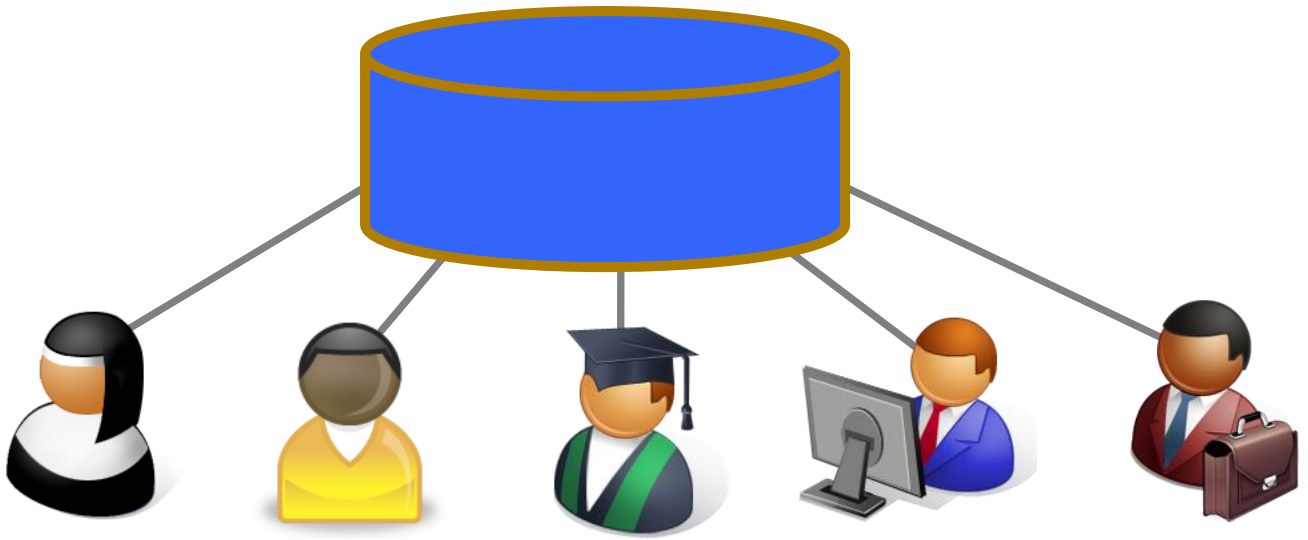
Sanitized Database

Interactive: Multiple Queries, Adaptively Chosen

General Setting

Medical data
Query logs
Social network data
…

Data mining
Statistical queries



Auxiliary Information

- Information from any source *other* than the statistical database
  - Other databases, including old releases of this one
  - Newspapers
  - General comments from insiders
  - Government reports, census website
  - Inside information from a *different* organization
    - Eg, Google's view, if the attacker/user is a Google employee

Linkage Attacks: Use of Aux Info

Semantic Security for Confidentiality
[Goldwasser-Micali ' 82]

Vocabulary

- Plaintext: the message to be transmitted
- Ciphertext: the encryption of the plaintext
- Auxiliary information: anything else known to attacker

The ciphertext leaks no information about the plaintext.

Formalization

- Compare the ability of someone seeing aux and ciphertext to guess (anything about) the plaintext, to the ability of someone seeing only aux to do the same thing. Difference should be "tiny".

Semantic Security for Privacy?

Dalenius, 1977

> Anything that can be learned about a respondent from the statistical database can be learned without access to the database.

Happily, Formalizes to Semantic Security

Unhappily, Unachievable [Dwork and Naor 2006]

> Both for not serious and serious reasons.

What can we do **efficiently**?

Allowed **"too" much power to the adversary**

- Number of queries: exponential
- Computation: exponential
- On the other hand: lack of **wild errors** in the responses

Theorem: For any sanitization algorithm:

If **all** responses are within $o(\sqrt{n})$ of the true answer, then it is blatantly non-private **even against a polynomial time adversary** making $O(n \log^2 n)$ random queries.

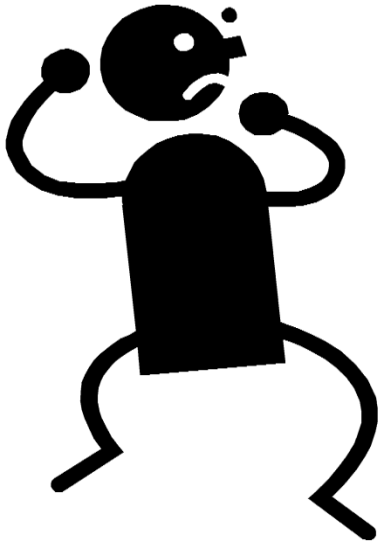
How can you allow meaningful usage of such datasets while preserving individual privacy?

Blatant Non-Privacy

- Leak individual records
- Can link with public databases to re-identify individuals
- Allow adversary to reconstruct database with significant probability

Attempt 1: Crypto-ish Definitions

I am releasing some useful statistic f(D), and nothing more will be revealed.

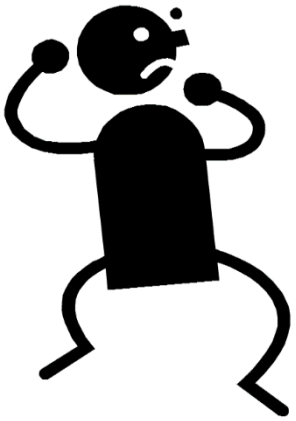


What kind of statistics are safe to publish?

How do you define privacy?

Attempt 2:

I am releasing researching findings showing that people who smoke are very likely to get cancer.



You cannot do that, since it will break my privacy. My insurance company happens to know that I am a smoker…

Attempt 2: Absolute Disclosure Prevention

"If the release of statistics *S* *makes it possible to* determine the value [of private information] more accurately than is possible without access to *S*, a disclosure has taken place." [Dalenius]

An Impossibility Result

[informal] It is not possible to design any non-trivial mechanism that satisfies such strong notion of privacy.

2. Past attempts

- ***Large query sets*:** Forbid queries about specific individuals.
  - **Non-specific queries can still reveal information.**

- ***Query auditing*:** Determine by analysis if a set of queries will reveal information about individuals.
  - **Computationally infeasible in general [J. Kleinberg et al.].**
  - **Rejecting a query leaks information.**

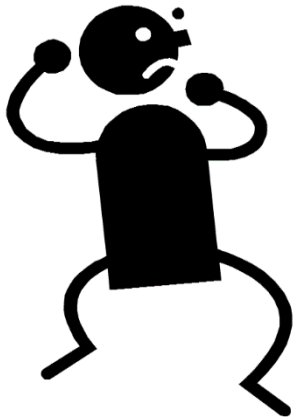
- ***Subset sampling*:** Release only a subset of the dataset.
  - **Punishes individuals in the subsample**

2. Past attempts

- ***Input perturbation*:** Choose a subsample based on the query and compute the response on that subsample.
  - **Does not protect outliers.**
- ***Randomized response*:** Randomize the data at collection time. "Randomize once."
  - **Does not work with complex data.**
- ***Random noise*:** Add random noise to query responses.
  - **If done naively, easy to defeat.**

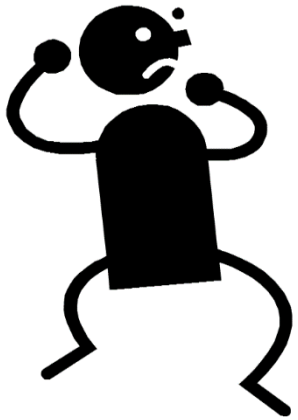
Attempt 3: “Blending into Crowd” or k-Anonymity

K people purchased A and B, and all of them also purchased C.



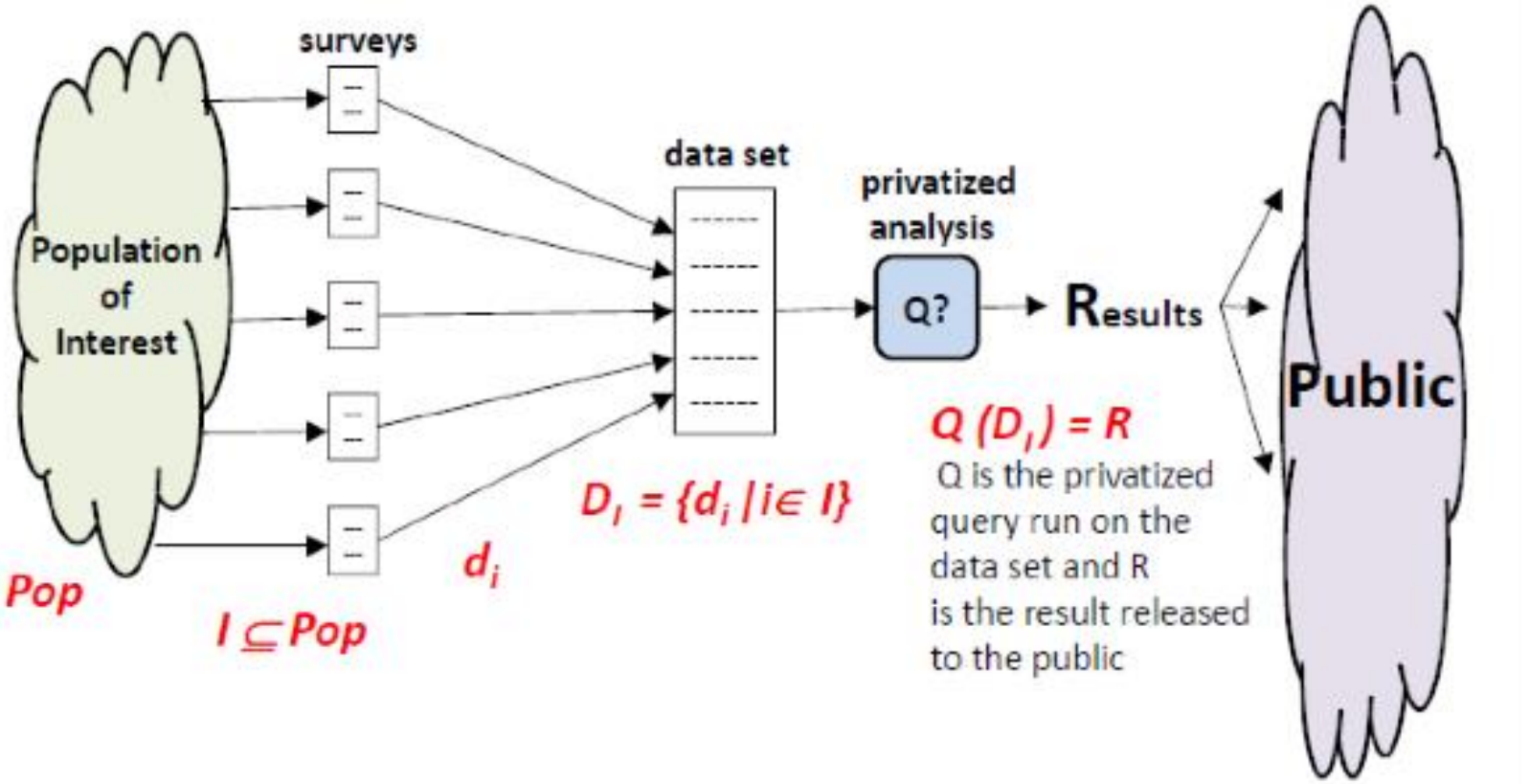
Attempt 3: “Blending into Crowd” or k-Anonymity

K people purchased A and B, and all of them also purchased C.



I know that Elaine bought A and B…

Participating a Survey?



When is safe?

I knew that my answer had no impact on the released results.

- $Q(D_{(I\text{-}me)}) = Q(D_I)$

I knew that any attacker looking at the published results R couldn't learn (with any high probability) any new information about me personally.

- $Prob(secret(me) \mid R) = Prob(secret(me))$

Why not?

If individual answers had no impact on the released results… Then the results would have no utility

If R shows there's a strong trend in my population (everyone is age 10-15 and likes Justin Bieber), with high probability, the trend is true of me too (even if I don't submit a survey).

- By induction,
  $Q(D_{(I\text{-}me)}) = Q(D_I) \Rightarrow$
  $Q(D_I) = Q(D_\varnothing)$

- Prob(*secret*(me) | *secret*(Pop))
  > Prob(*secret*(me))

Why not?

Even worse, if an attacker knows a function about me that's dependent on general facts about the population:

- I'm twice the average age
- I'm in the minority gender

Then releasing just those general facts gives the attacker specific information about me. (Even if I don't submit a survey!)

- $(age(\text{me}) = 2*\text{mean\_age}) \wedge$
  $(gender(\text{me}) \neq \text{mode\_gender}) \wedge$
  $(\text{mean\_age} = 14) \wedge$
  $(\text{mode\_gender} = F) \Rightarrow$

  $(age(\text{me}) = 28) \wedge$
  $(gender(\text{me}) = M)$

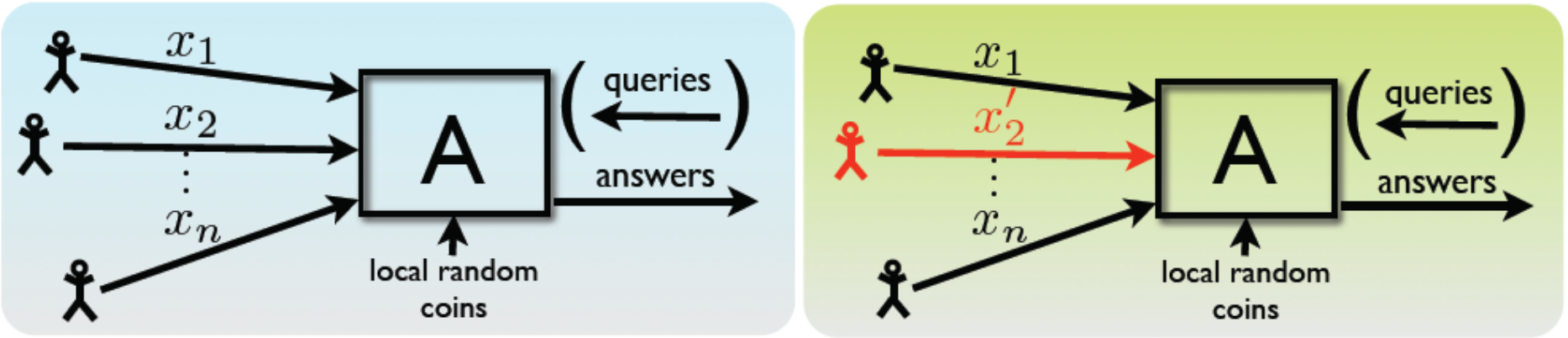
Disappointing Facts

- We cannot promise my data won't affect the result
- We cannot prevent an attacker from learning something about me with some background knowledge!
- What we can do then?

One more try?

- If the chance that the privatized release of the query (or statistics) would be nearly the same, whether or not you participated in the action.

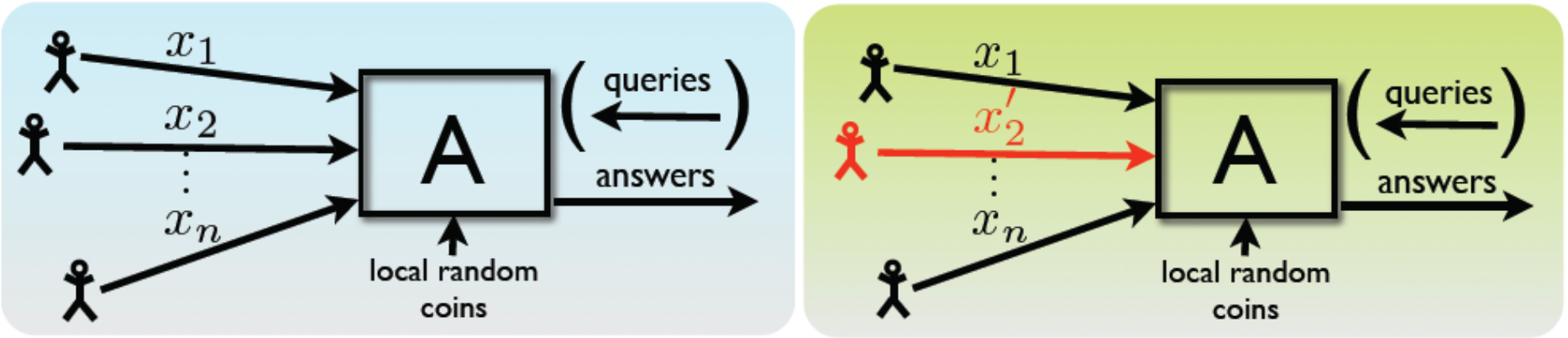
Attempt 4: Differential Privacy



x' is a neighbor of x
if they differ in one row

From the released statistics, it is hard to tell which case it is.

Attempt 4: Differential Privacy



x' is a neighbor of x
if they differ in one row

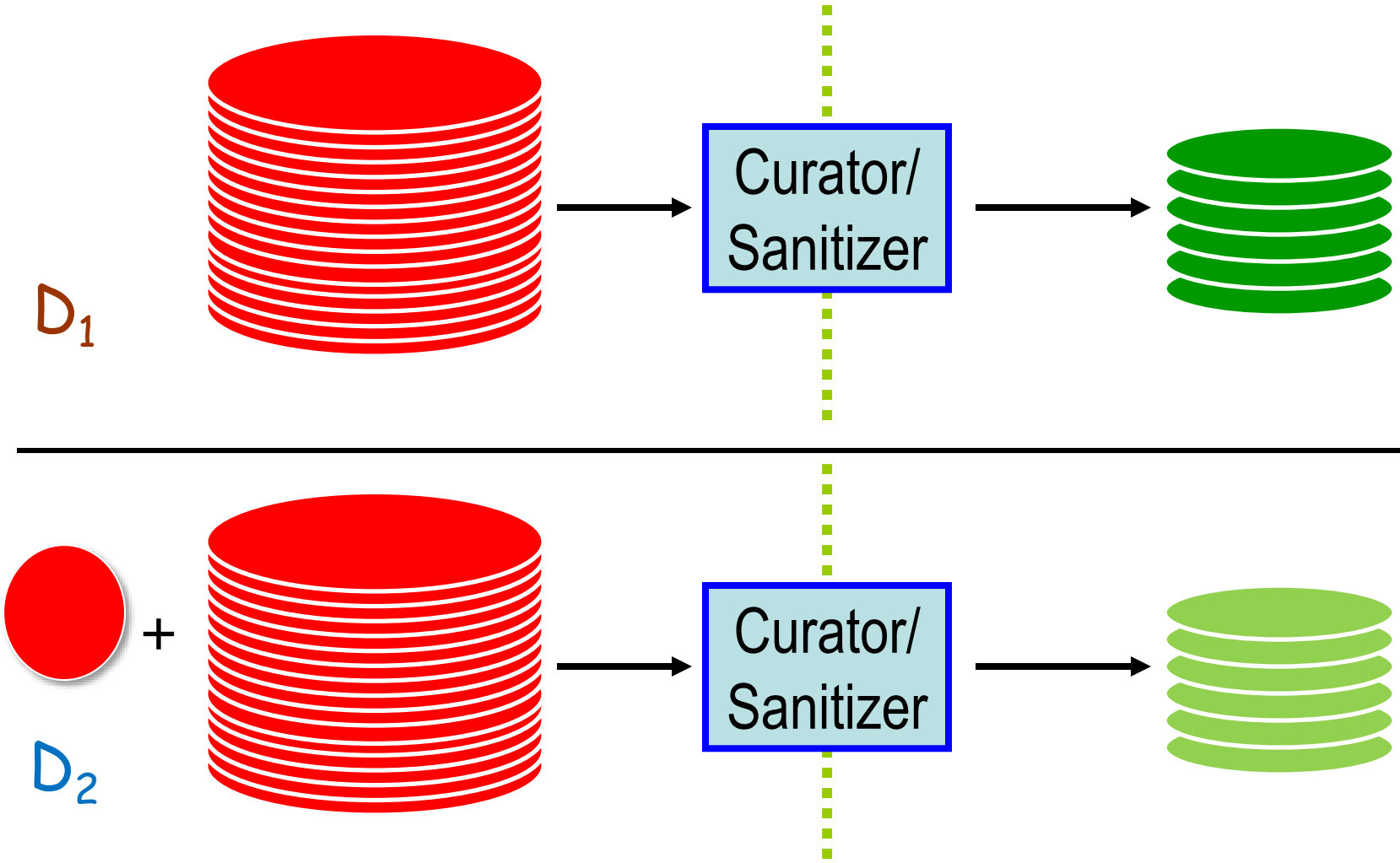
For all neighboring databases x and x'
For all subsets of transcripts:

$$\Pr[A(x) \in S] \leq e^{\varepsilon} \Pr[A(x') \in S]$$

Differential Privacy

Dwork, McSherry
Nissim & Smith
2006

Protect *individual* participants:



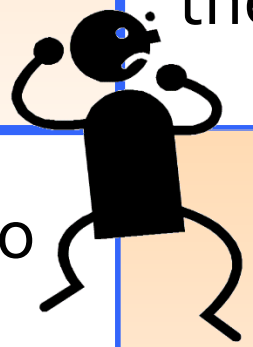
Attempt 4: Differential Privacy

I am releasing researching findings showing that people who smoke are very likely to get cancer.

1

Please don't blame me if your insurance company knows that you are a smoker, since I am doing the society a favor.

2



Oh, btw, please feel safe to participate in my survey, since you have nothing more to lose.

3

Since my mechanism is DP, whether or not you participate, your privacy loss would be roughly the same!

4

Sensitivity of Function

How Much Can f(DB + Me) Exceed f(DB - Me)?

Recall: $\mathcal{K}$(f, DB) = f(DB) + noise

Question Asks: What difference must noise obscure?

$$\Delta f = \max_{H(DB,\, DB')=1} ||f(DB) - f(DB')||_1$$

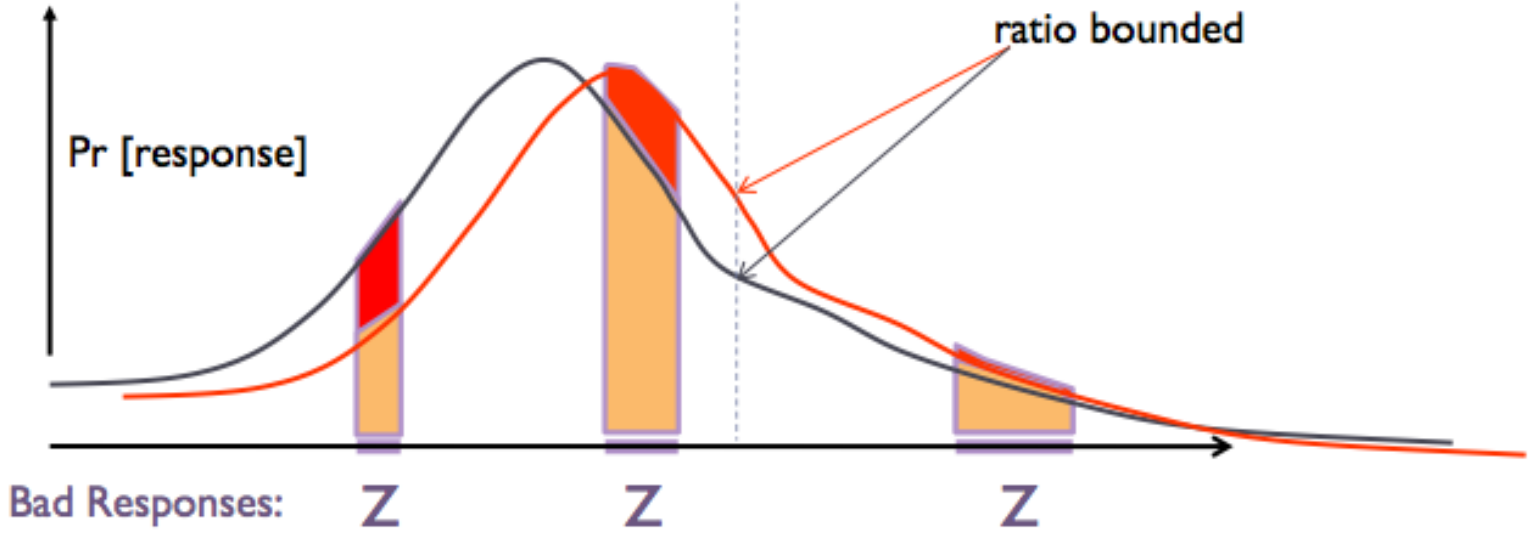
eg, ΔCount = 1

Differential privacy

Definition 2. A randomized function K gives ϵ-differential privacy if for all data sets D and D' differing on at most one row, and all $S \subseteq Range(K)$,

$$\Pr[K(D) \in S] \leq \exp(\epsilon) \times \Pr[K(D') \in S],$$

where the probability space in each case is over the coin flips of K.



Notable Properties of DP

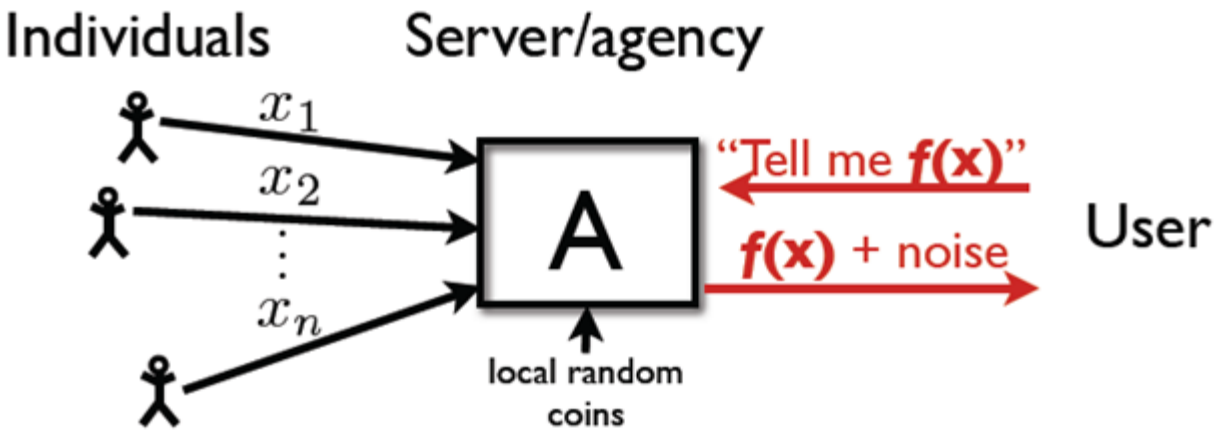
- Adversary knows arbitrary auxiliary information
  - No linkage attacks
- Oblivious to data distribution
- Sanitizer need not know the adversary's prior distribution on the DB

Notable Properties of DP

- Post-processing
- Non-trivial differentially private mechanisms cannot be deterministic
- Composability
  - If M_1, M_2 achieves ϵ_1 , ϵ_2 DP respectively, then M = (M_1, M_2) achieves $\epsilon_1 + \epsilon_2$ DP.

DP Techniques

Output Perturbation



- Global Sensitivity:

$$GS_f = max\|f(x) - f(x')\|_1$$

x,x′ neighbors

Example: $GS_{avg} = \frac{1}{n}$

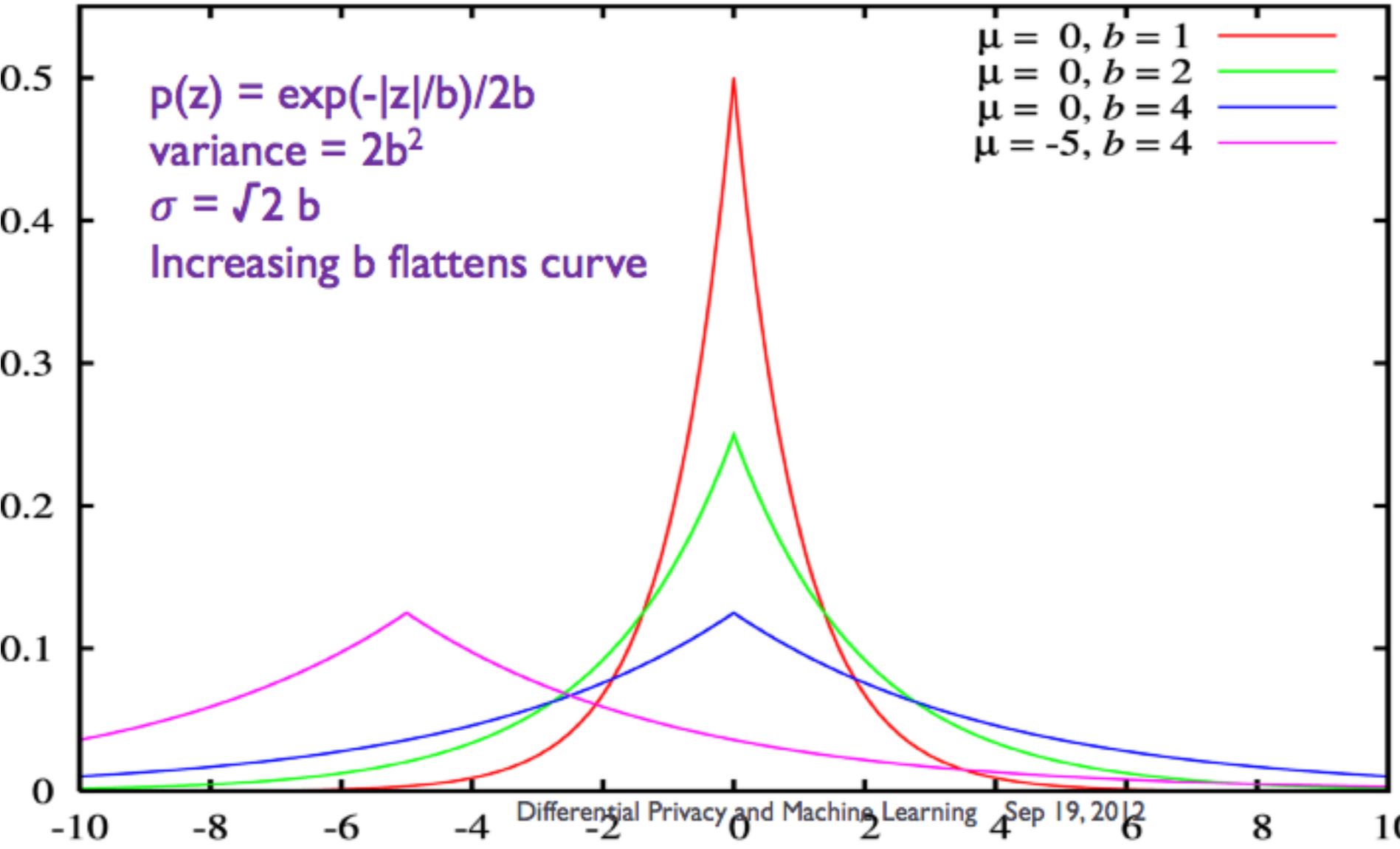
Output Perturbation: Continuous Output

Theorem:

$$A(x) = f(x) + \text{Lap}\left(\frac{GS_f}{\epsilon}\right) \text{ is } \epsilon\text{-DP}$$

- Intuition: add more noise when function is sensitive

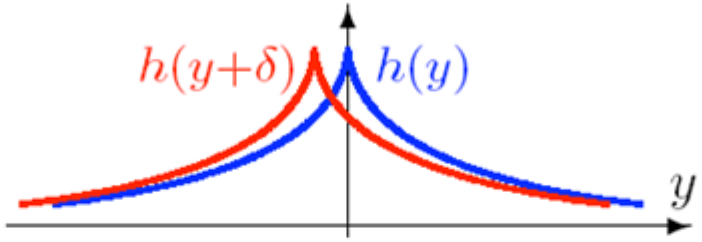
Lap(b)


p(z) = exp(-|z|/b)/2b
variance = 2b²
σ = √2 b
Increasing b flattens curve
μ = 0, b = 1
μ = 0, b = 2
μ = 0, b = 4
μ = -5, b = 4
0.5
0.4
0.3
0.2
0.1
0
-10
-8
-6
-4
-2
0
2
4
6
8
10

Output Perturbation: Continuous Output

$$A(x) = f(x) + \mathsf{Lap}\left(\frac{GS_f}{\epsilon}\right) \text{ is } \epsilon\text{-DP}$$

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:

$A(x)$: blue curve

$A(x')$: red curve

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

Proof Ideas

$$\frac{Pr(f(x)+Lap(\Delta f/\epsilon)=y)}{Pr(f(x')+Lap(\Delta f/\epsilon)=y)} = \frac{\exp\left(-\frac{|y-f(x)|\,\epsilon}{\Delta f}\right)}{\exp\left(-\frac{|y-f(x')|\epsilon}{\Delta f}\right)}$$

$$= \exp\left(\frac{\epsilon}{\Delta f}\left(|y-f(x')|-|y-f(x)|\right)\right)$$

$$\leq \exp\left(\frac{\epsilon}{\Delta f}\left(|f(x)-f(x')|\right)\right) \leq e^{\epsilon}$$

Sequence of queries

Given any query sequence $f_1, ..., f_m$, ϵ-differential privacy can be achieved by running K with noise distribution $\text{Lap}(\Sigma_{i=1}^{m} \Delta f_i / \epsilon)$ on each query.

Sequence of queries

Given any query sequence $f_1, ..., f_m$, ϵ-differential privacy can be achieved by running K with noise distribution $\text{Lap}(\Sigma_{i=1}^{m} \Delta f_i/\epsilon)$ on each query.

- We allow the quality of each answer to deteriorate with the sum of sensitivities of the queries, maintaining ε-differential privacy.
- A complex query need only be penalized by its aggregate sensitivity. This may be surprisingly small.
  - Example: the number of 2-bit rows whose entries are both 1 has sensitivity 1 despite involving 2 bits per row.

Examples of Low Global Sensitivity

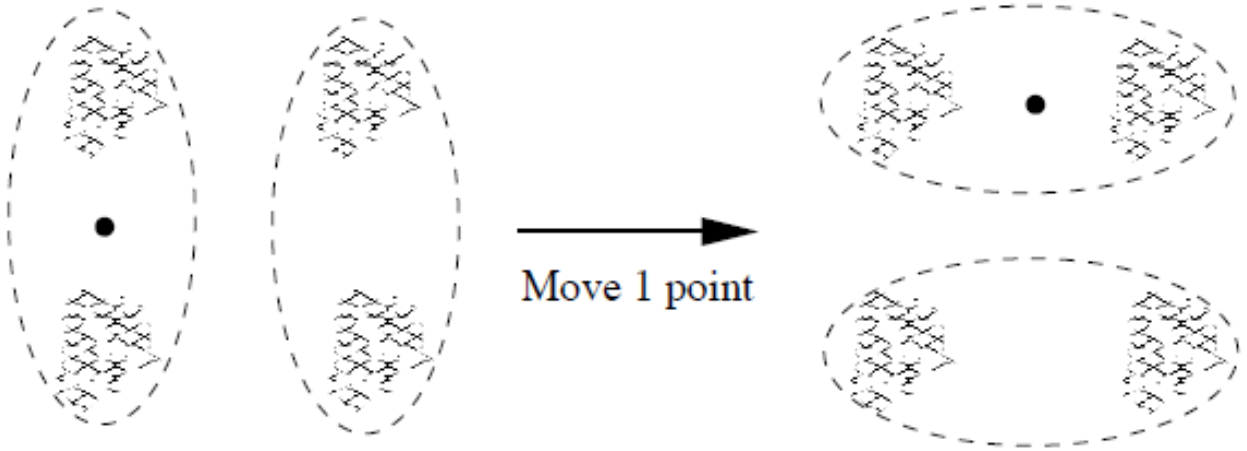
- Average
- Histograms and contingency tables
- Covariance matrix
- [BDMN] Many data-mining algorithms can be implemented through a sequence of low-sensitivity queries
  - Perceptron, some EM algorithms, SQ learning algorithms

Histogram queries

- Histogram queries are an example of the aforementioned principle.
- The addition or removal of a row can only change the count in a bucket by 1.
- This means we need only perturb the count in each bucket according to $\mathrm{Lap}(\Delta f / \varepsilon) = \mathrm{Lap}(1 / \varepsilon)$.
- The cost in noise of a query has the desired property: it increases when the query threatens individual privacy and shrinks when the query concerns an aggregate value.

Examples of High Global Sensitivity

- Order statistics
- Clustering



Vector Function

- **Add noise for each dimension.**

Definition 3.3 (The Laplace Mechanism). Given any function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(x, f(\cdot), \varepsilon) = f(x) + (Y_1, \ldots, Y_k)$$

where Y_i are i.i.d. random variables drawn from $\mathrm{Lap}(\Delta f/\varepsilon)$.

Definition 3.1 (ℓ_1-sensitivity). The ℓ_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is:

$$\Delta f = \max_{\substack{x,y \in \mathbb{N}^{|\mathcal{X}|} \\ \|x-y\|_1 = 1}} \|f(x) - f(y)\|_1.$$

Accuracy Bound of Query Answer

- **Bound on how accurate is the answer respect to the true answer**

Theorem 3.8. Let $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, and let $y = \mathcal{M}_L(x, f(\cdot), \varepsilon)$. Then $\forall \delta \in (0, 1]$:

$$\Pr\left[\|f(x) - y\|_\infty \geq \ln\left(\frac{k}{\delta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] \leq \delta$$

Proof. We have:

$$\begin{aligned}
\Pr\left[\|f(x) - y\|_\infty \geq \ln\left(\frac{k}{\delta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] &= \Pr\left[\max_{i \in [k]} |Y_i| \geq \ln\left(\frac{k}{\delta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] \\
&\leq k \cdot \Pr\left[|Y_i| \geq \ln\left(\frac{k}{\delta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] \\
&= k \cdot \left(\frac{\delta}{k}\right) \\
&= \delta
\end{aligned}$$

Fact 3.7. If $Y \sim \mathrm{Lap}(b)$, then:

$$\Pr[|Y| \geq t \cdot b] = \exp(-t).$$

Some Examples

Randomized Response

1. Flip a coin.
2. If **tails**, then respond truthfully.
3. If **heads**, then flip a second coin and respond "Yes" if heads and "No" if tails.

- **Randomized Response Mechanism is DP**

Claim 3.5. The version of randomized response described above is $(\ln 3, 0)$-differentially private.

Randomized Response

Proof. Fix a respondent. A case analysis shows that $\Pr[\text{Response} = \text{Yes} | \text{Truth} = \text{Yes}] = 3/4$. Specifically, when the truth is "Yes" the outcome will be "Yes" if the first coin comes up tails (probability 1/2) or the first and second come up heads (probability 1/4)), while $\Pr[\text{Response} = \text{Yes} | \text{Truth} = \text{No}] = 1/4$ (first comes up heads and second comes up tails; probability 1/4). Applying similar reasoning to the case of a "No" answer, we obtain:

$$\frac{\Pr[\text{Response} = \text{Yes} | \text{Truth} = \text{Yes}]}{\Pr[\text{Response} = \text{Yes} | \text{Truth} = \text{No}]}$$

$$= \frac{3/4}{1/4} = \frac{\Pr[\text{Response} = \text{No} | \text{Truth} = \text{No}]}{\Pr[\text{Response} = \text{No} | \text{Truth} = \text{Yes}]} = 3.$$

□

Reporting Noisy Max

- **Given an array A[] of values, reporting the index i where A[i] is the maximum**

Report Noisy Max. Consider the following simple algorithm to determine which of m counting queries has the highest value: Add independently generated Laplace noise $\mathrm{Lap}(1/\varepsilon)$ to each count and return the index of the largest noisy count (we ignore the possibility of a tie). Call this algorithm Report Noisy Max.

Claim 3.9. The Report Noisy Max algorithm is $(\varepsilon, 0)$-differentially private.

Objections or Research Questions?

1. **Do enough queries and we lose privacy.**
   Does this make sense? We just stop using the data?

2. **This work presumes a static dataset.**
   What happens to the analysis if the data is changing?

4. **Are there static parameters of the database that should be hidden as well?**

6. **Using differential privacy does not stop many other forms of information leakage.**

8. **Whoever does the calculation has the data.**

10. **How limited is the query model?**
    The standard database model does not fit with differential privacy.

Query model

- If it is natural to put things in terms of a histogram, then it is natural to query.

- Absurd example from the *Scientific American* article:

"If you wanted to generate a list of the top 100 baby names for 2012, for example, you could ask a series of questions of the form, "How many babies were given names that start with A?" (or Aa, Ab or Ac), and work your way through the possibilities."

Query model

"'One of the early results in machine learning is that almost everything that is possible in principle to learn can be learned through counting queries,' [Aaron] Roth said. 'Counting queries are not isolated toy problems, but a fundamental primitive' — that is, a building block from which many more complex algorithms can be built."

- This does not seem like a query model most people can use naturally.
- Perhaps Dwork defined away this problem by focusing on *statistical databases*.

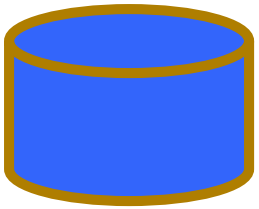
PINQ

http://research.microsoft.com/en-us/projects/pinq/default.aspx

PINQ

- Language for writing differentially-private data analyses
- Language extension to .NET framework
- Provides a SQL-like interface for querying data
- Goal: Hopefully, non-privacy experts can perform privacy-preserving data analytics

Scenario



Trusted curator

Query through
PINQ interface

Data analyst

Example 1

```
static void Main(string[] args)
{
    var source = Enumerable.Range(1, 1000).AsQueryable();
    var agent = new PINQAgentBudget(1.0);

    var data = new PINQueryable<int>(source, agent);

    Console.WriteLine("count: " + data.NoisyCount(0.01));
    Console.WriteLine("count: " + data.NoisyCount(0.10));
    Console.WriteLine("count: " + data.NoisyCount(1.00));
}
```

Composition and privacy budget

- Sequential composition
- Parallel composition

K-Means: Privacy Budget Allocation

- Total budget ϵ
- Guess number of iterations m
- Allocate $\frac{\epsilon}{m}$ per iteration

Privacy Budget Allocation

- Allocation between users/computation providers
  - Auction?
- Allocation between tasks
- In-task allocation
  - Between iterations
  - Between multiple statistics
  - Optimization problem

No satisfactory solution yet!

When Budget Has Exhausted

?

DP Pros, Cons, and Challenges?

- Utility v.s. privacy
- Privacy budget management and depletion
- Allow non-experts to use?
- Many non-trivial DP algorithms require really large datasets to be *practically* useful
- What privacy budget is reasonable for a dataset?
  - Implicit independence assumption? Consider replicating a DB k times

Privacy Loss?

ε–Διφφερεντιαλ Πριωαχψ

Kullback–Leibler Divergence or Relative Entropy

- The **KL-Divergence** or **Relative Entropy** between two random variables Y and Z is

  taking values from the same domain

Not a metric

$$KL(Y||Z) = E_{y\sim Y}\left[\ln \frac{\Pr[Y=y]}{\Pr[Z=y]}\right]$$

  - Non-symmetric measure of the difference
  - Measures the **expected** number of **extra** bits needed to code samples from Y when using a code based on Z.
  - Typically: Y represents the ``*true*'' distribution of data and Z represents an approximation or model of Y.

The Max Divergence

- The **Max-Divergence** between two random variables Y and Z is

  taking values from the same domain

$$KL_{\infty}(Y||Z) = Max_{S \subset Supp(Y)} \left[\ln \frac{\Pr[Y \in S]}{\Pr[Z \in S]}\right]$$

The **δ-Approximate Max-Divergence** is

$$KL_{\infty}^{\delta}(Y||Z) = Max_{S \subset Supp(Y), \Pr[Y \in S] \geq \delta} \left[\ln \frac{\Pr[Y \in S] - \delta}{\Pr[Z \in S]}\right]$$

Max Divergence and KL-Divergence

- Max Divergence (exactly the definition of ϵ!) :

$$D_\infty(Y||Z) = \max_{S \subset Supp(Y)} \left[\ln \frac{\Pr[Y \in S]}{\Pr[Z \in S]} \right]$$

- KL Divergence (average divergence)

$$D(Y||Z) = \mathrm{E}_{y \sim Y} \left[\ln \frac{\Pr[Y = y]}{\Pr[Z = y]} \right]$$

- The Useful Lemma gives a bound on KL-divergence.

$$\mathrm{D}(\mathrm{Y} \parallel \mathrm{Z}) \leq \epsilon(e^\epsilon - 1)$$

$$\epsilon(e^\epsilon - 1) \leq 2\epsilon^2 \; when \; \epsilon < 1$$

Martingales and Azuma's Inequality

A sequence of random variables $X_0, X_1 \ldots, X_m$ is a **martingale** if for $0 < i < m-1$

$$E[X_{i+1} | X_i] = X_i$$

Example: sum a gambler has in series of fair bets

Azuma's Inequality: Let $X_0=0, X_1 \ldots, X_m$ be a martingale with $|X_{i+1} - X_i| < 1$

Then for any $\lambda > 0$ we have

$$\Pr[X_m > \lambda \sqrt{m}] < e^{-\lambda^2/2}$$

High concentration around expectation

Azuma's Inequality (general)

Lemma 3.19 (Azuma's Inequality). Let $C_1, \ldots, C_k$ be real-valued random variables such that for every $i \in [k]$, $\Pr[|C_i| \leq \alpha] = 1$, and for every $(c_1, \ldots, c_{i-1}) \in \mathrm{Supp}(C_1, \ldots, C_{i-1})$, we have

$$\mathrm{E}[C_i | C_1 = c_1, \ldots, C_{i-1} = c_{i-1}] \leq \beta.$$

Then for every $z > 0$, we have

$$\Pr\left[\sum_{i=1}^{k} C_i > k\beta + z\sqrt{k} \cdot \alpha\right] \leq e^{-z^2/2}.$$

Azuma's Inequality (more general)

Theorem 3.3 (Azuma's Inequality). Let f be a function of m random variables $X_1, \ldots, X_m$, each X_i taking values from a set A_i such that $\mathbb{E}[f]$ is bounded. Let c_i denote the maximum effect of X_i on f — i.e., for all $a_i, a_i' \in A_i$:

$$|\mathbb{E}[f|X_1, \ldots, X_{i-1}, X_i = a_i] - \mathbb{E}[f|X_1, \ldots, X_{i-1}, X_i = a_i']| \leq c_i$$

Then:

$$\Pr[f(X_1, \ldots, X_m) \geq \mathbb{E}[f] + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{m} c_i^2}\right)$$

Extensions of DP

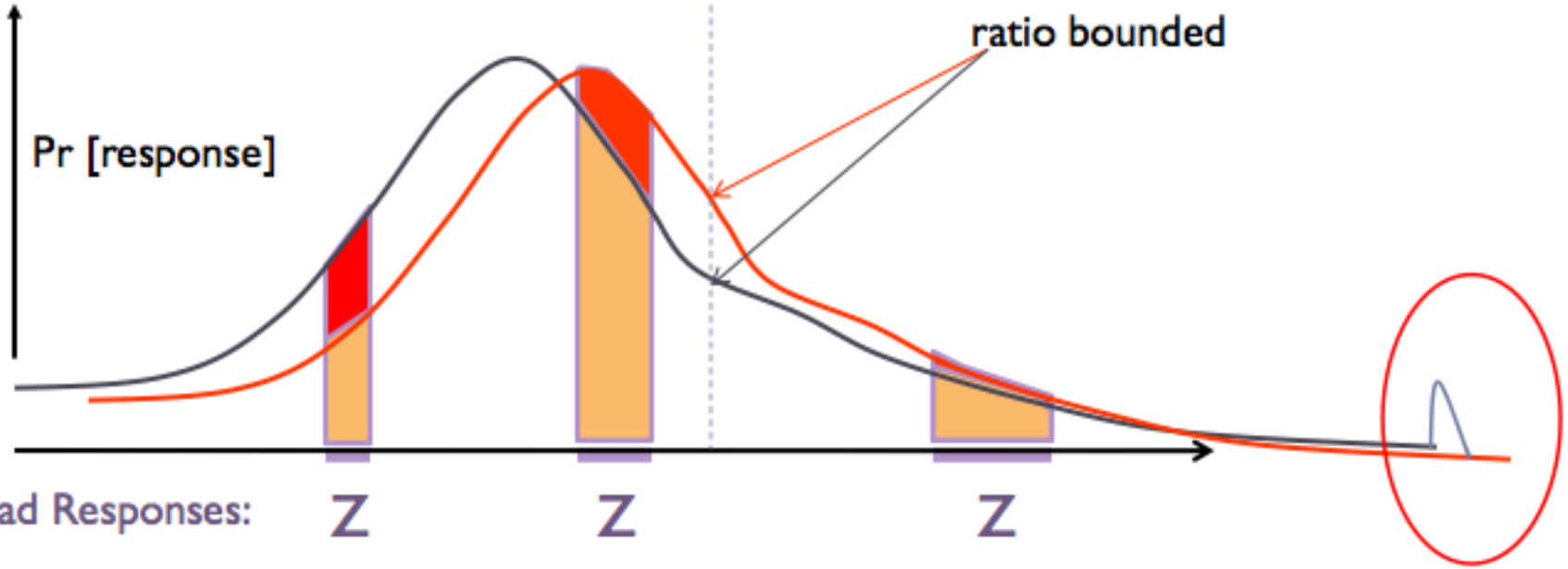
(ε ,™)–Διφφερεντιαλ Πριωαχψ

A Natural Relaxation: (ε, ™)-Differential Privacy

$\mathcal{M}$ gives (ϵ, δ)- differential privacy if for all adjacent x and x', and all $C \subseteq range(\mathcal{M})$: $\Pr[\mathcal{M}(D) \in C] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in C] + \delta$

Neutralizes all linkage attacks.

Composes unconditionally and automatically: $(\Sigma_i \varepsilon_i, \Sigma_i \delta_i)$



Simple Composition

- **k-fold composition of (ε, δ)-differentially private mechanisms is (kε, k δ)-differentially private.**
  - If want to keep original guarantee, must inject k times the noise
  - When k is large, this destroys utility of the output

- Can we do better than that by again leveraging the tradeoff?
  - Trade-off a little δ with a lot of ϵ?

What is Composition

1. Repeated use of differentially private algorithms on the same database. This allows both the repeated use of the same mechanism multiple times, as well as the modular construction of differentially private algorithms from arbitrary private building blocks.
2. Repeated use of differentially private algorithms on *different* databases that may nevertheless contain information relating to the same individual. This allows us to reason about the cumulative privacy loss of a single individual whose data might be spread across multiple data sets, each of which may be used independently in a differentially private way. Since new databases are created all the time, and the adversary may actually influence the makeup of these new databases, this is a fundamentally different problem than repeatedly querying a single, fixed, database.

Adversaries

We want to model composition where the adversary can adaptively affect the databases being input to future mechanisms, as well as the queries to those mechanisms. Let $\mathcal{F}$ be a family of database access mechanisms.

Experiment b for family $\mathcal{F}$ and adversary A:

For $i = 1, \ldots, k$:

1. A outputs two adjacent databases x_i^0 and x_i^1, a mechanism $\mathcal{M}_i \in \mathcal{F}$, and parameters w_i.
2. A receives $y_i \in_R \mathcal{M}_i(w_i, x_{i,b})$.

We allow the adversary A above to be stateful throughout the experiment, and thus it may choose the databases, mechanisms, and the parameters adaptively depending on the outputs of previous mechanisms. We define A's *view* of the experiment to be A's coin tosses and all of the mechanism outputs $(y_1, \ldots, y_k)$. (The x_i^j's, $\mathcal{M}_i$'s, and w_i's can all be reconstructed from these.)

Advanced Composition

Definition 3.7. We say that the family $\mathcal{F}$ of database access mechanisms satisfies *ε-differential privacy under k-fold adaptive composition* if for every adversary A, we have $D_\infty(V^0 \| V^1) \leq \varepsilon$ where V^b denotes the view of A in k-fold Composition Experiment b above.

(ε, δ)-differential privacy under k-fold adaptive composition instead requires that $D^\delta_\infty(V^0 \| V^1) \leq \varepsilon$.

Theorem 3.20 (Advanced Composition). For all $\varepsilon, \delta, \delta' \geq 0$, the class of (ε, δ)-differentially private mechanisms satisfies $(\varepsilon', k\delta + \delta')$-differential privacy under k-fold adaptive composition for:

$$\varepsilon' = \sqrt{2k \ln(1/\delta')}\varepsilon + k\varepsilon(e^\varepsilon - 1).$$

Extensions of DP

Γαυσσιαν Νοισε

Gaussian Noise and Gaussian Mechanism

Definition 3.8 (ℓ_2-sensitivity). The ℓ_2-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is:

$$\Delta_2(f) = \max_{\substack{x,y \in \mathbb{N}^{|\mathcal{X}|} \\ \|x-y\|_1=1}} \|f(x) - f(y)\|_2.$$

The *Gaussian Mechanism* with parameter b adds zero-mean Gaussian noise with variance b in each of the k coordinates. The following

Theorem 3.22. Let $\varepsilon \in (0,1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta_2(f)/\varepsilon$ is (ε, δ)-differentially private.

Exponential Mechanism for Discrete Outputs

Discrete-valued functions: $f(x) \in R = \{y_1, y_2, \ldots, y_k\}$

- Strings, experts, small databases, …

The Exponential Mechanism [McSherry Talwar]

A general mechanism that yields

- Differential privacy
- ***May*** yield utility/approximation
- Is defined and evaluated by considering **all possible answers**

The definition does not yield an efficient way of evaluating it

Application/original motivation:

Approximate truthfulness of auctions

- Collusion resistance
- Compatibility

Exponential Mechanism

- Define utility function:
  - Each $y \in R$ has a utility for x, denoted $q(x, y)$

- Exponential Mechanism [McSherry-Talwar'07]

Output y with probability $\propto e^{\frac{\epsilon q(x, y)}{2\Delta q}}$

  - Idea: Make high utility outputs exponentially more likely at a rate that depends on the sensitivity of $q(x, y)$.

It is Private

Theorem: The Exponential Mechanism preserves $(\epsilon, 0)$-differential privacy.

Proof: Fix any $D, D' \in \mathbb{N}^{|X|}$ with $\left|\left|D, D'\right|\right|_1 \leq 1$ and any $r \in R$...

$$\frac{\Pr[\text{Exponential}(D, R, q, \epsilon) = r]}{\Pr[\text{Exponential}(D', R, q, \epsilon) = r]} =$$

$$\frac{\left(\frac{\exp(\frac{\epsilon q(D,r)}{2\Delta})}{\sum \exp(\frac{\epsilon q(D,r')}{2\Delta})}\right)}{\left(\frac{\exp(\frac{\epsilon q(D',r)}{2\Delta})}{\sum \exp(\frac{\epsilon q(D',r')}{2\Delta})}\right)} = \left(\frac{\exp(\frac{\epsilon q(D,r)}{2\Delta})}{\exp(\frac{\epsilon q(D',r)}{2\Delta})}\right)\left(\frac{\sum_{r'} \exp(\frac{\epsilon q(D',r')}{2\Delta})}{\sum_{r'} \exp(\frac{\epsilon q(D,r')}{2\Delta})}\right)$$

$$\text{✦} = \left(\frac{\exp(\frac{\epsilon q(D,r)}{2\Delta})}{\exp(\frac{\epsilon q(D',r)}{2\Delta})}\right) =$$

$$\exp\left(\frac{\epsilon(q(D,r)-q(D',r))}{2\Delta}\right) \leq$$

$$\exp\left(\frac{\epsilon\Delta}{2\Delta}\right) = \exp\left(\frac{\epsilon}{2}\right)$$

$$\text{✦✦} = \left(\frac{\sum_{r'}\exp(\frac{\epsilon q(D',r')}{2\Delta})}{\sum_{r'}\exp(\frac{\epsilon q(D,r')}{2\Delta})}\right) \leq$$

$$\left(\frac{\sum_{r'}\exp(\frac{\epsilon(q(D,r')+\Delta)}{2\Delta})}{\sum_{r'}\exp(\frac{\epsilon q(D,r')}{2\Delta})}\right) =$$

$$= \left(\frac{\exp(\frac{\epsilon}{2})\sum_{r'}\exp(\frac{\epsilon q(D,r')}{2\Delta})}{\sum_{r'}\exp(\frac{\epsilon q(D,r')}{2\Delta})}\right) = \exp(\frac{\epsilon}{2})$$

Accuracy

- How good is the output?

  Define:

  $$OPT_q(D) = \max_{r \in R} q(D, r)$$
  $$R_{OPT} = \{r \in R : q(D, r) = OPT_q(D)\}$$
  $$r^* = \text{Exponential}(D, R, q, \epsilon)$$

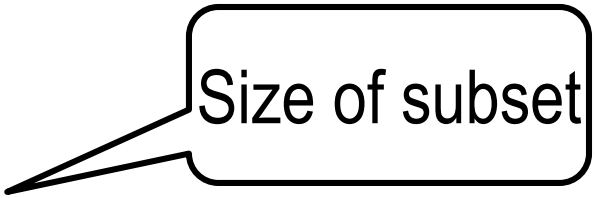
  Theorem:

  $$\Pr\left[q(r^*) \leq OPT_q(D) - \frac{2\Delta}{\epsilon}\left(\log\left(\frac{|R|}{|R_{OPT}|}\right) + t\right)\right] \leq e^{-t}$$

- The results depends ONLY on Δ (logarithm to $|R|$).
- Example: counting query. What is the majority gender that likes Justin Bieber? $|R| = 2$
  - Error is $\frac{2}{\epsilon}(\log(2) + 5)$ with probability $1 - e^{-5}$! Percent error$\to 0$, when number of data become large.

Example of the Exponential Mechanism

- 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 **with probability proportional to** $e^{\varepsilon q(y,X)}$

- Popular sites exponentially more likely than rare ones

  Website scores don't change too quickly

Extended Setting

- For input D in U^n want to find an output r from a set R
- Base measure λ on R - usually uniform
- Score function q': $(U^n, R) \to R$

  assigns any pair (D,r) a real value

  – Want to maximize it (approximately)

The exponential mechanism

  – Assign output r from R with probability proportional to

$$e^{\varepsilon q'(D,r)} \lambda(r)$$

Normalizing factor: $\sum_r e^{\varepsilon q'(D,r)} \lambda(r)$

The exponential mechanism is private

- Let $\otimes = \max_{D,D',r} |q'(D,r)-q'(D',r)|$

adjacent

Claim: The exponential mechanism yields a $2\sum\otimes$ differentially private solution

- Prob[output = r on input D]
  $= e^{\sum q'(D,r)} \lceil (r) / \square_r e^{\sum q'(D,r)} \lceil (r)$
- Prob[output = r on input D']
  $= e^{\sum q'(D',r)} \lceil (r) / \square_r e^{\sum q'(D',r)} \lceil (r)$

Ratio is bounded by $e^{\sum\otimes} e^{\sum\otimes}$

Private Ranking

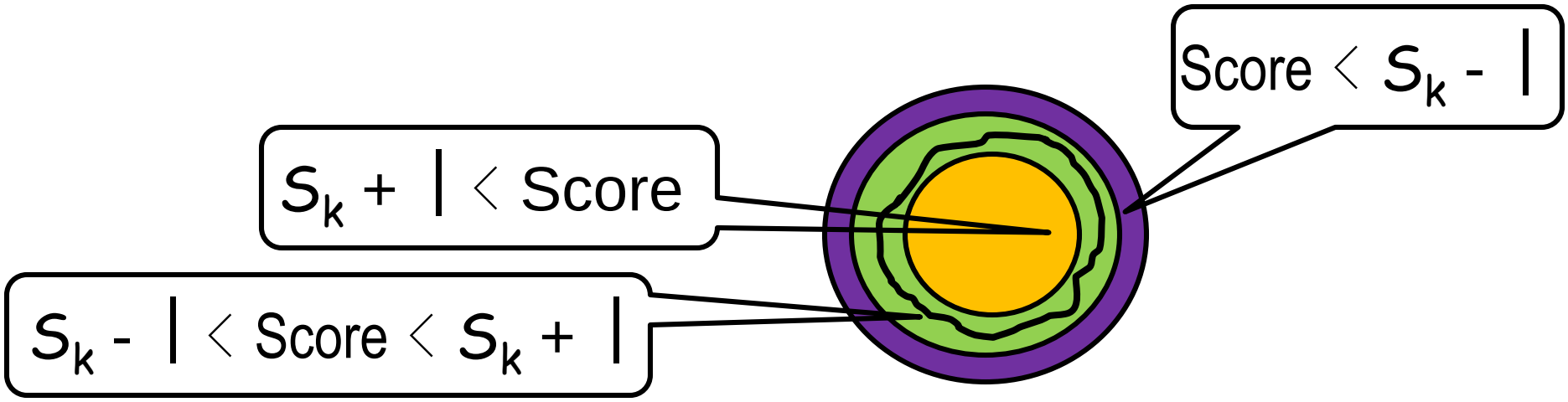
- Each element i in $\{1, \ldots n\}$ has a real valued score $s_D(i)$ based on a data set D as its key.
- **Goal: Output k elements with highest scores.**
- **Privacy**
- Data set D consists of n entries in domain D.
  - **Differential privacy: Protects privacy of entries in D.**
- **Condition: Insensitive Scores**
  - for any element i, for any data sets D, D' that differ in one entry: $|s_D(i) - s_{D'}(i)|$ is at most 1

Approximate ranking

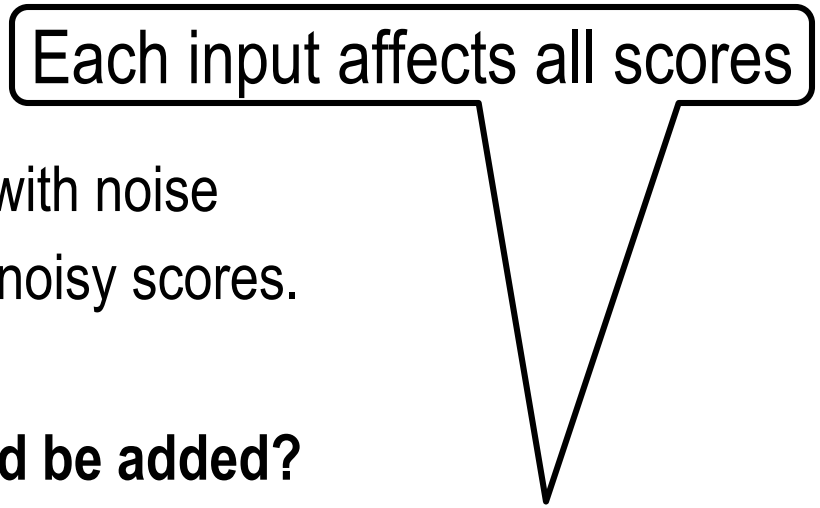
- Let s_k be the k^{th} highest score in the data set D.
- An output list is ε **-useful** if:

  Soundness: No element in the output has score $< s_k - \varepsilon$

  Completeness: Every element with score $> s_k + \varepsilon$ is in the output.



Two Approaches



- **Score perturbation**
  - Perturb the scores of the elements with noise
  - Pick the top **k** elements in terms of noisy scores.
  - Fast and simple implementation

  Question: what sort of noise should be added?

  What sort of guarantees?
- **Exponential sampling**
  - Run the exponential mechanism **k** times.
  - more complicated and slower implementation

  What sort of guarantees?

Extensions of DP

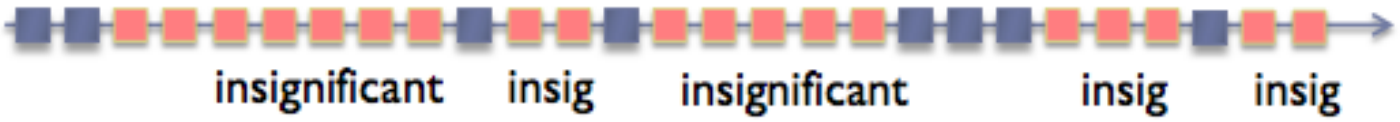
Πριωαχψ ωσ # Θυεριεσ

Queries vs DP Privacy !

- Dinur and Nissim shows that the following negative results:
  - If adversary has exponential computational power (ask exp(n)) questions, a $O(n)$ perturbation is needed for privacy.
  - If adversary has poly(n) computation powers, at least $O(\sqrt{n})$ perturbation is needed.
- They also gave the following positive news:
  - If adversary can ask only less than $T(n)$ questions, then a perturbation error of roughly $\sqrt{T(n)}$ is sufficient to guarantee privacy.

Only Significant Queries?

- Database size n
- # Queries $m \gg n$, eg, m super-polynomial in n
- # "Significant" Queries $k \in O(n)$
  - For now: Counting queries only
  - Significant: count exceeds publicly known threshold T
- Goal: Find, and optionally release, counts for significant queries, *paying only for significant queries*



Algorithm and Analysis

[Hardt-Rothblum]

Caution: Conditional branch leaks private information!

Need noisy threshold $T + Lap(\sigma)$

Algorithm:

When given query f_t:

- If $f_t(x) \leq T$: [insignificant]
  - Output $\perp$
- Otherwise [significant]
  - Output $f_t(x) + Lap(\sigma)$

- First attempt: It's obvious, right?
  - Number of significant queries $k \Rightarrow \leq k$ invocations of Laplace mechanism
  - Can choose σ so as to get error $k^{1/2}$

Algorithm and Analysis

Caution:
Conditional branch leaks private information!

Algorithm:

When given query f_t:

- If $f_t(x) \leq T + Lap(\sigma)$: [insignificant]
  - Output $\perp$
- Otherwise [significant]
  - Output $f_t(x) + Lap(\sigma)$

- Intuition: counts far below T leak nothing
  - Only charge for noisy counts in this range:

Summary of Sparse Vector Tech

Expected total privacy loss $EX = O(\frac{k}{\sigma^2})$

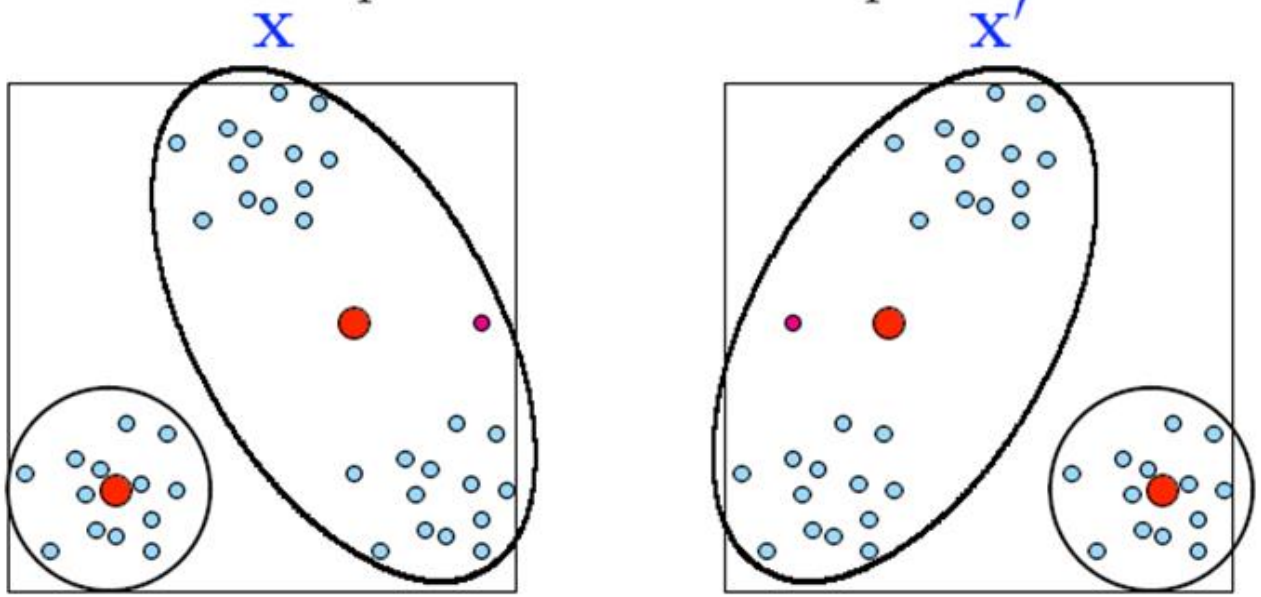
- Probability of (significantly) exceeding expected number of borderline events is negligible (Chernoff)
- Assuming not exceeded: Use Azuma to argue that whp actual total loss does not significantly exceed expected total loss
- Utility: With probability at least $1 - \beta$ all errors are bounded by $\sigma(\ln m + \ln(\frac{1}{\beta}))$.
- Choose $\sigma = 8\sqrt{2\ln(\frac{2}{\delta})(4k + \ln\left(\frac{2}{\delta}\right))/\epsilon}$
  - Linear in k, and only log in m!

Local Sensitivity and Smooth Sensitivity

Adding smaller noise?

Examples of High Global Sensitivity

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.

Examples of High Global Sensitivity

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?

Global Sensitivity vs. Local sensitivity

- Global sensitivity is worst case over inputs

Local sensitivity of query q at point D

$$LS_q(D) = \max_{D'} |q(D) - q(D')|$$

- Reminder: $GS_q(D) = \max_D LS_q(D)$

Goal: add less noise when local sensitivity is small

- Problem: can **leak information** by **amount of noise**

Local Sensitivity

- The sensitivity depends on f but not the range of data.
- Consider f= median income

$$D = \{\underbrace{0, 0, \ldots, 0}_{\frac{n-1}{2}}, 0, \underbrace{k, k, \ldots, k}_{\frac{n-1}{2}}\} \quad D' = \{\underbrace{0, 0, \ldots, 0}_{\frac{n-1}{2}}, k, \underbrace{k, k, \ldots, k}_{\frac{n-1}{2}}\}$$

  - Global sensitivity is k! Perturb by Lap(k/ϵ) gives no utility at all.
- For typical data however we may do better:

$$D = \{1, 2, 2, \ldots, \frac{k}{2}, \frac{k}{2}, \frac{k}{2}, \frac{k}{2} + 1, \ldots, k, k\}$$

The local sensitivity of a function $f : 2^X \to \mathbf{R}$ *at a database* D *is:*

$$LS_f(D) = \max_{D' \in N(D)} |f(D) - f(D')|$$

Local sensitivity of Median

- For $X = x_1, x_2, \ldots, x_n$
- $LS_{median}(X) = \max(x_m - x_{m-1}, x_{m+1} - x_m)$

$x_1, x_2, \ldots, x_{m-1}, x_m, x_{m+1}, \ldots, x_n$

Local Sensitivity: LS

- Local sensitivity is defined to particular data D, but adding $Lap(LS_f(D)/\epsilon)$ doesn't guarantee ϵ-dp.
  - Because $LS_f(D)$ itself is sensitive to the values in D!

$$D = \{\underbrace{0, 0, \ldots, 0}_{\frac{n-1}{2}}, 0, 0, \underbrace{k, k, \ldots, k}_{\frac{n-1}{2}-1}\} \quad D' = \{\underbrace{0, 0, \ldots, 0}_{\frac{n-1}{2}}, 0, \underbrace{k, k, \ldots, k}_{\frac{n-1}{2}}\}$$

$$LS_{\text{median}}(D) = 0, \text{ but } LS_{\text{median}}(D') = k.$$

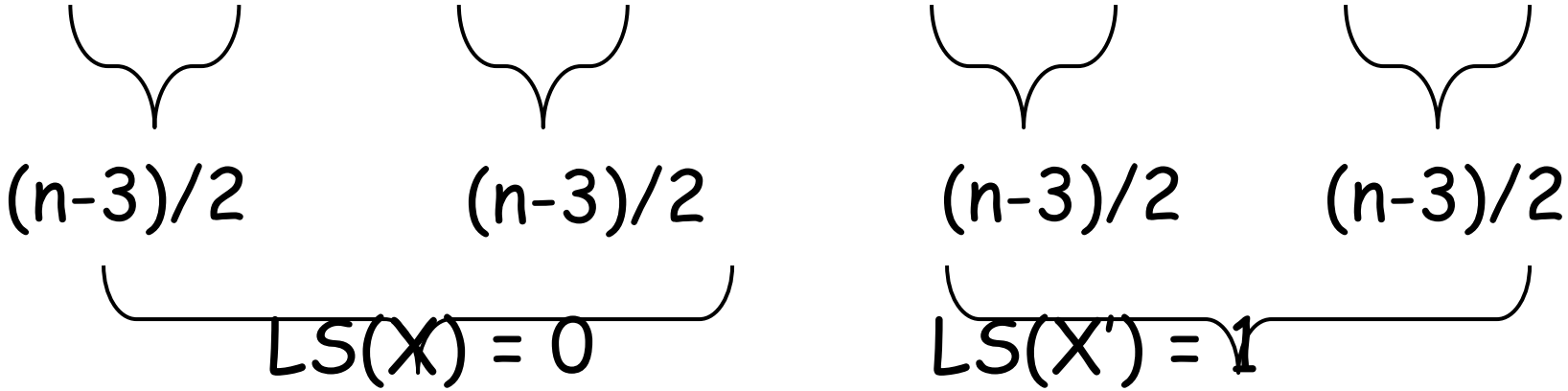
$$\Pr[f(D) + \text{Lap}(LS_f(D)/\epsilon) = 0] = 1 \qquad \Pr[f(D') + \text{Lap}(LS_f(D')/\epsilon) = 0] = 0.$$

- Solution: Smooth the upper bound of LS_f

Sensitivity of *Local* Sensitivity of **Median**

Median of $x_1, x_2, \ldots, x_n$ in [0,1]

- X= 0,…,0,0,0,0,1,…,1 X'= 0,…,0,0,0,1,1,…,1



Noise magnitude **must** be an **insensitive** function!

Smooth Upper Bound

Compute a "smoothed" version of local sensitivity

- Design **sensitivity function** $S(X)$

$S(X)$ is an Σ**-smooth** upper bound on $LS_f(X)$ if:

- **for all x:** $S(X) \geq LS_f(X)$
- **for all neighbors X** and **X':** $S(X) \leq e^{\Sigma} S(X')$

Theorem: if a response

$$A(x) = f(x) + Lap(S(x)/\varepsilon)$$

is given, then A is 2ε-differentially private.

Gives ε-DP about $f(x)$ and about $S(x)$

Smooth Sensitivity Mechanism

(Smooth Sensitivity) *For $\beta > 0$ the β-smooth sensitivity of f is:*

$$S^*_{f,\beta}(D) = \max_{D' \subset X} LS_f(D') \exp(-\beta d(D', D))$$

Theorem: Let S_f be an ϵ-smooth upper bound on f. Then an algorithm that output:

$$M(D) = f(D) + Lap(S_f(D)/\epsilon)$$

is 2ϵ-differentially private.

- Simple proof similar to original Laplace mechanism proof.

Sensitivity and Differential Privacy

- Differential Privacy
- Sensitivity:
  - **Global sensitivity** of query $q: U^n \rightarrow R^d$

$$GS_q = \max_{D,D'} ||q(D) - q(D')||_1$$

  - **Local sensitivity** of query q at point D

$$LS_q(D) = \max_{D'} |q(D) - q(D')|$$

  - **Smooth sensitivity**

$$S_f^*(X) = \max_y \{LS_f(Y) e^{-\sum \text{dist}(x,y)}\}$$

Advanced Results

accuracy vs privacy

Net Mechanism

Synthetic Database

Many (fractional) counting queries [Blum, Ligett, Roth'08]:

Given n-row database x, set Q of properties, produce a synthetic database y giving good approx to "What fraction of rows of x satisfy property P?" $\forall P \in Q$.

- S is set of all databases of size $m \in \tilde{O}(\log |Q|/\alpha^2) \ll n$
- $u(x, y) = -\max_{q \in Q} |q(x) - q(y)|$

- The size of m is the α-net cover number of D with respect to query class Q.

The Net Mechanism

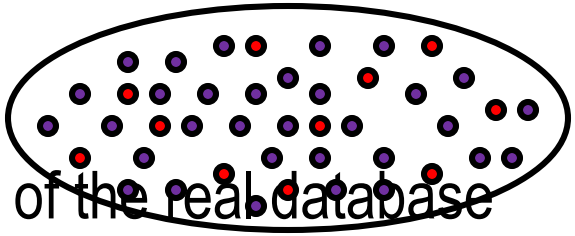
- Idea: limit the number of possible outputs of # databases
  - Want $|\mathcal{R}|$ to be small
- Why is it good?
  - The **good** (accurate) output has to compete with a few possible outputs
  - If there is a guarantee that there is at least one good output, then the total weight of the bad outputs is limited

⟨ Nets

A collection **N** of databases is called an ⟨ **-net** of databases for a ***class*** of queries ***C*** if:

- for all possible databases **x** there exists a **y** in **N** such that

$$Max_{q \in C} |q(x) - q(y)| <= \langle$$



If we use the **closest member** of **N** instead of the real database

lose at most ⟨

In terms of worst query

The Net Mechanism

For a class of queries C, privacy Σ and accuracy $\langle$, on data base x

- Let **N** be an $\langle$ **-net** for the class of queries C
- Let $w(x,y) = - \mathrm{Max}_{q \text{ in } C} |q(x) - q(y)|$
- Sample and output according to exponential mechanism with x, w, Σ and **R=N**
  - For y in **N:** Prob[y] proportional to $e^{\Sigma w(x,y)}$

$$\text{Prob}[y] = e^{\Sigma w(x,y)} / \square_{z \text{ in } N} e^{\Sigma w(x,z)}$$

Privacy and Utility

Claims:

Sensitivity of $w(x,y)$

Privacy: the net mechanism is $\Sigma\otimes$**differentially private**

Utility: the net mechanism is $(\langle +©, ®)$ **accurate** for any $\langle$, $©$ and $®$ such that

$$© > \log(|N|/®)/\Sigma$$

Proof:

- there is **at least** one **good** solution: gets weight **at least** $e^{-\Sigma\langle}$
- there are **at most** $|N|$ (bad) outputs: each get weight **at most** $e^{-\Sigma(\langle +©)}$
- Use the **Union Bound**

Accuracy less than $\langle +©$

$$|N|e^{-\Sigma(\langle +©)} \cdot ®e^{-\Sigma\langle}$$

Privacy and Accuracy?

(α, β)-usefulness of a private algorithm

- A mechanism M is (α, β)-useful with respect to queries in class C if for every database $D \in N^{|X|}$ with probability at least $1 - \beta$, the output

$$\max_{Q_i \in C} |Q_i(D) - M(Q_i, D)| \leq \alpha$$

- So it is to compare the private algorithm with non-private algorithm in PAC setting.
- A remark here: The tradeoff privacy may be absorbed in the inherent noisy measurement! Ideally, there can be no impact scale-wise!

Usefulness of Net Mechanism

- **Usefulness**

For any class of queries C the Net Mechanism is $(2\alpha, \beta)$-useful for any α

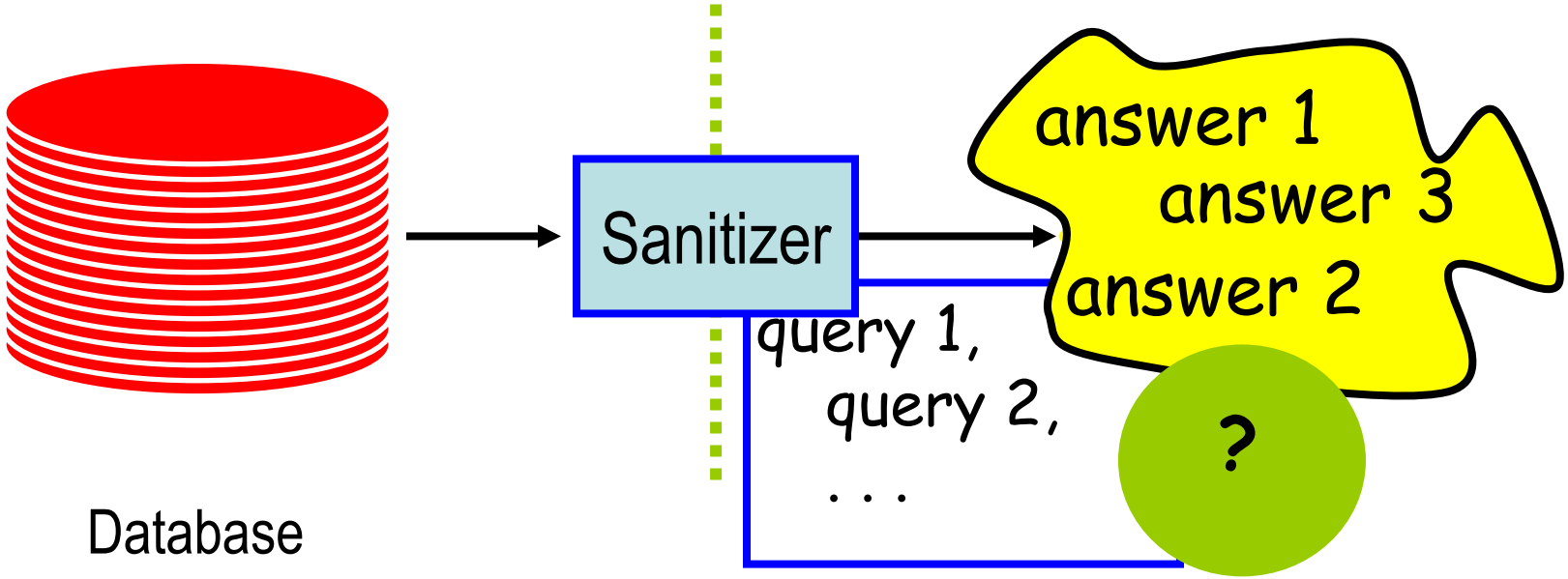
$$\alpha \geq \frac{2\Delta}{\epsilon} \log \frac{N_\alpha(C)}{\beta} \qquad \textit{Where } \Delta = \max_{Q \in C} GS(Q).$$

- **For counting queries** $|N_\alpha(C)| \leq |X|^{\frac{\log |C|}{\alpha^2}}$

- Logarithm to number of queries! Private to exponential number of queries!
- Well exceeds the fundamental limit of Dinur-Nissim03 for perturbation based privacy guarantee. (why?)

Synthetic Database

Synthetic DB: Output is a DB



Synthetic DB: output is always a DB

- Of entries from same universe **U**
- User reconstructs answers to queries by evaluating the query on output DB

Software and people compatible
Consistent answers

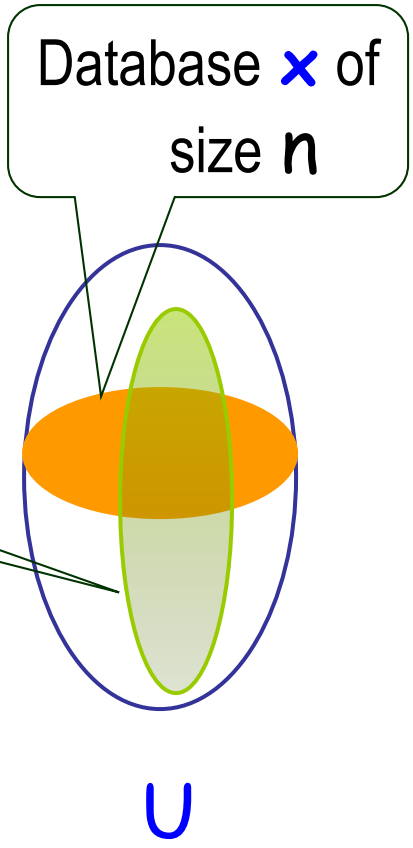
Counting Queries

- Queries with low sensitivity

Counting-queries

$\mathcal{C}$ is a set of predicates $q: U \to \{0,1\}$

Query: how many x participants satisfy q?



Relaxed accuracy:

- Answer query within α **additive error** w.h.p

Not so bad: ***error anyway inherent in statistical analysis***

Assume all queries given in advance

Non-interactive

$\langle$ -Net For Counting Queries

If we want to answer many counting queries C with differential privacy:

- Sufficient to come up with an $\langle$ -Net for C
- Resulting accuracy $\langle + \log(|N|/®)/\Sigma$

Claim: the set N consisting of **all** databases of **size** m where $m = \log|C|/\langle^2$

Consider each element in the set to have weight n/m

is an $\langle$ -Net for any collection C of counting queries

- Error is $\tilde{O}(n^{2/3}\log|C|)$

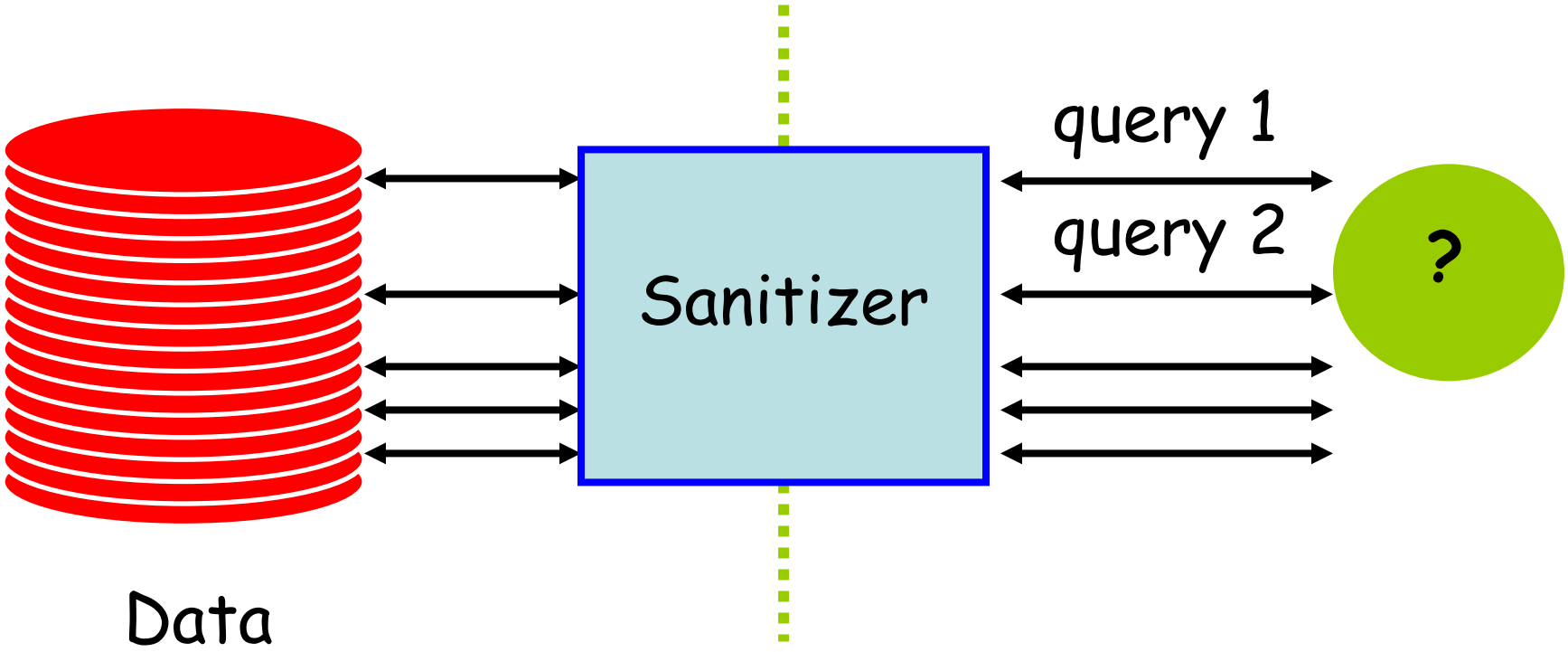
Remarkable

Hope for rich private analysis of small DBs!

- Quantitative: #queries >> DB size,

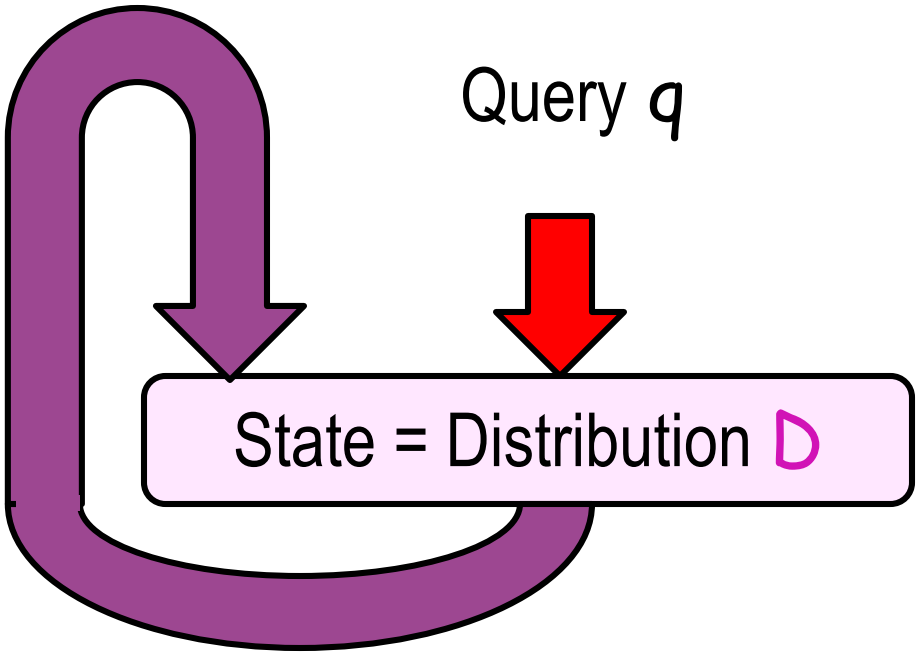
- Qualitative: output of sanitizer **-synthetic DB-**
  output is a DB itself

Interactive Model



Multiple queries, chosen adaptively

Maintaining State



Sequence of distributions $D_1, D_2, \ldots, D_t$

General structure

- Maintain public D_t (distribution, data structure)
- On query q_i:
  - try to answer according to D_t
  - If answer is not accurate enough:
    - Answer q_i using another mechanism
    - Update: D_{t+1} as a function of D_t and q_i

Lazy Round

Update Round

The Multiplicative Weights Algorithm

- Powerful tool in algorithms design
- Learn a Probability Distribution iteratively
- In each round:
  - either current distribution is good
  - or get a lot of information on distribution
    - **Update distribution**

The PMW Algorithm

Maintain a distribution D_t on universe U

This is the state. Is completely public!

- Initialize D_0 to be uniform on U

Repeat up to L times

Algorithm fails if more than L updates

- Set $\hat{T} \leftarrow T + Lap(\mu)$
- Repeat while no update occurs:
  - Receive query $q \in Q$
  - Let $\hat{a} = x(q) + Lap(\mu)$

The true value

  - **Test**: If $|q(D_t) - \hat{a}| \leq \hat{T}$: **output** $q(D_t)$.
  - **Else** (**update**):
    - **Output** $\hat{a}$
    - Update $D_{t+1}[i] \propto D_t[i]\ e^{\pm T/4q[i]}$ and re-weight.

the plus or minus are according to the sign of the error

New dist. is D_{t+1}

Overview: Privacy Analysis

For the query family $Q = \{0,1\}^U$ for $(®, ™, \sum)$ and $†$ the PMW **mechanism** is

- $(\sum, ™)$ **–differentially private**
- $(\langle, ®)$ **accurate** for up to $†$ queries where

$$\langle = \tilde{O}(1/(\sum n)^{1/2})$$

accuracy

Log dependency on $|U|$, $™$, $®$ and $†$

- State = Distribution is privacy preserving for individuals (but not for queries)

Epochs

- Epoch: the period between two updates

D_0 D_1 D_{t-1}

$q_1, q_2, \ldots, q_{\ell_1}, q_{\ell_1+1}, \ldots, q_{\ell_2}, \cdots q_{\ell_t+1}, \ldots, q_{\ell_{t+1}}, \cdots$

1st epoch 2nd epoch t^{th} epoch

The t^{th} epoch starts with distribution D_{t-1}

Queries $q_{\ell_t+1}, q_{\ell_t+2}, \ldots, q_{\ell_{t+1}-1}, q_{\ell_{t+1}}$

Lazy queries: response $q_j(D_t)$

update: response $\hat{a} = x(q) + \mathrm{Lap}(\mu)$

Epochs

- The t^{th} epoch starts with distribution D_{t-1}

  Queries $q_i, q_{i+1}, \ldots, q_{i+\ell-1}, q_{i+\ell}$

  Lazy queries: response $q_j(D_t)$

  update: response $\hat{a} = x(q) + \mathrm{Lap}(\mu)$

  For two inputs x and x', if:

  - agree on all responses up to q_i
  - agree that queries $q_i, q_{i+1}, \ldots, q_{i+\ell-1}$ are lazy:
  - agree that $q_{i+\ell}$ needs an update
  - agree on $\hat{a}$

  then agree on D_{t+1}

Epochs

- For two inputs x and x' for queries $q_i, q_{i+1}, \ldots, q_{i+\ell-1}$ suppose that the same random choices where made at step

$$\hat{a} = x(q) + \mathrm{Lap}(\mu)$$

Call the two sequences of choices

$$a_i, a_{i+1}, \ldots, a_{i+\ell-1}$$

$$a'_i, a'_{i+1}, \ldots, a'_{i+\ell-1}$$

The L_∞ difference is at most 2

The queries $q_i, q_{i+1}, \ldots, q_{i+\ell-1}$ are lazy in x iff

$$\max_{i \le j \le i+\ell} |a_j - q_j(D_{t-1})| \le \hat{T}$$

The queries $q_i, q_{i+1}, \ldots, q_{i+\ell-1}$ are lazy in x' iff

$$\max_{i \le j \le i+\ell} |a'_j - q_j(D_{t-1})| \le \hat{T}'$$

if $\hat{T}$ and $\hat{T}'$ are ± 2 of each other

Utility Analysis

Kullbeck Liebler Divergence

- **Potential function**

$$\Phi(t) = KL(x||D_t) = \sum_i x[i] \log\left(\frac{x[i]}{D_t[i]}\right)$$

- **Observation 1**: $\Phi(0) \leq \log|U|$ (initial distribution uniform)
- **Observation 2**: $\Phi(t) \geq 0$
  - non-negativity of Relative Entropy

- **Potential drop** in round t:

$$\Phi(t-1) - \Phi(t)$$

… Utility Analysis

- By the high concentration properties of the Laplacian mechanism,
  - with probability at least 1- ®all the noise added is of magnitude at most ⌈ log(†/®)

Set T >6 ⌈ log(t/®)and ⌈ > ⟨ $_0$. Suppose no such exception occurred.

- ®upper bound on the failure probability
- † – number of rounds

Setting the parameters

- **Maximize potential drop** $\eta T - \eta^2$
  - Decreases number of update rounds
- **Minimize threshold** T
  - Decreases noise in lazy rounds
- Setting $\eta = n^{-\frac{1}{2}} \log^{1/4} N$ and $T = 2\eta$
- Gives error $O(T)$

If an update step occurs, then

$$|q(D) - q(x)| > T - 2\sqrt{\log\{t/\delta\}} > T/2$$

The argument is based on the fact that each update reduces $KL(x||D)$ by $\wedge(T^2)$.

Since the initial value of $KL(x||D)$ is at most $\log |U|$, the maximum number of update is bounded by $O(\log|U|/T^2)$.

The bound L on the number of epochs, should to be this value.

Reading list

- Cynthia Dwork's video tutoial on DP
- [Cynthia 06] Differential Privacy (Invited talk at ICALP 2006)
- [Frank 09] Privacy Integrated Queries
- [Mohan et. al. 12] GUPT: Privacy Preserving Data Analysis Made Easy
- [Cynthia Dwork 09] The Differential Privacy Frontier