Building Privacy-Aware Database Systems

CS848 Winter 2021

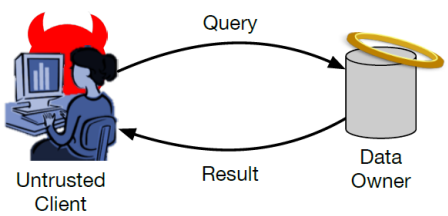
Module 1

UNIVERSITY OF WATERLOO | DSg Data Systems Group

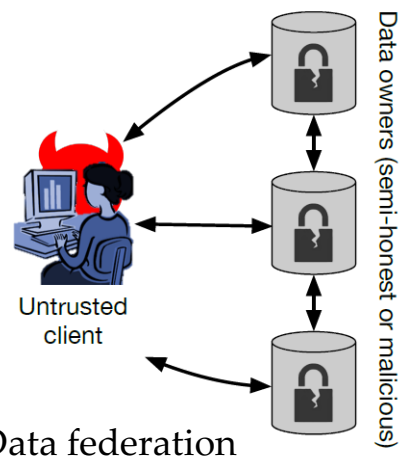
This course will explore …

- How to define a good privacy promise?
- How to design a privacy-preserving algorithms?
- How to build a privacy-aware database systems?

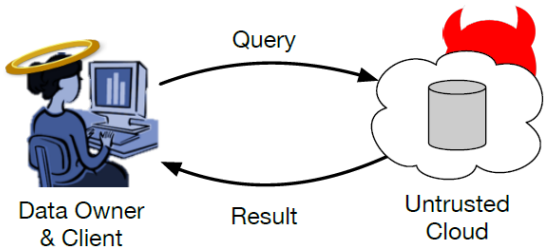
Greatly depend on
the architecture setup and trust assumptions



Client-server with trusted data curator



Data federation



Cloud service provider

Outline

- Part I: Differential Privacy Basics
- Part II: Implementation Challenges
- Upcoming Papers and Announcements

Desiderata for a Privacy Definition

1. Resilience to background knowledge
   - A privacy mechanism must be able to protect individuals' privacy from attackers who may possess background knowledge

2. Privacy without obscurity
   - Attacker must be assumed to know the algorithm used as well as all parameters [MK15]

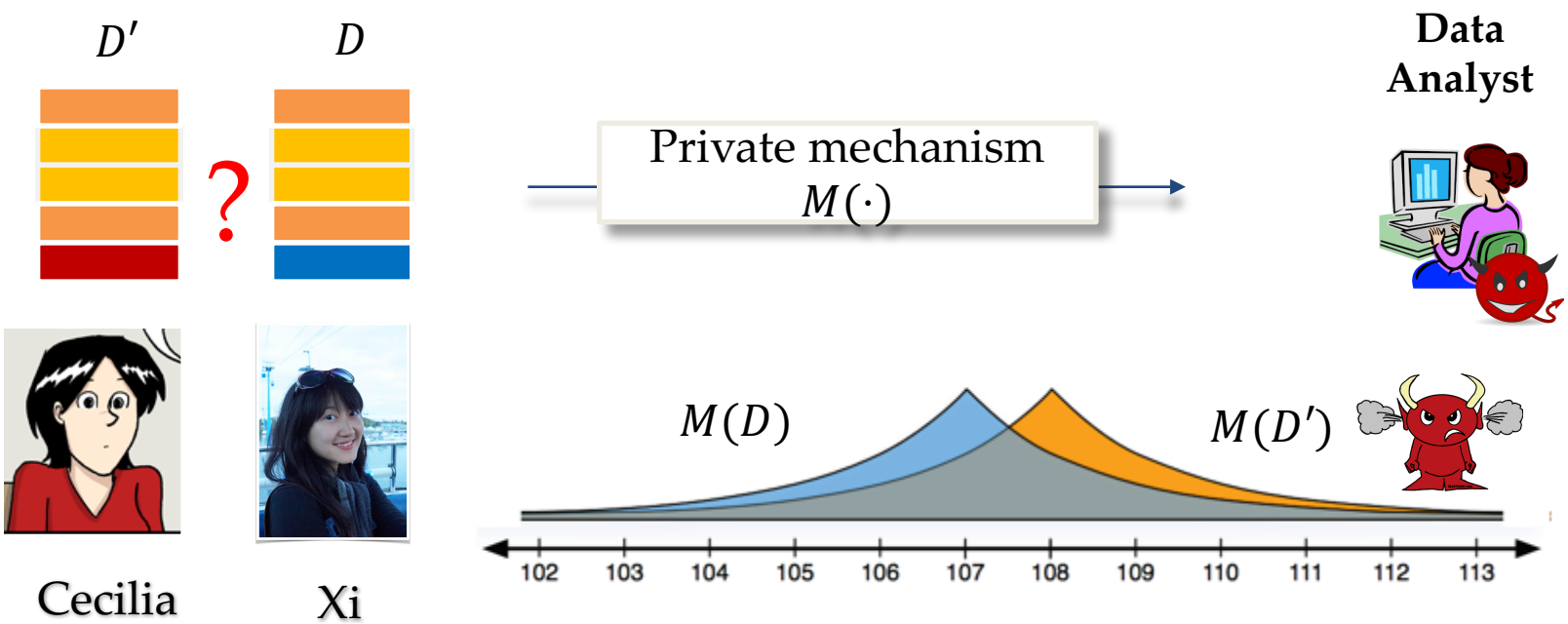
3. Post-processing
   - Post-processing the output of a privacy mechanism must not change the privacy guarantee [KL10, MK15]

4. Composition over multiple releases
   - Allow a graceful degradation of privacy with multiple invocations on the same data [DN03, GKS08]

Differential Privacy

[Dwork06]

- *"An algorithm satisfies differential privacy (DP) if its output is insensitive to adding, removing or changing one record in its input database"*

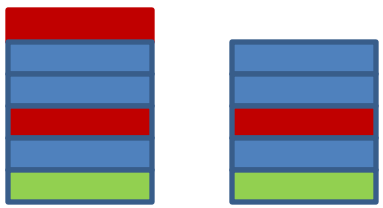


Differential Privacy

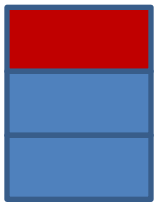
[Dwork ICALP 2006]

For every pair of inputs that differ in one row

For every output …



D_1 D_2



O

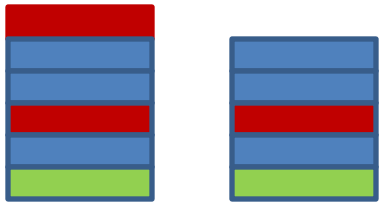
Adversary should not be able to distinguish between any D_1 and D_2 based on any O

$$\ln\left(\frac{\Pr[A(D_1) = o]}{\Pr[A(D_2) = o]}\right) \leq \varepsilon, \qquad \varepsilon > 0$$

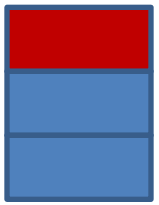
Why pairs of datasets *that differ in one row*?

For every pair of inputs that differ in one row

For every output …



D_1 D_2



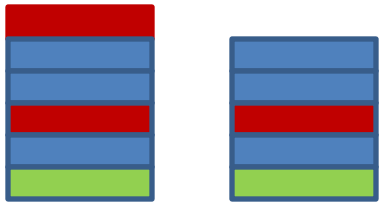
O

Simulate the presence or absence of a single record

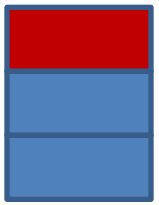
Why *all* pairs of datasets …?

For every pair of inputs that differ in one row

For every output …



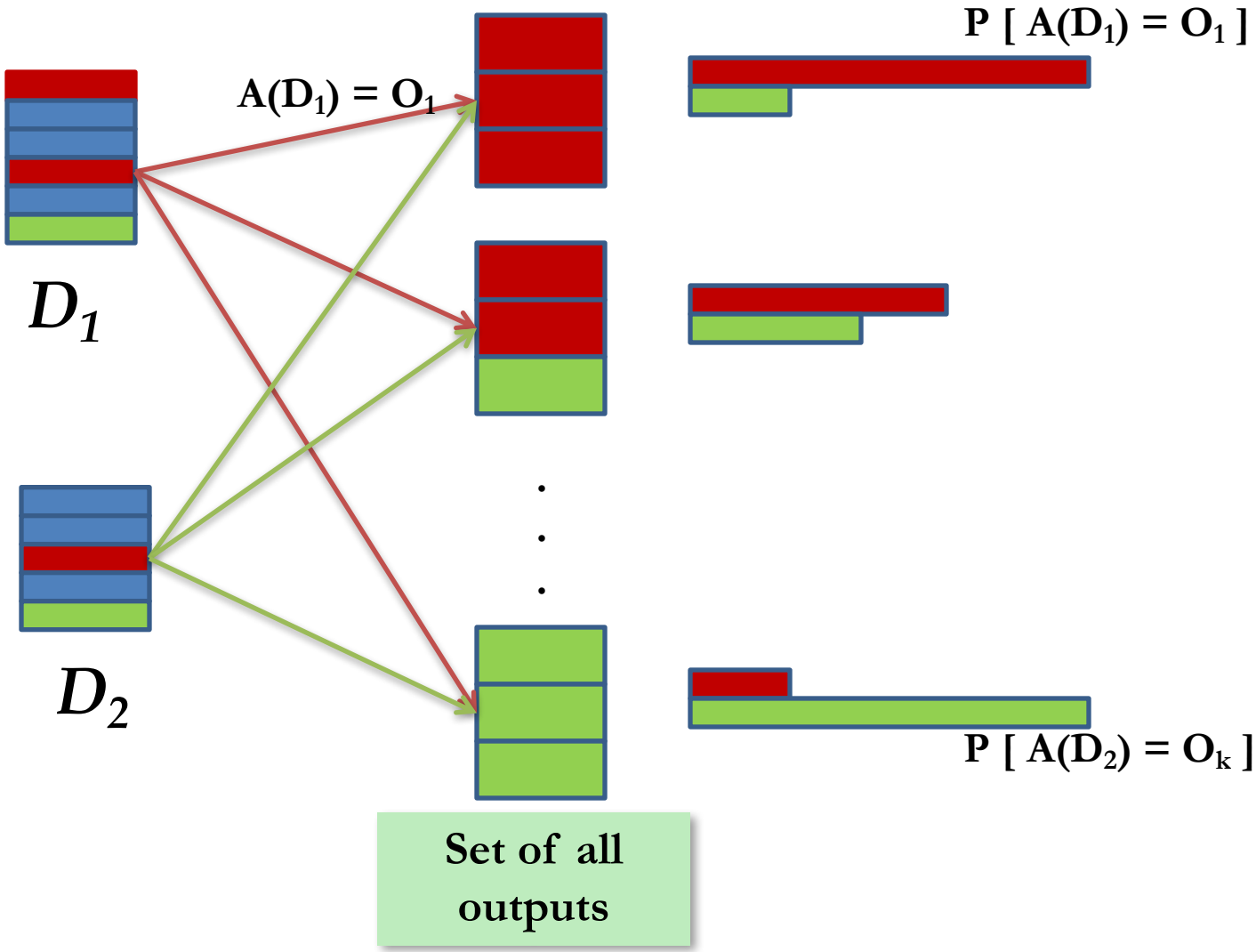
D_1 D_2



O

Guarantee holds no matter what the other records are.

Why *all* outputs?

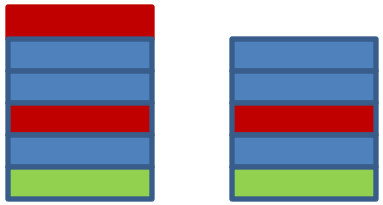


Should not be able to distinguish whether input was D_1 or D_2 no matter what the output

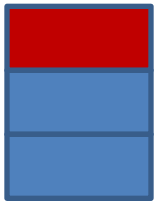
Privacy Parameter ε

For every pair of inputs that differ in one row

For every output …



D_1 D_2



O

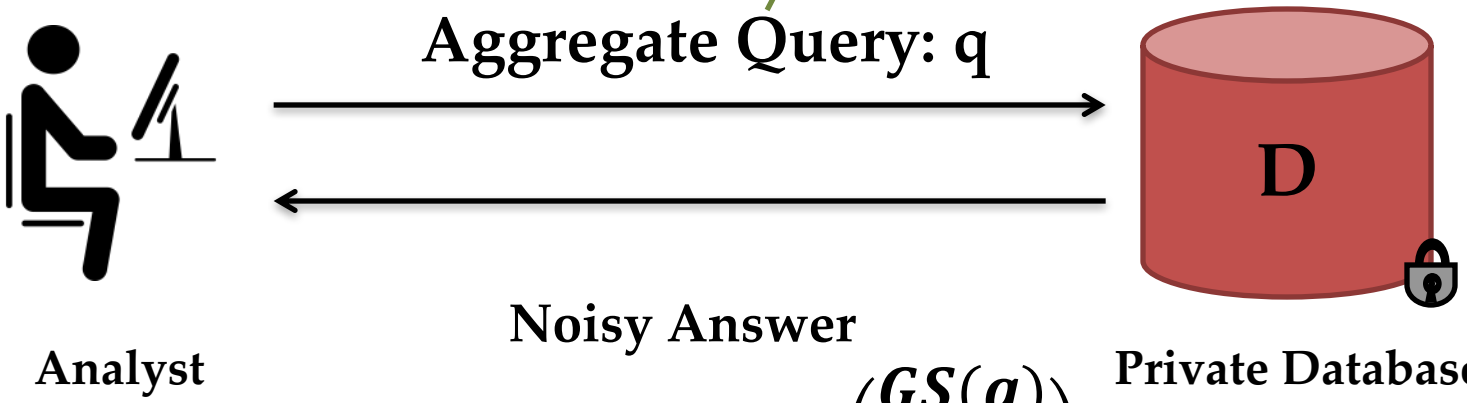
$$\Pr[A(D_1) = o] \leq e^{\varepsilon} \Pr[A(D_2) = o]$$

Controls the degree to which D_1 and D_2 can be distinguished.
Smaller the ε more the privacy (and worse the utility)

[DMNS 06]

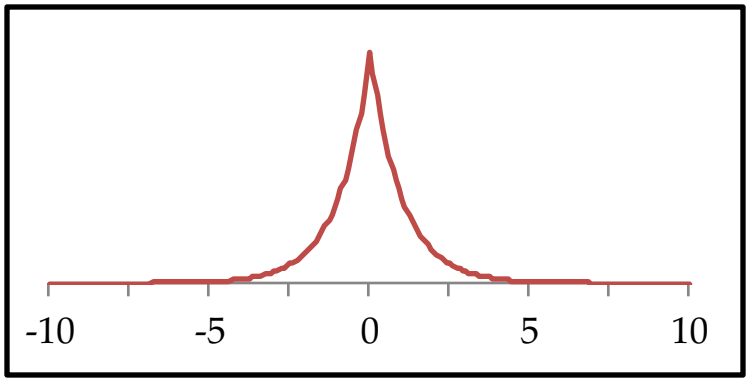
Laplace Mechanism

e.g., COUNT



$$\tilde{q}(D) = q(D) + \mathbf{Lap}\left(\frac{GS(q)}{\varepsilon}\right)$$

Global Sensitivity



$$\mathbf{Lap}(\lambda)\text{: } h(\eta) \propto \exp\left(-\frac{|\eta|}{\lambda}\right)$$

How much noise for privacy?

Global Sensitivity of a query q that outputs a real number:
the maximum change to the query output,
for any neighboring tables D_1, D_2 that differ in a row,

$$GS(q) = \max_{\forall\, neighbor(D_1, D_2)} |q(D_1) - q(D_2)|$$

$$= \max_{D_2 \in dom} \; \max_{\forall\, D_1 \in neighbors(D_2)} |q(D_1) - q(D_2)|$$

Any possible database D_2

Add/remove any record from D_2

Theorem: $\boldsymbol{q(D)} + \mathbf{Lap}\left(\frac{\boldsymbol{GS(q)}}{\boldsymbol{\varepsilon}}\right)$ satisfies $\boldsymbol{\varepsilon}$-DP.

Global Sensitivity: COUNT query

- # of people having flu?
- Sensitivity = 1

D

Sex	Height	Age	Disease	Drug X
M	6′2″	56	Cancer	3.5
F	5′3″	30	Diabetes	2.3
F	5′9″	24	Healthy	1.0
M	5′3″	36	Flu	4.0
M	6′7″	22	Flu	2.2

- Solution: $2 + \eta$
  where η is drawn from $Lap(\frac{1}{\epsilon})$
  - Mean = 0
  - Variance = $2/\varepsilon^2$

Global Sensitivity: SUM query

- Total usage of drug X?

D

Sex	Height	Age	Disease	Drug X
M	6′2″	56	Cancer	3.5
F	5′3″	30	Diabetes	2.3
F	5′9″	24	Healthy	1.0
M	5′3″	36	Flu	4.0
M	6′7″	22	Flu	2.2

- Suppose all values x are in [a,b]

- Sensitivity = b

Privacy of Laplace Mechanism

- Consider neighboring databases D and D'
- Consider some output O

$$\frac{\Pr\left[A(D) = O\right]}{\Pr\left[A(D') = O\right]} = \frac{\Pr\left[q(D) + \eta = O\right]}{\Pr\left[q(D') + \eta' = O\right]}$$

$$= \frac{\Pr[\eta = O - q(D)]}{\Pr[\eta' = O - q(D')]}$$

$$= \frac{e^{-|O-q(D)|/\lambda}}{e^{-|O-q(D')|/\lambda}}$$

$$\leq e^{|q(D)-q(D')|/\lambda} \leq e^{GS(q)/\lambda} = e^{\varepsilon}$$

$\mathbf{Lap}(\boldsymbol{\lambda})$: $\mathrm{h}(\eta) \propto e^{-|\eta|/\lambda}$

$\lambda = GS(q)/\epsilon$

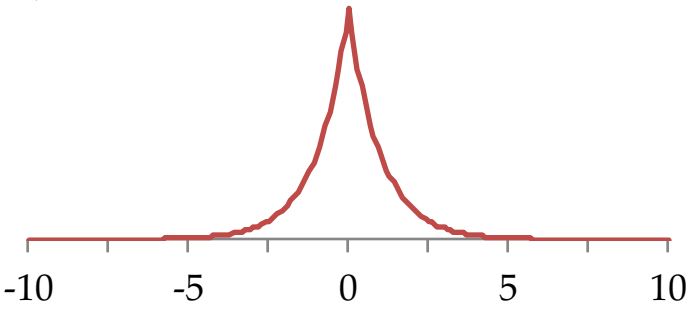
Utility of Laplace Mechanism

- Laplace mechanism works for **any function** that returns a real number

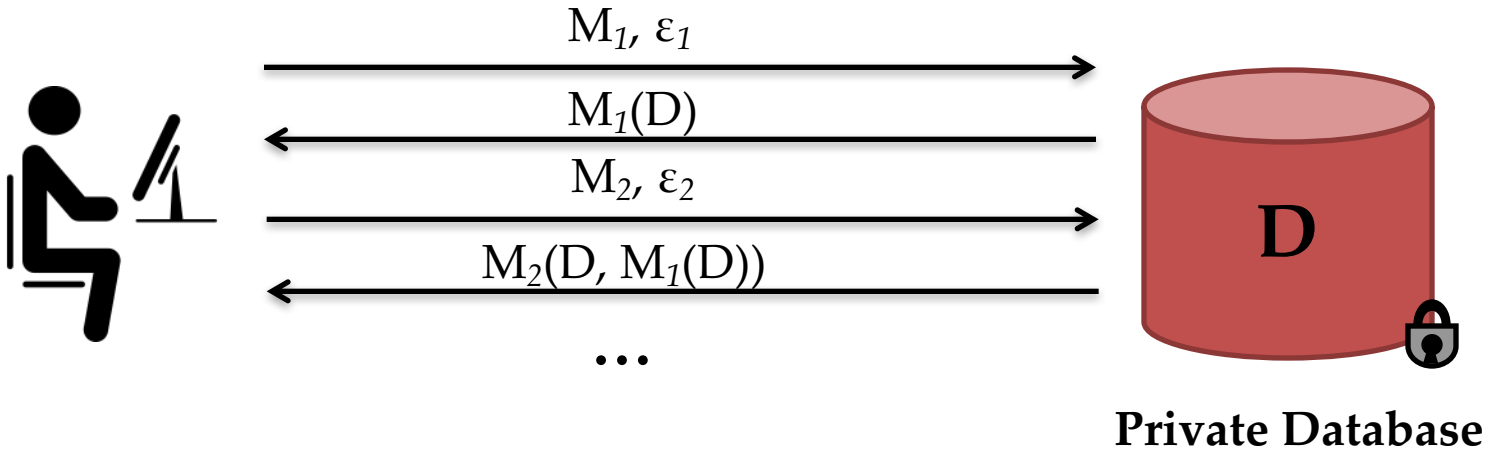
- Error: E(true answer – noisy answer)^2

$$= \mathrm{Var}(\ \mathrm{Lap}(GS(q)/\varepsilon)\ )$$

$$= 2 * GS(q)^2 / \varepsilon^2$$



Sequential Composition

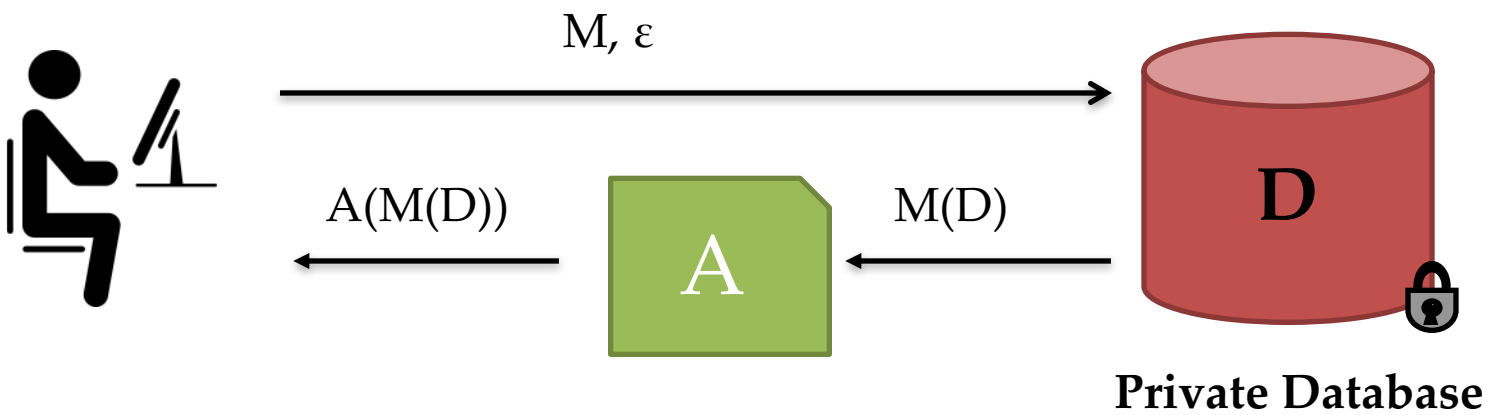


- If $M_1, M_2, \ldots, M_k$ are algorithms that access a private database D such that each M_i satisfies ε_i -differential privacy,

  then the combination of their outputs satisfies ε-differential privacy with

$$\varepsilon = \varepsilon_1 + \ldots + \varepsilon_k$$

Postprocessing



- If M is an ε-differentially private algorithm, any additional post-processing $A \circ M$ also satisfies ε-differential privacy.

Outline

- Part I: Differential Privacy Basics
  - Laplace mechanism
  - **Global sensitivity analysis**
- Part II: Implementation Challenges
- Upcoming Papers and Announcements

Global Sensitivity: Vector of Counts

- q: # of males and # of females with flu?
  - Return $\begin{pmatrix} c_m \\ c_f \end{pmatrix} + \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}$, where $\eta_i \sim Lap\left(\frac{GS(q)}{\epsilon}\right)$

$$GS(q) = \max_{\forall\, neighbor(D_1, D_2)} ||q(D_1) - q(D_2)||_1$$

- Sensitivity = ?

$$\max\left(\left|\begin{pmatrix} +1 \\ 0 \end{pmatrix}\right|_1, \left|\begin{pmatrix} -1 \\ 0 \end{pmatrix}\right|_1, \left|\begin{pmatrix} 0 \\ +1 \end{pmatrix}\right|_1, \left|\begin{pmatrix} 0 \\ -1 \end{pmatrix}\right|_1\right)$$

D

Sex	Height	Age	Disease	Drug X
M	6′2″	56	Cancer	3.5
F	5′3″	30	Diabetes	2.3
F	5′9″	24	Healthy	1.0
M	5′3″	36	Flu	4.0
M	6′7″	22	Flu	2.2

Global Sensitivity: Complex Queries

- # of (distinct) diseases?
- # of rare diseases (appear < 5 times) and # of common diseases (appear >= 5 times)?
- …

How to automate this process for a large class of queries?

D

Sex	Height	Age	Disease	Drug X
M	6′2″	56	Cancer	3.5
	5′3″	30	Diabetes	2.3
	5′9″	24	Healthy	1.0
	5′3″	36	Flu	4.0
M	6′7″	22	Flu	2.2

Transformations & Stability

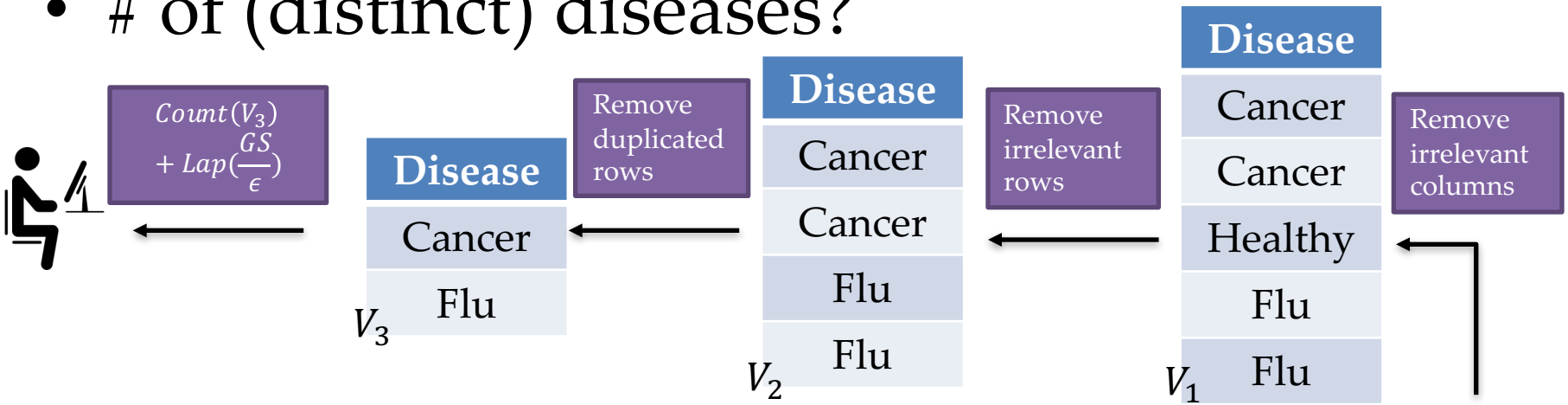
- Express a counting query as a sequence of pre-defined transformations
  - $q(D)$: $V_m \ldots ((V_2(V_1(D))))$

- Track the **stability of each transformation** s_V:
  - Maximum number of rows in V that can change due to changing a single row in the input table

- Bound global sensitivity $GS(q) \leq s_{V_m} \ldots s_{V_2} s_{V_1}$

Transformations & Stability

- # of (distinct) diseases?

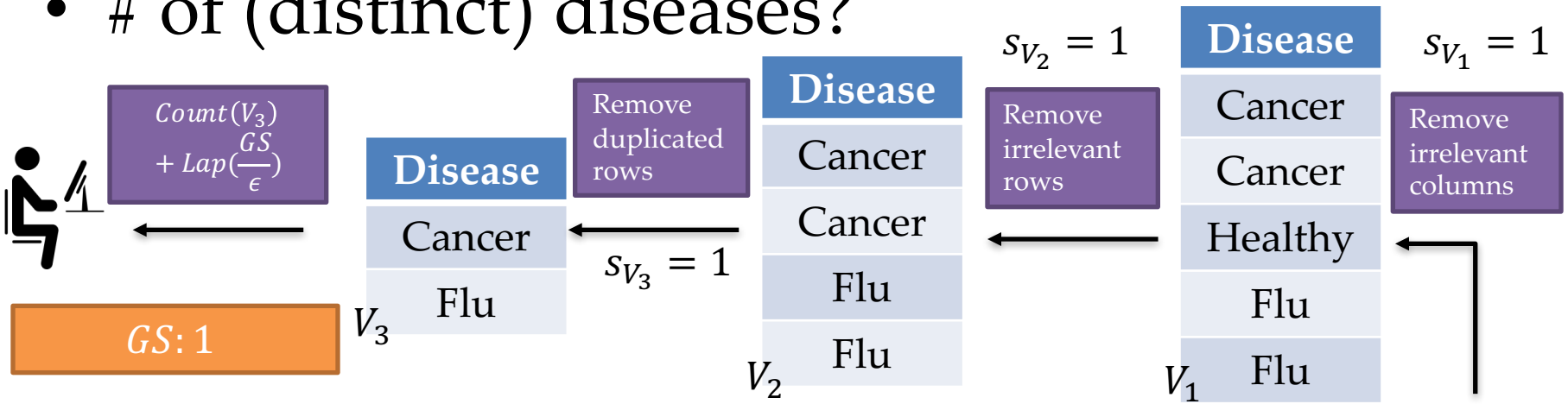


Sex	Height	Age	Disease	Drug X
M	6′2″	56	Cancer	3.5
F	5′3″	30	Cancer	2.3
F	5′9″	24	Healthy	1.0
M	5′3″	36	Flu	4.0
M	6′7″	22	Flu	2.2

D

Transformations & Stability

- # of (distinct) diseases?



Sex	Height	Age	Disease	Drug X
M	6′2″	56	Cancer	3.5
F	5′3″	30	Cancer	2.3
F	5′9″	24	Healthy	1.0
M	5′3″	36	Flu	4.0
M	6′7″	22	Flu	2.2

D

Transformations & Stability

- Frequency of the word "covid"

$s_{V_2} = 1$

Remove irrelevant rows and columns

Tweet id	Word
Id 1	"privacy"
Id 1	"covid"
Id 1	…
Id 2	"mask"
…	…
Id 5	"risk"
Id 5	…

V_1

Word
"covid"
"covid"
…

V_2

GS: unbounded

$Count(V_2) + Lap(\frac{GS}{\epsilon})$

Unbounded!!

$s_{V_1} = ?$

Split rows

t id	Words
Id 1	{"privacy", "covid",…}
Id 2	{"mask", "covid",…}
Id 3	{"covid", "covid",…}
Id 4	{"privacy", "risk" …}
Id 5	{"water","plant"…}

D

Transformations & Stability

- Frequency of the word "covid"

$s_{V_2} = 1$

Remove irrelevant rows and columns

Tweet id	Word
Id 1	"privacy"
Id 1	"covid"
Id 1	…
Id 2	"mask"
…	…
Id 5	"risk"
Id 5	…

V_1

Word
"covid"
"covid"
…

V_2

$GS: 1 \cdot k = k$

$Count(V_2) + Lap(\frac{GS}{\epsilon})$

$s_{V_1} = ?$

Split

$s_{V_1} = k$

Take first k words per row

Tweet id	Words
Id 1	{"privacy", "covid",…}
Id 2	{"mask", "covid",…}
Id 3	{"covid", "covid",…}
Id 4	{"privacy", "risk" …}
Id 5	{"water","plant"…}

D

Outline

- Part I: Differential Privacy Basics
  - Laplace mechanism
  - Global sensitivity analysis
  - **PINQ: Implementation of DP**
- Part II: Implementation Challenges
- Upcoming Papers and Announcements

[McSherry SIGMOD'09]

PINQ (Privacy Integrated Queries)

- Implementation is based on C#'s LINQ language

Example 1 Counting searches from distinct users in PINQ.

```
var data = new PINQueryable<SearchRecord>( ... ... );

var users = from record in data
            where record.Query == argv[0]
            groupby record.IPAddress

Console.WriteLine(argv[0] + ": " + users.NoisyCount(0.1));
```

Transformations

Example 3 [Abbreviated] Implementation of NoisyCount.

```
double NoisyCount(double epsilon)
{
    if (myagent.Alert(epsilon))
        return mysource.Count() + Laplace(1.0/epsilon);
    else
        throw new Exception("Access is denied");
}
```

Track privacy budget

PINQAgent

- Keeps track of privacy budget

Example 2 Implementing a fixed budget in a PINQAgent.

```
public class PINQAgentBudget : PINQAgent
{
    private double budget;

    public override bool Alert(double epsilon)
    {
        if (budget < epsilon)
            return false;

        budget = budget - epsilon;
        return true;
    }

    public PINQAgentBudget(double b)  { budget = b; }
}
```

PINQ: Aggregation Operators

- Laplace Mechanism
  - NoisyCount
  - NoisySum
- Exponential Mechanism
  - NoisyMedian
  - NoisyAverage

PINQ: Transformation Operators

- Aggregations are computed on transformed data
  - ***Where***: takes as input a predicate (arbitrary C# function), and outputs a subset of the data satisfying the predicate Stability = 1
  - ***Select***: Maps each input record into a different record using a C# function Stability = 1
  - ***GroupBy***: Groups records by key values Stability =?
  - ***Join***: Takes two datasets, and key values for each and returns groups of pairs of records for each key.

Join Operator

- **SELECT COUNT(*) FROM** A **JOIN** B ON A.k = B.k

k	v
1	a

A

k
1
1

B

k	v
1	a
1	a

A JOIN B

k	v
1	a
1	b

A

k
1
1

B

k	v
1	a
1	a
1	b
1	b

A JOIN B

Add a row to Table A

+2 in join output size for this given instance

Consider All Possible DB Pairs

- **SELECT COUNT(*) FROM** A **JOIN** B ON A.k = B.k

k	v
1	a

A

k
1
..
..

B

K	v
1	a
..	..
..	..

A JOIN B

k	v
1	a
1	b

A

k
1
..
..

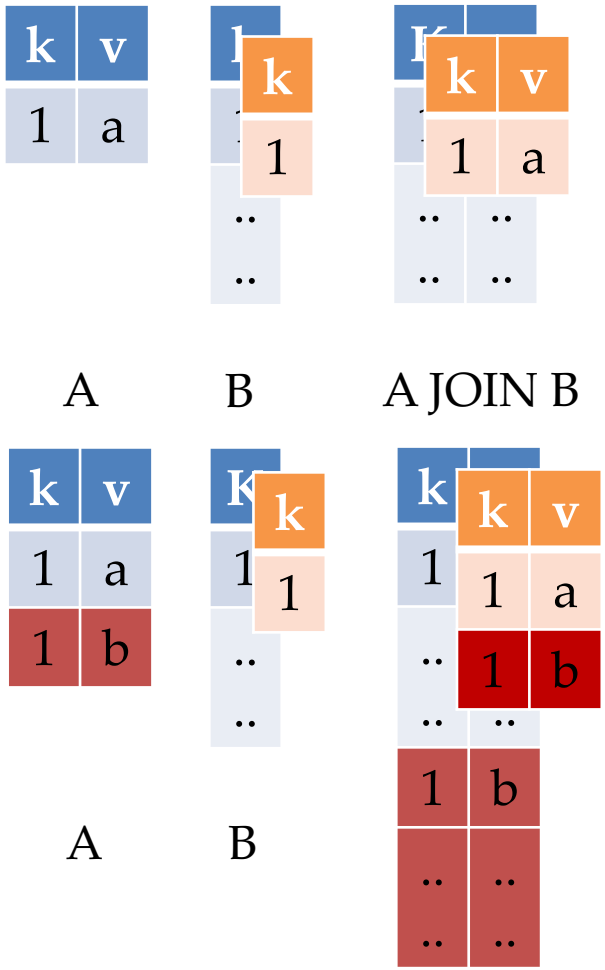
B

k	v
1	a
..	..
..	..
1	b
..	..
..	..

Unbounded change in join output size

Special "Join" in PINQ

- **SELECT COUNT(*) FROM** A **JOIN** B ON A.k = B.k



Keep one copy for each key value

Limited change to join output size

Limited queries or poor utility !!!

Outline

- Part I: Differential Privacy Basics
  - Laplace mechanism
  - Global sensitivity analysis
  - PINQ: Implementation of DP
- Part II: Implementation Challenges
  - **Highly sensitive queries (Joins)**
  - Attacks on DP Implementations
- Upcoming Papers and Announcements

Local Sensitivity of True Database

- **SELECT COUNT(*) FROM** A **JOIN** B ON A.k = B.k

k	v
1	a

A

k
1
1

B

k	v
1	a
1	a

A JOIN B

Can we just add noise proportional to LS?

No

k	v
1	a
1	b

A

K
1
1

B

k	v
1	a
1	a
1	b
1	b

A JOIN B

Add a row to Table A

+2 in join output size for this given instance

Local sensitivity = 2

Local Sensitivity v.s. Global Sensitivity

- **Local sensitivity**:
  - max change in query output when adding/removing a row from the true database instance D

$$LS(q,D) = \max_{\forall D_1 \in neighbors(D)} |q(D) - q(D_1)|$$

Add/remove any record from D

- **Global sensitivity**:
  - independent of the true database instance

$$GS(q) = \max_{D_2 \in dom} \max_{\forall D_1 \in neighbors(D_2)} |q(D_1) - q(D_2)|$$

Any possible database D_2

Add/remove any record from D_2

Local Sensitivity v.s. Global Sensitivity

- **Local sensitivity**:
  - max change in query output when adding/removing a row from the true database instance D

$$LS(q,D) = \max_{\forall D_1 \in neighbors(D)} |q(D) - q(D_1)|$$

Add/remove any record from D

- **Global sensitivity**:
  - independent of the true database instance

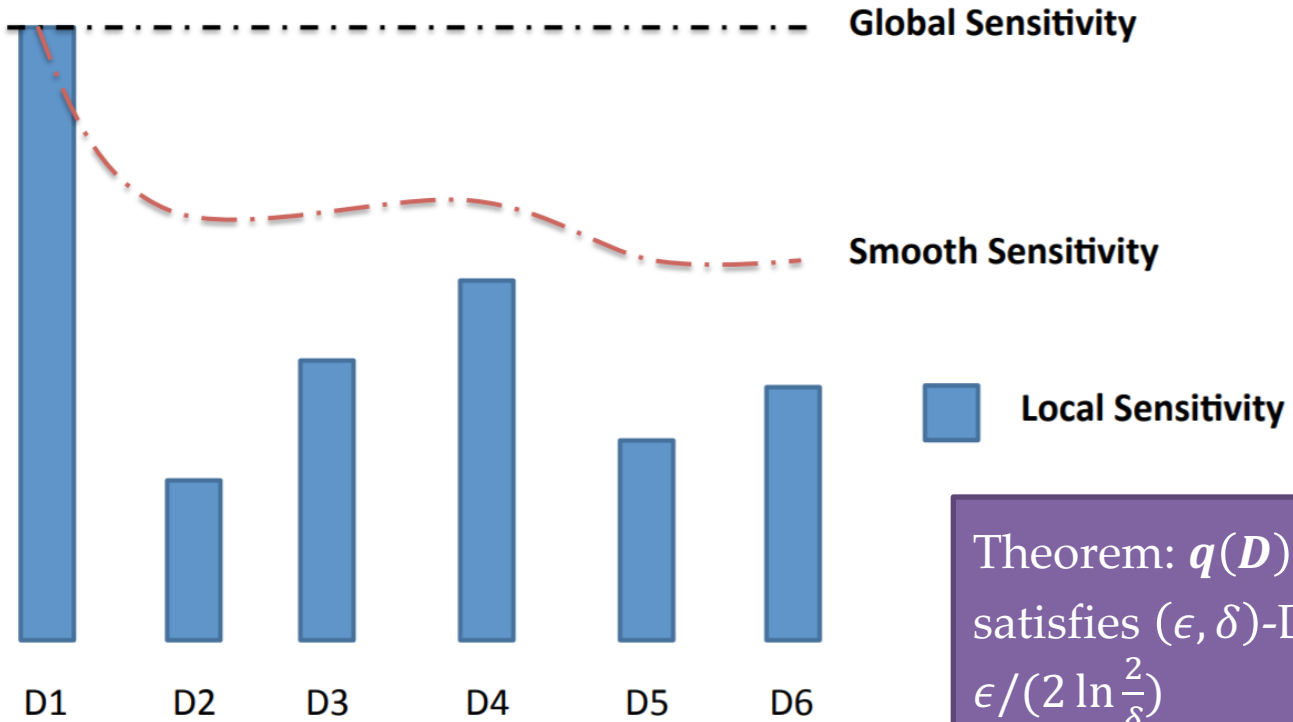
$$GS(q) = \max_{D_i \in dom} LS(q, D_i)$$

[Nissim et al., STOC 2007]

Smooth Sensitivity (SS)

$$SS(q,D) = \max_{D_i \in dom} e^{-\beta \cdot d(D_i,D)} LS(q, D_i)$$

Decay with the distance from true database



Theorem: $\boldsymbol{q}(\boldsymbol{D}) + Lap(\frac{2SS(q,D)}{\epsilon})$ satisfies (ϵ, δ)-DP, where $\beta = \epsilon/(2\ln\frac{2}{\delta})$

[FLEX, JNS, VLDB18]

Elastic Sensitivity (ES)

- Computing LS and SS is computationally expensive

- Elastic sensitivity: loose upper bound of local sensitivity instead of exact local sensitivity [FLEX]

$$ES(q,D) = \max_{D_i \in dom} e^{-\beta \cdot d(D_i,D)} \; LS'(q, D_i)$$

Upper bound of $LS(q, D_i)$

  – Recursively compute upper bound of $LS(q, D_i)$ for D_i at distance d from the true database

$$ES(q,D) = \max_{d=0,1,\ldots} e^{-\beta \cdot d} \max_{dist(D,D_i)=d} LS'(q, D_i)$$

Elastic Sensitivity (ES)

- Express each query as query plan (relational algebra)
- Recursive computation of statistics
  - max frequency and elastic stability

Core relational algebra:

Attribute names

a

Value constants

v

Relational transformations

$$R ::= t \mid R_1 \underset{x=y}{\bowtie} R_2 \mid \Pi_{a_1,\ldots,a_n} R \mid \sigma_\varphi R \mid \mathit{Count}(R)$$

Selection predicates

$$\varphi ::= a_1 \theta a_2 \mid a \theta v$$
$$\theta ::= < \mid \leq \mid = \mid \neq \mid \geq \mid >$$

Counting queries

$$Q ::= \mathit{Count}(R) \mid \underset{G_1..G_n}{\mathit{Count}}(R)$$

(a)

Definition of elastic stability:

$$\hat{S}_R^{(k)} :: R \to D^n \to \textit{elastic stability}$$
$$\hat{S}_R^{(k)}(t, x) = 1$$
$$\hat{S}_R^{(k)}(r_1 \underset{a=b}{\bowtie} r_2, x) = \begin{cases} \max(\mathsf{mf}_k(a, r_1, x)\hat{S}_R^{(k)}(r_2, x), \\ \quad \mathsf{mf}_k(b, r_2, x)\hat{S}_R^{(k)}(r_1, x)) & |\mathcal{A}(r_1) \cap \mathcal{A}(r_2)| = 0 \\ \mathsf{mf}_k(a, r_1, x)\hat{S}_R^{(k)}(r_2, x) + \\ \quad \mathsf{mf}_k(b, r_2, x)\hat{S}_R^{(k)}(r_1, x) + \\ \quad \hat{S}_R^{(k)}(r_1, x)\hat{S}_R^{(k)}(r_2, x) & |\mathcal{A}(r_1) \cap \mathcal{A}(r_2)| > 0 \end{cases}$$
$$\hat{S}_R^{(k)}(\Pi_{a_1,\ldots,a_n} r, x) = \hat{S}_R^{(k)}(r, x)$$
$$\hat{S}_R^{(k)}(\sigma_\varphi r, x) = \hat{S}_R^{(k)}(r, x)$$
$$\hat{S}_R^{(k)}(\mathit{Count}(r)) = 1$$

Definition of elastic sensitivity:

$$\hat{S}^{(k)} :: Q \to D^n \to \textit{elastic sensitivity}$$
$$\hat{S}^{(k)}(\mathit{Count}(r), x) = \hat{S}_R^{(k)}(r, x)$$
$$\hat{S}^{(k)}(\underset{G_1..G_n}{\mathit{Count}}(r), x) = 2\hat{S}_R^{(k)}(r, x)$$

(b)

Maximum frequency at distance k:

$$\mathsf{mf}_k :: a \to R \to D^n \to \mathbb{N}$$
$$\mathsf{mf}_k(a, t, x) = \mathsf{mf}(a, t, x) + k$$
$$\mathsf{mf}_k(a_1, r_1 \underset{a_2=a_3}{\bowtie} r_2, x) = \begin{cases} \mathsf{mf}_k(a_1, r_1, x)\mathsf{mf}_k(a_3, r_2, x) & a_1 \in r_1 \\ \mathsf{mf}_k(a_1, r_2, x)\mathsf{mf}_k(a_2, r_1, x) & a_1 \in r_2 \end{cases}$$
$$\mathsf{mf}_k(a, \Pi_{a_1,\ldots,a_n} r, x) = \mathsf{mf}_k(a, r, x)$$
$$\mathsf{mf}_k(a, \sigma_\varphi r, x) = \mathsf{mf}_k(a, r, x)$$
$$\mathsf{mf}_k(a, \mathit{Count}(r), x) = \bot$$

(c)

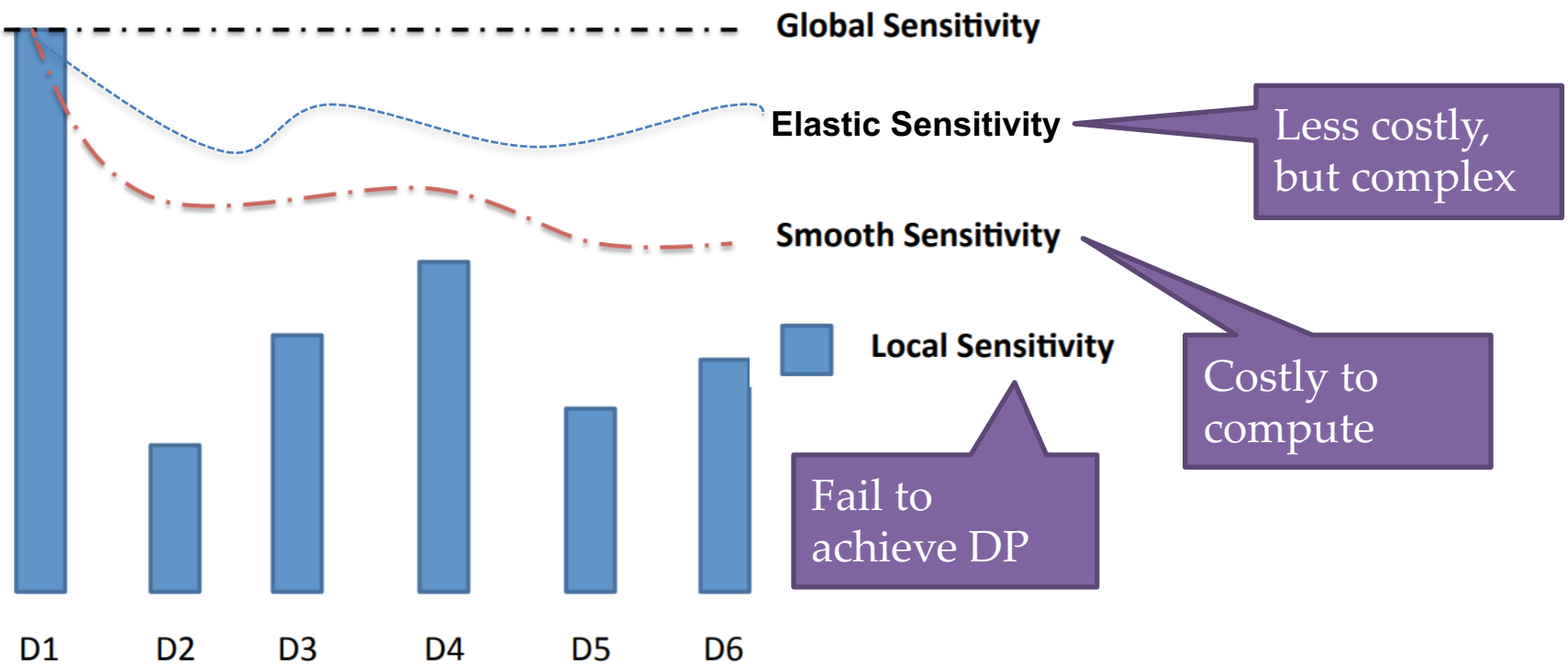
Ancestors of a relation:

$$\mathcal{A} :: R \to \{R\}$$
$$\mathcal{A}(t) = \{t\}$$
$$\mathcal{A}(r_1 \underset{a=b}{\bowtie} r_2) = \mathcal{A}(r_1) \cup \mathcal{A}(r_2)$$
$$\mathcal{A}(\Pi_{a_1,\ldots,a_n} r) = \mathcal{A}(r)$$
$$\mathcal{A}(\sigma_\varphi r) = \mathcal{A}(r)$$

(d)

Figure 1: (a) syntax of core relational algebra; (b) definition of elastic stability and elastic sensitivity at distance k; (c) definition of maximum frequency at distance k; (d) definition of ancestors of a relation.

Comparisons



At worst case, the local sensitivity of the true database instance is as large as the global sensitivity, all these approaches need to add a large amount of noise!

Other Approaches

- Apply a transformation V() over the database instance D, such that
  - the sensitivity of $q(V(D))$ is small, and
  - $q(V(D))$ is close to $q(D)$
  - E.g. transformation example on slide 23, slide 35

- Prior work
  - Sample & Aggregate (GUPT, Mohan et al. SIGMOD'12)
  - Truncation (PrivateSQL, Kotsogiannis et al. VLDB 2019)

Outline

- Part I: Differential Privacy Basics
  - Laplace mechanism
  - Global sensitivity analysis
  - PINQ: Implementation of DP
- Part II: Implementation Challenges
  - Highly sensitive queries (Joins)
  - **Attacks on DP Implementations**
- Upcoming Papers and Announcements

[HPN SEC11]

Covert Channel

- Key assumption in DP implementations: **The querier can *only* observe the result of the query, and nothing else.**
  - This answer is guaranteed to be differentially private.

- In practice: The querier can observe other effects.
  - E.g.Time taken by the query to complete, power consumption, etc.
  - Suppose a system takes 1 minute to answer a query if Bob has cancer and 1 microsecond otherwise, then based on query time the adversary may know that Bob has cancer.

[HPN SEC11]

Threat Model

- Assume the adversary (querier) does not have physical access to the machine.
  - Poses queries over a network connection.
- Given a query, the adversary can observe:
  - **Answer to their question**
  - **Time that the response arrives at their end of the connection**
  - **The system's decision to execute the query or deny (since the new query would exceed the privacy budget)**

[HPN SEC11]

Timing Attack

```
Function is_f(Record r){
    if(r.name = Bob && r. disease = Cancer)
        sleep(10 sec);          // or go into infinite loop, or throw exception
    return f(r);
}

Function countf(){
    var  fs = from record in data
              where (is_f(record))
    print fs.NoisyCount(0.1);
}
```

If Bob has Cancer, then the query takes > 10 seconds; otherwise, the query takes less than a second

[HPN SEC11]

Global Variable Attack

```
Boolean found = false;
Function f(Record r){
    if(found)       return 1;
    if(r.name = Bob && r.disease = Cancer){
            found = true; return 1;
    } else return 0;
}

Function countf(){
    var  fs = from record in data
                where (f(record))
    print fs.NoisyCount(0.1);
}
```

This transformation changes the sensitivity –
if Bob has Cancer, then all subsequent records return 1

[HPN SEC11]

Privacy Budget Attack

```
Function is_f(Record r){
    if(r.name = Bob && r.disease = Cancer){
        run a sub-query that uses a lot of the privacy budget;
    }
    return f(r);
}

Function countf(){
    var  fs = from record in data
              where (f(record))
    print fs.NoisyCount(0.1);
}
```

If Bob does not have Cancer, then privacy budget decreases by 0.1.

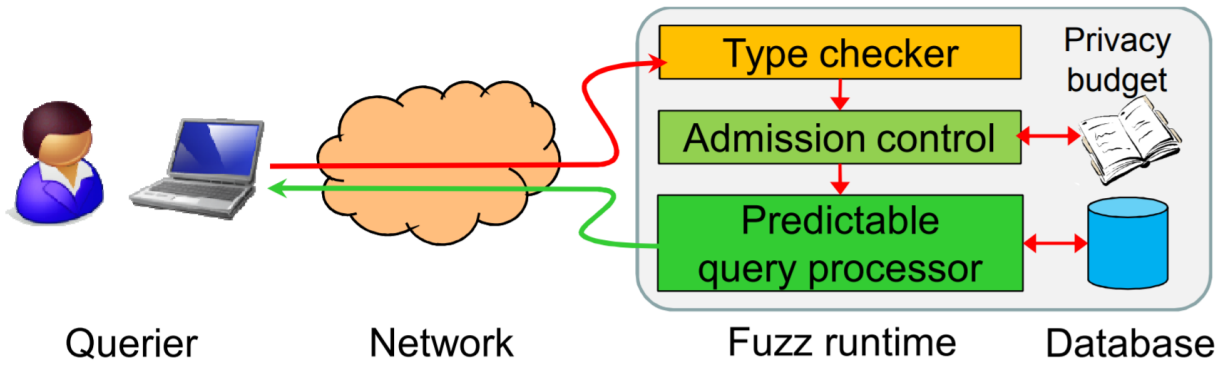
If Bob has Cancer, then privacy budget decreases by 0.1 + Δ.

Even if adversary can't query for the budget, he can detect the change in budget by counting how many more queries are allowed.

[HPN SEC11]

Avoid Covert Channel Attacks

- Fuzz:



  - Global variables are not supported in this language, thus ruling our *state attacks*.
  - **Type checker** rules out *budget-based channels* by statically checking the sensitivity of a query *before* they are executed
  - **Predictable query processor** ensures that each microquery takes the same amount of time, ruling out *timing attacks*.

[HPN SEC11]

Handling Timing Attacks

- Each microquery takes exactly the same time T
  - If it takes less time – delay the query
  - If it takes more time – abort the query
- **But this can leak information!**
  - **Wrong Solution**
- If it takes more time – **return a *default* value**

Summary

- Part I: Differential Privacy Basics
  - Laplace mechanism
  - Global sensitivity analysis
    - Transformations and stability (first proposed in PINQ)
  - PINQ: Implementation of DP

- Part II: Implementation Challenges
  - Highly sensitive queries (Joins)
    - Global sensitivity, local sensitivity, smooth sensitivity, elastic sensitivity (FLEX)
  - Attacks on DP Implementations
    - Timing, global variable, privacy budget (Fuzz)

Paper Readings

- Week 3
  - 1a. PrivateSQL
  - 1b. Airavat
  - 1c. VideoDP

- Week 4
  - 1d. DP for Streams
  - 1e. DP for Growing DB
  - 1f. Formalize Data Deletion

Announcement

- Paper reviews (next week)
  - Submit on Learn by Tue 9am
- Paper presentation
  - Either during the Tue live session
  - Or upload a video to Learn by Mon, 11pm
- Assignment 1
  - Release after Wed session
  - Latex file will be available
  - Submit pdf on Learn, by Feb 1, 11pm

Paper Review Guideline

- ½ page – 1 page
- A summary of the paper:
  - Motivation, problem, approach, result
- 3 Strengths
- 3 Weaknesses

Paper Presentation Guideline

- A summary of the paper
  - Motivation, problem, approach, result
- One technical piece in the paper
  - e.g. how result was reached
- One slide on points for discussion/open questions

Discussion Time